

Подробное руководство по созданию формул в Excel 2002

Джон Уокенбах

Автор книги *Excel 2002 Power Programming with VBA*

“Даже если ваша полка забита книгами по Excel, найдите на ней место еще и для этой”.

— Отзыв журнала *Microsoft OfficePro* о предыдущем издании книги

PROFESSIONAL MINDWARE™



**Подробное
руководство
по созданию формул
в Excel 2002**

Excel 2002 Formulas

John Walkenbach



M&T Books

An imprint of Hungry Minds, Inc.

Best-Selling Books ♦ Digital Downloads ♦ e-Book ♦ Answer Networks ♦
e-Newsletters ♦ Branded Web Sites ♦ e-Learning

New York, NY ♦ Cleveland, OH ♦ Indianapolis, IN

Подробное руководство по созданию формул в Excel 2002

Джон Уокенбах



ДИАЛЕКТИКА
Москва ♦ Санкт-Петербург ♦ Киев
2002

ББК 32.973.26-018.2.75

У62

УДК 681.3.07

Компьютерное издательство “Диалектика”

Зав. редакцией В.В. Александров

Перевод с английского *А.С. Барбаш, Ю.Г. Гусар, В.Н. Захаренко, В.А. Мартыновского, Е.А. Мастерских, Н.В. Наумовой и В.И. Сябера*

Под редакцией *И.В. Василенко*

По общим вопросам обращайтесь в издательство “Диалектика” по адресу:
info@dialektika.com, <http://www.dialektika.com>

Уокенбах, Джон.

У62 Подробное руководство по созданию формул в Excel 2002. : Пер. с англ. — М. :
Издательский дом “Вильямс”, 2002. — 624 с. : ил. — Парал. тит. англ.

ISBN 5-8459-0314-9 (рус.)

Большинство изданий, посвященных Excel, представляют собой универсальные руководства по программному продукту, содержащие информацию об использовании всех его средств и инструментов. Наряду с этим вы вряд ли найдете книгу, посвященную одному из основополагающих компонентов Excel, имеющему исключительно важное значение для любого пользователя, а именно — формулам. Ведь именно формулы делают электронную таблицу действительно электронной. Следовательно, чем больше вы знаете о формулах, тем лучше будут ваши таблицы. Большая часть материала, представленного в этой книге, посвящена именно тем вопросам, которые пользователи задают в ходе решения поставленной задачи. Вы также найдете ответы на те вопросы, которые не приходят в голову даже в конце обучения. Эта книга не предназначена для начинающих пользователей Excel. В ней рассматривается достаточно сложный материал, который утомителен для начинающих пользователей, но чрезвычайно порадует тех, кому часто и долго приходится управлять электронными таблицами.

ББК 32.973.26-018.2.75

Все названия программных продуктов являются зарегистрированными торговыми марками соответствующих фирм.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотокопирование и запись на магнитный носитель, если на это нет письменного разрешения издательства Hungry Minds, Inc.

Copyright © 2002 by Dialektika Computer Publishing.
Original English language edition copyright © 2001 by Hungry Minds, Inc.

All rights reserved including the right of reproduction in whole or in part in any form.
This edition published by arrangement with the original publisher, Hungry Minds, Inc.

ISBN 5-8459-0314-9 (рус.)
ISBN 0-7645-4800-X (англ.)

© Компьютерное изд-во “Диалектика”, 2002
© Hungry Minds, Inc., 2001

Оглавление

Введение	24
ЧАСТЬ I. ОСНОВНЫЕ СВЕДЕНИЯ	31
Глава 1. Коротко об Excel	33
Глава 2. Основные сведения о формулах	55
Глава 3. Работаем с именами	77
ЧАСТЬ II. ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ В ФОРМУЛАХ	107
Глава 4. Введение в функции Excel	109
Глава 5. Работа с текстом	123
Глава 6. Работа с данными даты и времени	145
Глава 7. Способы суммирования и подсчета данных	177
Глава 8. Поиск данных	201
Глава 9. Базы данных и списки	221
Глава 10. Дополнительные вычисления	247
ЧАСТЬ III. ФИНАНСОВЫЕ ФОРМУЛЫ	265
Глава 11. Знакомство с финансовыми формулами	267
Глава 12. Финансовые функции дисконтирования и обесценивания	293
Глава 13. Дополнительное использование финансовых функций и формул	311
ЧАСТЬ IV. ФОРМУЛЫ МАССИВА	325
Глава 14. Введение в массивы	327
Глава 15. Магия формул массива	343
ЧАСТЬ V. СОВЕРШЕННОЕ ВЛАДЕНИЕ ФОРМУЛАМИ	363
Глава 16. Умышленные циклические ссылки	365
Глава 17. Методы построения диаграмм	377
Глава 18. Сводные таблицы	415
Глава 19. Условное форматирование и проверка вводимых значений	433
Глава 20. Создание мегаформул	453
Глава 21. Инструменты и методы отладки формул	467
ЧАСТЬ VI. РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ	485
Глава 22. Введение в VBA	487
Глава 23. Основы процедур функций	497
Глава 24. Принципы программирования на VBA	513
Глава 25. Примеры пользовательских функций VBA	539
ПРИЛОЖЕНИЯ	573
Приложение А. Работа с импортированными файлами Lotus 1-2-3	574
Приложение Б. Описание функций Excel	580
Приложение В. Пользовательские форматы чисел	592
Приложение Г. Дополнительные ресурсы Excel	610
Предметный указатель	615

Содержание

Введение	24
ЧАСТЬ I. ОСНОВНЫЕ СВЕДЕНИЯ	31
Глава 1. Коротко об Excel	33
Краткая история Excel	33
Сначала была VisiCalc	34
Потом появился Lotus	34
Программные продукты компании Microsoft	34
Версии Excel	35
Концепция объектной модели	37
Работа с книгами	37
Таблицы	38
Листы диаграмм	39
Листы макросов XLM	39
Листы диалоговых окон	39
Пользовательский интерфейс Excel	40
Меню	40
Контекстные меню	40
Смарт-тэги	41
Диалоговые окна	41
Панели инструментов	42
Перетаскивание данных	42
Комбинации клавиш	42
Настройка отображения дополнительных элементов интерфейса	43
Ввод данных в ячейки	43
Выделение ячеек и других объектов	43
Форматирование ячеек	44
Числовое форматирование	44
Стилевое форматирование	45
Формулы и функции электронных таблиц	46
Объекты графического слоя	46
Фигуры	46
Диаграммы	46
Ссылки на графические объекты	47
Карты	47
Элементы управления диалоговыми окнами	47
Диаграммы	48
Настройка Excel в соответствии с требованиями пользователя	48
Макросы	48
Панели инструментов	48
Надстройки	49

Средства анализа	49
Доступ к базам данных	49
Структура	50
Управление сценариями	51
Надстройка “Пакет анализа”	51
Сводные таблицы	51
Средства проверки данных	51
Надстройка “Поиск решения”	51
Уровни защиты	52
Защита формулы от перезаписи	52
Защита структуры рабочей книги	52
Резюме	53
Глава 2. Основные сведения о формулах	55
Ввод и редактирование формул	55
Элементы формул	55
Ввод формул	56
Вставка имен	57
Пробел и разрыв строки	57
Предельный размер формул	58
Примеры формул	58
Редактирование формул	59
Применение операторов в формулах	60
Операторы ссылки	61
Примеры формул с операторами	61
Приоритет операторов	62
Вложенные скобки	63
Вычисление формул	64
Ссылки на ячейки и диапазоны ячеек	65
Создание абсолютной ссылки	65
Создание ссылок на ячейки других рабочих листов или рабочих книг	67
Создание точной копии формулы	67
Преобразование формул в значения	68
Скрытие формул	70
Ошибки в формулах	71
Работа с циклическими ссылками	72
Функция подбора параметра	73
Пример подбора параметров	74
Еще немного о подборе параметра	75
Резюме	75
Глава 3. Работаем с именами	77
Что такое Имя?	77
Способы присвоения имен ячейкам и диапазонам	78
Присвоение имен в окне Присвоение имени	78
Присвоение имени с помощью поля Имя	79
Автоматическое присвоение имени	80
Имя столбца или строки	81

Имена, созданные программой Excel	82
Присваивание имен в нескольких листах	83
Область действия имен	84
Создание имен уровня рабочего листа	84
Комбинирование имен уровня рабочего листа с именами уровня рабочей книги	85
Ссылка на имя из другой рабочей книги	85
Работа с именами диапазонов и ячеек	85
Создание списка имен	85
Использование имен в формулах	86
Использование операторов пересечения	87
Использование операторов диапазона с именами	88
Ссылка на отдельную ячейку именованного диапазона	89
Применение имен в уже существующих формулах	89
Автоматическое применение имен при создании формул	90
Отмена имени	90
Удаление имен	90
Удаление именованных ячеек и диапазонов	91
Переназначение имени	91
Изменение имени	91
Просмотр именованных диапазонов	91
Использование имен в диаграммах	92
Поддержка имен ячеек и диапазонов	92
Вставка строки или столбца	92
Удаление строки или столбца	93
Вырезание и вставка именованного диапазона	93
Проблемы при работе с именами	93
Проблемы, возникающие при копировании рабочих листов	93
Проблемы, возникающие при удалении рабочих листов	94
Ключ к пониманию термина “имя”	95
Присвоение имени константе	96
Присвоение имени текстовым константам	96
Использование функций рабочего листа в именованных формулах	97
Использование ссылок на ячейки и диапазоны в именованных формулах	98
Использование именованных формул с относительными ссылками	98
Решение сложных задач с помощью формул	101
Функция ДВССЫЛ и именованный диапазон	101
Использование функции ДВССЫЛ для создания именованного диапазона с фиксированным адресом	102
Использование массивов в именованных формулах	102
Создание формулы с динамическим именем	104
Резюме	105

ЧАСТЬ II. ИСПОЛЬЗОВАНИЕ ФУНКЦИЙ В ФОРМУЛАХ 107

Глава 4. Введение в функции Excel	109
Что такое функция?	109

Упрощение формулы	109
Вычисления, невозможные без функций	110
Ускорение выполнения задач редактирования	110
Способность к принятию решений	110
Еще немного о функциях	110
Типы аргументов функций	111
Имена в качестве аргументов	111
Задание строки или столбца в качестве аргумента	112
Точные значения в качестве аргументов	113
Математические выражения в качестве аргументов	113
Функции в качестве аргументов	113
Массивы в качестве аргументов	114
Способы введения функции в формулу	114
Ручной ввод функций	114
Вставка функции с помощью диалогового окна Мастер функций	115
Несколько полезных советов по вводу функций	116
Категории функций	118
Финансовые функции	118
Функции даты и времени	118
Математические функции	118
Статистические функции	119
Функции ссылок и массивов	119
Функции работы с базами данных	119
Текстовые функции	119
Логические функции	119
Информационные функции	119
Инженерные функции	120
Функции, созданные пользователем	120
Другие категории функций	120
Функции настройки “Пакет анализа”	121
Резюме	121
Глава 5. Работа с текстом	123
Несколько слов о самом тексте	123
Количество символов в ячейке	123
Числа как текст	124
Текстовые функции	125
Определение наличия в ячейке текста	125
Работа с кодами символов	126
Равенство строк?	128
Объединение двух и более ячеек	129
Отображение значений, отформатированных в текстовом формате	130
Отображение денежных значений, отформатированных как текст	131
Повторяющиеся символы или строки	131
Создание текстовой гистограммы	131
Добавление к числу определенных символов	132
Удаление лишних пробелов и непечатаемых символов	133

Подсчет символов в строке	133
Изменение регистра текста	134
Извлечение из строки определенных символов	134
Замена одного текста другим	135
Поиск и определение расположения символа в строке	136
Поиск и замена символа в пределах строки	136
Дополнительные текстовые формулы	137
Подсчет символов в ячейке	137
Подсчет строк, содержащихся в ячейке	137
Определение заданного числа как порядкового номера	138
Определение литеры столбца по заданному номеру	138
Извлечение имени файла из заданного пути	139
Извлечение первого слова строки	139
Извлечение последнего слова строки	139
Извлечение всех слов строки кроме первого	140
Извлечение имени, отчества и фамилии	140
Удаление из имен титулов и званий	141
Подсчет количества слов в ячейке	141
Создание собственных текстовых функций с помощью языка VBA	142
Резюме	143
Глава 6. Работа с данными даты и времени	145
Принципы управления данными даты и времени в Excel	145
Понятие даты как порядкового номера	146
Введение даты	146
Понятие времени как порядкового номера	148
Введение времени	149
Форматирование даты и времени	150
Трудности, возникающие при использовании даты	151
Функции управления датами	153
Отображение текущей даты	154
Отображение различных дат	154
Создание ряда последовательных дат	155
Преобразование в дату строки, не означающей дату	156
Вычисление количества дней между двумя датами	156
Вычисление количества рабочих дней между двумя датами	157
Вычисление даты по графику рабочих дней	158
Вычисление количества лет между двумя датами	159
Определение возраста человека	159
Вычисление дня года	159
Определение дня недели	160
Определение даты самого ближайшего воскресенья	161
Вычисление дня недели, следующего за определенной датой	162
Определение даты дня недели в заданном месяце	162
Определение числа заданных дней недели в месяце	162
Вычисление праздничных дат	163
Определение даты последнего дня месяца	165

Определение високосного года?	166
Вычисление квартального периода	166
Отображение года римскими цифрами	166
Создание календаря в отдельном диапазоне ячеек	167
Функции управления временем	167
Отображение текущего времени	168
Отображение времени	168
Суммирование времени, превышающего 24 часа	169
Определение разницы между двумя временными значениями	171
Преобразование “военного времени”	172
Преобразование часов, минут, секунд в десятичном исчислении	172
Добавление часов, минут или секунд к заданному времени	173
Работа с временными поясами	173
Округление времени	174
Работа со значениями, не имеющими отношения ко времени суток	175
Резюме	176
Глава 7. Способы суммирования и подсчета данных	177
Подсчет и суммирование ячеек электронной таблицы	177
Подсчет или суммирование записей в базах данных и сводных таблицах	178
Основные формулы подсчета	179
Подсчет общего количества ячеек	180
Подсчет пустых ячеек	180
Подсчет непустых ячеек	181
Подсчет ячеек, содержащих числовые значения	181
Подсчет ячеек, содержащих нетекстовые значения	181
Подсчет ячеек, содержащих текстовое значение	182
Подсчет логических значений	182
Значения ошибки диапазона данных	182
Дополнительные формулы подсчета ячеек	182
Подсчет ячеек с использованием функции СЧЕТЕСЛИ	183
Подсчет ячеек с использованием нескольких условий	184
Подсчет часто встречающихся записей	185
Подсчет количества экземпляров определенного текста	186
Подсчет количества одинаковых значений	187
Распределение частот	188
Формулы суммирования	193
Суммирование всех ячеек диапазона	193
Общая сумма	194
Суммирование n-го количества наибольших или наименьших значений	195
Условное суммирование по одному критерию	195
Суммирование только отрицательных значений	197
Суммирование значений на основе диапазона Difference	197
Суммирование значений на основе сравнения текста	197
Суммирование значений на основе сравнения даты	198
Условные суммы на основе множества критериев	198
Использование оператора И	198

Использование оператора ИЛИ	199
Использование условий И и ИЛИ	199
Использование функций VBA для подсчета и суммирования данных	200
Резюме	200
Глава 8. Поиск данных	201
Знакомство с формулой поиска данных	201
Функции, используемые для поиска данных	202
Основные формулы поиска данных	202
Функция ВПР	202
Функция ГПР	204
Функция ПРОСМОТР	205
Комбинирование функций ПОИСКПОЗ и ИНДЕКС	206
Специальные формулы поиска данных	208
Поиск точного значения	208
Поиск значения слева	209
Поиск с учетом регистра	210
Поиск значения в множестве таблиц	210
Определение оценки по количеству набранных баллов	211
Вычисление среднего уровня оценок	212
Поиск на пересечении строк и столбцов	213
Поиск информации в двух столбцах	214
Определение расположения значения в диапазоне	215
Поиск значения, максимально приближенного к заданному	216
Поиск значения с использованием линейной интерполяции	217
Резюме	219
Глава 9. Базы данных и списки	221
Знакомство со списками и базами данных электронной таблицы	221
Использование средства Автофильтр	223
Принципы работы средства Автофильтр	223
Подсчет и суммирование отфильтрованных данных	225
Копирование и удаление отфильтрованных данных	226
Использование расширенного фильтра	227
Создание диапазона условий	228
Фильтрация списка	229
Определение условий расширенного фильтра	230
Определение одного условия	231
Определение множества условий	233
Определение вычисляемых условий	236
Использование функций базы данных	238
Суммирование списка с помощью таблицы подстановки	240
Создание промежуточных итогов	243
Резюме	245
Глава 10. Дополнительные вычисления	247
Преобразование единиц измерений	247
Таблицы преобразования единиц измерения	247
Использование таблиц преобразования единиц измерения	248

Преобразования единиц измерения длины	249
Преобразования единиц измерения веса	249
Преобразования единиц измерения жидкости	249
Преобразования единиц измерения площади	249
Преобразования единиц измерения объема	249
Преобразования единиц измерения силы	249
Преобразования единиц измерения энергии	250
Преобразования единиц измерения времени	250
Преобразование единиц измерения температуры	253
Решение задач для прямоугольных треугольников	253
Вычисление периметра, длины окружности, площади плоских фигур, поверхности и объема тел	255
Вычисление площади и периметра квадрата	255
Вычисление площади и периметра прямоугольника	256
Вычисление площади и периметра окружности	256
Вычисление площади трапеции	256
Вычисление площади треугольника	256
Вычисление площади поверхности и объема шара	256
Вычисление площади поверхности и объема куба	257
Вычисление площади поверхности и объема конуса	257
Вычисление объема цилиндра	257
Вычисление объема пирамиды	257
Решение систем линейных уравнений	258
Округления	259
Основные формулы округления	260
Округление значений денежных величин	260
Работа с дробными значениями денежных единиц	261
Использование функций ЦЕЛОЕ и ОТБР	262
Округление до четного и нечетного целого	262
Округление значения до n-го разряда значащих цифр	262
Резюме	263

ЧАСТЬ III. ФИНАНСОВЫЕ ФОРМУЛЫ **265**

Глава 11. Знакомство с финансовыми формулами	267
Основные финансовые функции Excel	267
Анализ денежных потоков	268
Функции накопления, дисконтирования и амортизации	269
Простые задачи накопления	270
Сложные задачи накопления	273
Простые задачи дисконтирования	275
Сложные задачи дисконтирования	277
Задачи амортизации	278
Преобразование процентных ставок	281
Методы указания процентных ставок	282
Преобразование процентных ставок с помощью встроенных финансовых функций	282

Эффективная стоимость займов	283
Воздействие платежей на эффективную ставку	284
Займы с единообразной ставкой	284
Беспроцентные займы	285
Заем с “ежегодными” выплатами	285
Вычисление процента и основных платежей	286
Использование функций ОСПЛТ и ПРПЛТ	286
Использование функций ОБЩДОХОД и ОБЩПЛАТ	287
Соответствие процентов и частоты выплаты	288
Ограничения финансовых функций Excel	289
Отсроченное начало ряда регулярных выплат	289
Оценка серии регулярных платежей	290
Резюме	291
Глава 12. Финансовые функции дисконтирования и обесценивания	293
Использование функции ЧПС	293
Определение ЧПС	294
Примеры функции ЧПС	294
Использование функции ЧПС для вычисления накопленных сумм	299
Использование функции ВСД	300
Несколько ставок в функциях ВСД и МВСД	304
Использование функции БЗРАСПИС	306
Вычисления обесценивания	308
Резюме	310
Глава 13. Дополнительное использование финансовых функций и формул	311
Создание динамических финансовых планов	311
Создание планов амортизации	312
Простой план амортизации	312
Детальный план амортизации	314
План амортизации займа с переменной ставкой	315
Задание параметров займа с помощью таблицы данных	316
Создание односторонней таблицы данных	316
Создание двухсторонней таблицы подстановки	317
План накопления	319
Планы будущего поступления	320
Функции ЧИСТВНДОХ и ЧИСТНЗ	321
Резюме	323
ЧАСТЬ IV. ФОРМУЛЫ МАССИВА	325
Глава 14. Введение в массивы	327
Введение в формулы массива	327
Формула массива, возвращающая несколько значений	327
Формула массива, возвращающая одно значение	329
Создание массива констант	329
Элементы массива констант	330
Понятие размерности массива	330
Одномерные горизонтальные массивы	330

Одномерные вертикальные массивы	331
Двумерные массивы	331
Присвоение имени массивам констант	332
Работа с формулами массива	333
Ввод формулы массива	333
Выделение диапазона формулы массива	333
Редактирование формулы массива	334
Расширение или сокращение формулы массива	334
Использование формул массива, возвращающих несколько значений	335
Создание массива из значений в диапазоне	335
Создание массива констант из значений диапазона	336
Выполнение действий в массиве	336
Использование функций в операциях с массивами	337
Транспонирование массива	337
Генерирование массива последовательных целых чисел	338
Использование формул массива, возвращающих одно значение	338
Подсчет количества символов в диапазоне	339
Сложение трех наименьших элементов диапазона	339
Подсчет количества текстовых ячеек в диапазоне	340
Устранение промежуточных формул	340
Использование массива вместо диапазона ссылок	341
Резюме	342
Глава 15. Магия формул массива	343
Работа с формулами массива, возвращающими одно значение	343
Суммирование в диапазоне, который содержит ошибки	343
Подсчет количества ошибок в диапазоне	344
Суммирование при выполнении условия	344
Суммирование n-го числа наибольших значений в диапазоне	346
Вычисление среднего без учета нулевых значений	346
Поиск значения в диапазоне	347
Подсчет отличающихся значений в двух диапазонах	348
Возвращение расположения максимального значения диапазона	349
Поиск номера строки, в которой находится n-ое значение, совпадающее с заданным	349
Возвращение самого длинного текста в диапазоне	349
Определение допустимых значений диапазона	350
Определение суммы цифр числа	350
Суммирование округленных значений	351
Суммирование каждого n-го значения в массиве	352
Исключение нечисловых символов из текстовой строки	353
Определение ближайшего значения в диапазоне	353
Возвращение последнего значения в столбце	353
Возвращение последнего значения в строке	354
Упорядочение данных с помощью формулы массива	354
Создание динамических перекрестных таблиц	355
Работа с формулами массива, возвращающими несколько значений	356

Извлечение положительных значений из диапазона	356
Извлечение непустых ячеек из диапазона	357
Изменение на противоположный порядка следования элементов в диапазоне	357
Динамическая сортировка величин в диапазоне	357
Возвращение списка уникальных значений диапазона	358
Отображение календаря в диапазоне	358
Возвращение массива с помощью VBA-функций	359
Резюме	361

ЧАСТЬ V. СОВЕРШЕННОЕ ВЛАДЕНИЕ ФОРМУЛАМИ **363**

Глава 16. Умышленные циклические ссылки	365
Что такое циклические ссылки?	365
Исправление случайных циклических ссылок	366
Косвенные циклические ссылки	367
Умышленные циклические ссылки	367
Определение параметров вычислений и итераций	369
Примеры формул, использующих циклические ссылки	370
Создание временного снимка ячейки	370
Вычисление всегда-самого-большого значения	370
Генерирование уникальных случайных чисел	371
Решение рекурсивного уравнения	372
Решение системы уравнений с помощью циклических ссылок	373
Возможные проблемы, связанные с использованием циклических ссылок	374
Резюме	375
Глава 17. Методы построения диаграмм	377
Представление данных на диаграммах	377
Знакомство с формулой РЯД	377
Создание ссылок на ячейки	380
График выполнения работ	383
Создание диаграммы Гантта	384
Создание гистограммы сравнения	386
Создание диаграммы разброса данных	387
Построение каждой n-ой точки данных	389
Автоматическое обновление рядов данных	390
Отображение n последних точек данных	391
Интерактивное отображение данных	392
Построение диаграммы на основе выбора строки	392
Выбор данных из списка	393
Построение графиков функций с одной переменной	395
Построение графиков функций с двумя переменными	397
Создание экстравагантных проектов	399
Диаграммы тенденций	399
Линейные тенденции	400
Нелинейные тенденции	403
Полезные методы построения диаграмм	407
Сохранение нескольких диаграмм на одном листе	407

Просмотр внедренной диаграммы в окне	408
Изменение значений рабочего листа путем перетаскивания точек данных	409
Использование анимированных диаграмм	409
Создание “измерительной” диаграммы	410
Создание диаграммы часов	410
Рисование с помощью графиков	413
Резюме	414
Глава 18. Сводные таблицы	415
О сводных таблицах	415
Пример сводной таблицы	416
Данные, используемые в сводной таблице	418
Создание сводной таблицы	419
Шаг 1. Указание расположения данных	420
Шаг 2. Определение данных	421
Шаг 3. Завершение создания сводной таблицы	421
Группирование элементов сводной таблицы	426
Создание вычисляемых полей или вычисляемых элементов	428
Создание в сводной таблице вычисляемого поля	428
Вставка в сводную таблицу вычисляемого элемента	431
Резюме	432
Глава 19. Условное форматирование и проверка вводимых значений	433
Условное форматирование	433
Настройка условного форматирования	434
Возможные типы форматирования	434
Определение условий	435
Работа с условными форматами	437
Формулы условного форматирования	439
Использование пользовательских функций в формулах условного форматирования	445
Проверка вводимых значений	447
Определение критерия достоверности данных	448
Типы возможных условий проверки вводимых значений	449
Использование формул в условиях проверки вводимых значений	450
Использование формул проверки вводимых значений для ввода только определенных значений	451
Резюме	452
Глава 20. Создание мегаформул	453
Что такое мегаформула?	453
Создание мегаформулы: простой пример	453
Примеры мегаформул	455
Использованию мегаформул для удаления вторых имен и инициалов	455
Использование мегаформулы для нахождения положения последнего пробела в строке	458
Использование мегаформулы для определения действительности номера кредитной карточки	461
Преимущества и недостатки использования мегаформул	464
Резюме	465

Глава 21. Инструменты и методы отладки формул	467
Отладка формул?	467
Проблемы формул и их решение	468
Недостающие скобки	468
Ячейки заполнены знаками #####	469
Пустые ячейки не пустые	470
Формула возвращает ошибку	470
Абсолютные и относительные ссылки	473
Приоритет операторов	473
Формулы не вычислены	474
Реальные и отображаемые значения	475
Ошибки плавающей запятой	475
Ошибка “фантомных ссылок”	476
Ошибка циклических ссылок	476
Средства проверки данных Excel	476
Нахождение ячейки определенного типа	476
Просмотр формул	477
Отслеживание связей ячейки	478
Отслеживание ошибочных значений	480
Исправление ошибок циклической ссылки	480
Использование средства фоновой проверки ошибок Excel 2002	480
Проверка формул в Excel 2002	482
Средства проверки данных сторонних разработчиков	482
Power Utility Pak	483
Spreadsheet Detective	484
Excel Auditor	484
Резюме	484
ЧАСТЬ VI. РАЗРАБОТКА ПОЛЬЗОВАТЕЛЬСКИХ ФУНКЦИЙ	485
Глава 22. Введение в VBA	487
Несколько слов о VBA	487
Введение в редактор Visual Basic	487
Запуск редактора Visual Basic	488
Компоненты редактора Visual Basic	489
Использование окна проекта	490
Использование окна кода	492
Введение кода	493
Сохранение проекта	496
Резюме	496
Глава 23. Основы процедур функций	497
Для чего нужны пользовательские функции?	497
Простой пример VBA-функции	497
Работа процедуры функции	499
Объявления функции	499
Выбор имени функции	500
Использование функций в формулах	500

Аргументы функций	501
Использование диалогового окна вставки функции	501
Добавление описания функции	502
Определение категории функции	502
Тестирование и отладка функций	504
Использование окна сообщений VBA	505
Использование в коде операторов Debug.Print	506
Вызов функции из процедуры	507
Установка точек останова в функции	509
Создание надстроек	510
Резюме	511
Глава 24. Принципы программирования на VBA	513
Пример процедуры функции	513
Использование комментариев в коде	515
Использование переменных, типов данных и констант	515
Определение типов данных	516
Объявление переменных	517
Использование констант	518
Использование строк	519
Работа с датами	519
Использование выражений присвоения	520
Использование массивов	521
Объявление массива	521
Описание многомерных массивов	521
Использование встроенных функций VBA	522
Управление выполнением кода	523
Конструкция If-Then	524
Конструкция Select Case	525
Создание цикла операторов	526
Оператор On Error	529
Работа с диапазонами	531
Конструкция For Each-Next	531
Ссылка на диапазон	532
Часто используемые свойства диапазона	533
Ключевое слово Set	536
Функция Intersect	536
Функция Union	537
Свойство UsedRange	537
Резюме	538
Глава 25. Примеры пользовательских функций VBA	539
Простые функции	539
Есть ли в ячейке формула?	539
Возвращение формулы ячейки	540
Скрыта ли ячейка?	540
Возвращение имени рабочего листа	540
Возвращение имени рабочей книги	541

Возвращение имени приложения	542
Возвращение версии Excel	542
Возвращение информации о форматировании ячейки	542
Определение типа данных в ячейке	543
Многофункциональная функция	544
Генерирование случайных чисел	546
Генерирование неизменяемых случайных чисел	546
Выбор случайной ячейки	546
Вычисление комиссионных от продаж	547
Функция простой структуры начисления комиссионных	548
Функция сложной структуры начисления комиссионных	549
Функции управления текстом	549
Реверсирование строки	550
Перетасовка текста	550
Возвращение аббревиатуры	551
Соответствие текста шаблону?	551
Содержится ли в ячейке текст?	552
Извлечение n-ого элемента строки	552
Написание цифры прописью	554
Функции подсчета и суммирования	554
Подсчет ячеек с определенными значениями	554
Подсчет видимых ячеек диапазона	555
Суммирование видимых ячеек диапазона	555
Функции управления датами	556
Расчет следующего понедельника	556
Вычисление следующего дня недели	557
Неделя месяца	557
Управление датами до 1900 года	558
Возвращение последней непустой ячейки столбца или строки	558
Функция LASTINCOLUMN	559
Функция LASTINROW	559
Функции для работы в нескольких листах	559
Возвращение максимального значения из рабочих листов	560
Функция SHEETOFFSET	561
Дополнительные пользовательские функции	562
Возвращение значения Error	562
Возвращение массива из функции	563
Возвращение массива из неповторяющихся случайных целых чисел	565
Перетасовка массива	566
Использование необязательных аргументов	567
Использование неопределенного количества аргументов	568
Резюме	571

ПРИЛОЖЕНИЯ	573
Приложение А. Работа с импортированными файлами Lotus 1-2-3	574
О файлах Lotus 1-2-3	574

Формулы Lotus 1-2-3	575
Порядок вычислений	576
Текст в вычислениях	576
Логические значения	576
Проблемы с датами	577
Критерии базы данных	577
Совместимость функций Lotus 1-2-3	578
Функции конвертирования баз данных	578
Приложение Б. Описание функций Excel	580
Функции Excel по категориям	580
Приложение В. Пользовательские форматы чисел	592
О форматировании числа	592
Автоматическое форматирование цифр	592
Форматирование чисел с помощью кнопок панели инструментов	593
Использование клавиатурных сокращений для форматирования чисел	593
Использование диалогового окна Формат ячеек для форматирования чисел	594
Создание пользовательского формата числа	595
О пользовательских форматах чисел	596
Части строки с форматом числа	596
Коды пользовательского формата числа	597
Примеры пользовательских форматов	599
Масштаб	599
Скрытие нуля	602
Отображение нулей впереди	602
Форматирование процентных значений	602
Отображение дробей	602
Отображение текста в кавычках	603
Повтор текста	603
Отображение знака минус справа	604
Условное форматирование числа	604
Выделение цветом	605
Форматирование дат и времени	605
Отображение текста с числами	606
Отображение нуля с тире	606
Использование специальных символов	606
Устранение определенных типов элементов	607
Заполнение ячейки повторяющимися символами	608
Отображение точек впереди	609
Приложение Г. Дополнительные ресурсы Excel	610
Техническая поддержка Microsoft	610
Параметры поддержки	610
База знаний Microsoft	610
Домашняя страница Microsoft Excel	610
Инструменты Microsoft Office в Web	611
Форумы в Internet	611
Форум по электронным таблицам	611

Форум Microsoft	611
Поиск по форуму	612
Web-узлы	613
Страница электронных таблиц	613
Web-ресурс по Excel	613
Страница по Excel Стефена Баллена	614
Вопросы и ответы	614
Предметный указатель	615

Об авторе

Джон Уокенбах (John Walkenbach) входит в узкий круг лучших специалистов по использованию программного обеспечения, предназначенного для управления электронными таблицами. Кроме того, он является директором консалтинговой фирмы Jwalk and Associates Inc., которая специализируется на разработке приложений для процессоров электронных таблиц. За свою многолетнюю практику Джон успел написать около 30 книг и опубликовать более 300 статей в самых различных изданиях, посвященных электронному представлению табличных данных, таких как *PC World*, *InfoWorld*, *PC Magazine* и *PC Computing*. Он также самостоятельно поддерживает весьма популярный Web-узел The Spreadsheet Page (www.j-walk.com/ss). В число его собственных разработок входит пакет Power Utility Pak, лучшей из внедряемых в Excel надстроек.

Введение

Благодарю вас за то, что вы приобрели мою книгу. Если вы хотите научиться применять формулы и тем самым начать работу в Excel на качественно новом уровне, эта книга поможет вам в достижении поставленной цели. Гарантирую — вы не пожалеете о том, что потратили деньги на ее приобретение.

Почему я написал эту книгу

К работе над проектом я приступил с единственной целью: написать максимально подробное руководство по использованию формул в Excel, адресованное широкому кругу пользователей. Поставленная задача была не из легких. Однако, отзывы читателей о первом издании книги помогли мне справиться с ней.

Я работаю с Excel вот уже почти десять лет. Кроме того, принимал активное участие в сетевых конференциях, посвященных Excel. Поэтому мне очень хорошо известны те вопросы, которые постоянно возникают при использовании данного программного продукта. Большая часть материала, представленного в этой книге, посвящена именно тем вопросам, которые пользователи задают в ходе конференций. Однако, в этой книге вы также найдете ответы на те вопросы, которые не приходят в голову в конце обучения.

Вы, вероятно, знаете, что в большинстве книжных магазинов вам могут предложить с десяток книг по Excel. В основном эти издания представляют собой универсальные руководства по программному продукту, содержащие информацию об использовании возможностей Excel (зачастую в этих книгах просто воспроизводится содержимое файлов справочной системы программы). Некоторые издания посвящены специфическим вопросам, таким как программирование на макроуровне или прикладные системы для научных исследований. И не существует ни одной книги (честное слово!), посвященной одному из основополагающих компонентов Excel, имеющему исключительно важное значение для любого пользователя, а именно — формулам. Фактически, именно формулы делают электронную таблицу действительно электронной. Следовательно, чем больше вы знаете о формулах, тем лучше будут ваши таблицы. Все просто.

Excel — лидер среди программных средств, позволяющих создавать электронные таблицы. Причем Excel намного опережает остальных. И не только благодаря сильной маркетинговой политике компании Microsoft. Просто Excel — действительно лучший из существующих на сегодняшний день процессоров электронных таблиц. Формулы — это та область, в которой преимущество Excel перед другими подобными программными средствами наиболее очевидно. В арсенале Excel припасено для вас множество трюков с формулами. Очень скоро вы убедитесь в том, что Excel позволит вам осуществить многое из того, что не под силу другим программам.

Могу поспорить, что не более десяти процентов пользователей Excel знают, насколько эффективно применять формулы в таблицах. Я надеюсь, что эта книга поможет вам пополнить ряды избранных. Итак, вы готовы?

Что вы должны знать

Эта книга не предназначена для начинающих пользователей Excel. Если у вас нет никакого опыта работы с этим программным продуктом, то книга для вас не самая подходящая.

Или, может быть, вы относитесь к той редкой породе людей, которые могут научиться пользоваться новой программой в одно мгновение?

Чтобы эта книга принесла вам максимальную пользу, вы должны иметь некоторый опыт работы в Excel. В частности предполагается, что вы знаете как:

- ◆ создавать рабочие книги Excel, вставлять таблицы, сохранять файлы и выполнять другие основные операции;
- ◆ перемещаться по рабочей книге;
- ◆ использовать меню, панели инструментов и диалоговые окна Excel;
- ◆ применять основные методы управления, принятые в Windows, в частности работать с файлами, копировать и вставлять те или иные элементы.



Если вы уже имеете опыт работы с электронными таблицами, но программа Excel вам не знакома, обратитесь к главе 1. Эта глава содержит краткий обзор того, что может предложить вам этот программный продукт.

Системные требования

Чтобы освоить материал данной книги, вам понадобится программа Microsoft Excel. Работая над книгой, я использовал Excel 2002 (которая входит в пакет Microsoft Office XP). За некоторым исключением (исключения отмечены в тексте), материал этой книги относится и ко всем ранним версиям Excel, которые все еще активно используются.



Я работаю исключительно в Excel для Windows, а не Macintosh. Следовательно, я не могу гарантировать, что все примеры будут работать в Excel для Macintosh. Для Excel характерна достаточно неплохая, но все же не идеальная межплатформенная совместимость.

Чем быстрее работает ваша система, тем лучше. И, естественно, емкая оперативная память придется вам как нельзя кстати. Настоятельно рекомендую: используйте высокий режим представления видеоизображений с разрешением, не менее 1024×768.

Условные обозначения, используемые в этой книге

Не поленитесь и потратьте немного времени на этот раздел. В нем вы узнаете, какие условные обозначения используются в этой книге.

Работа с клавиатурой

Клавиатура нужна для того, чтобы вводить формулы. Кроме того, с ее помощью вы можете работать с меню и диалоговыми окнами. Этот метод удобен для тех пользователей, которые хорошо владеют обеими руками.

Листинги формул

Формулы обычно располагаются в отдельной строке и выделяются моноширинным шрифтом. В качестве примера я приведу следующую запись:

```
=VLOOKUP (StockNumber, PriceList, 2, False)
```

Excel поддерживает такой специфический тип формул, как *формулы массивов*. Когда вы вводите формулу массива, нажмите <Ctrl+Shift+Enter> (а не просто <Enter>). Excel заключает такую формулу в фигурные скобки, таким образом напоминая вам о том, что это формула массива. Когда я привожу листинг формулы массива, я также заключаю ее в скобки. Например:

```
{=SUM (LEN (A1:A10) ) }
```



Записывая формулу массива не вводите скобки вручную. Excel сделает это автоматически.

Листинги кодов VBA

Эта книга содержит примеры кодов VBA. Каждый листинг выделяется моноширинным шрифтом; каждая строка кода располагается в отдельной строке листинга. Для того чтобы код легче воспринимался, я обычно использую несколько уровней отступов с помощью символов табуляции. Отступы не являются обязательными, но они помогают разделить код на блоки, строки которых связаны между собой.

Если строка кода не помещается в одной строке листинга, я использую стандартное для VBA обозначение продолжения строки: пробел, за которым следует символ подчеркивания. Это означает, что строка кода продолжается на следующей строке листинга. Например, две строки приведенного ниже листинга содержат одно выражение VBA:

```
If Right (cell.Value, 1) = "!" Then cell.Value _  
= Left (cell.Value, Len (cell.Value) - 1)
```

Вы можете ввести код в двух строках, как показано выше, или в одной строке без символа подчеркивания.

Названия клавиш

Названия клавиш клавиатуры приводятся обычным шрифтом, например <Alt>, <Home>, <PgDn> и <Ctrl>. Если необходимо нажать две клавиши одновременно, эти клавиши объединяются знаком плюс: “Для вызова диалогового окна Go To, нажмите <Ctrl+G>.”

Функции, процедуры и именованные диапазоны

Все функции таблиц Excel записываются в верхнем регистре, например: “Для того чтобы добавить значения в столбце A, используйте функцию SUM.”

Макронимена и имена процедур приводятся обычным шрифтом: “Выполните процедуру InsertTotals.” Я часто использую символы верхнего и нижнего регистра, что упрощает чтение таких имен. Именованные диапазоны выделены курсивом: “Выделите диапазон *InputArea*.”

Excel не чувствительна к регистру, если дело не касается текста внутри кавычек. Иными словами, две приведенные ниже формулы будут иметь одинаковый эффект:

```
=SUM (A1:A50)  
=sum (a1:a50)
```

Однако программа автоматически преобразует символы второй формулы к верхнему регистру.

Работа с мышью

В этой книге используется стандартная терминология для обозначения действий, производимых с помощью мыши: вы можете “указывать”, “щелкать на кнопке мыши”, “щелкать правой кнопкой мыши”, “перетаскивать” и т.д. Вы ведь знаете, о чем идет речь?

Пиктограммы

В этой книге пиктограммы располагаются у левого поля страницы. Они предназначены для того, чтобы обратить ваше внимание на материал, имеющий важное значение.



Эта пиктограмма указывает на нововведения, появившиеся в Excel 2002.



Эта пиктограмма используется для того, чтобы подчеркнуть важную в будущем информацию. Например, новый подход, который поможет вам справиться со сложным заданием, или же основополагающий принцип, необходимый для четкого понимания темы.



“Советы” укажут вам новый эффективный путь решения задачи, или методику, которую вы не догадались применить. Это наверняка поможет вам произвести впечатление на коллег.



Пиктограмма “Внимание” предостерегает вас от ошибок, которые можно совершить в том случае, если неправильно выполнить описываемую операцию.



Перекрестные ссылки указывают на другие главы, где вы найдете более подробную информацию по данной теме.

Как организована эта книга

Существует множество способов систематизации материала. Я выбрал такую схему, при которой книга делится на пять основных частей. Кроме того, данное издание имеет несколько приложений с дополнительной информацией, которая будет вам весьма полезна.

Часть I. Основные сведения

Эта часть — вводная. Она состоит из трех глав. Глава 1 представляет краткий обзор основных характеристик Excel. Она адресована пользователям, имеющим опыт работы с программными средствами создания электронных таблиц, но ранее не работавших в Excel. Глава 2 содержит основные сведения о формулах. Если вы хотите полностью освоить материал данной книги, обязательно прочитайте эту главу. Глава 3 посвящена именам. Вы думали, что только ячейки и диапазоны имеют имена? Прочитав эту главу вы поймете, что заблуждались.

Часть II. Использование функций в формулах

Эту часть составляют главы с 4 по 10. Глава 4 содержит основные сведения о функциях таблиц в формулах. Последующие главы посвящены конкретным темам. Прочитав главу 5, вы научитесь работать с текстом. Глава 6 содержит информацию о датах и времени, а глава 7 повествует о различных техниках вычислений. В главе 8 я расскажу вам о типах формул поиска данных. Глава 9 посвящена базам данных и спискам, а в главе 10 рассматриваются такие виды вычислений, как перевод одних единиц в другие и округление.

Часть III. Финансовые формулы

Часть III состоит из трех глав (с 11 по 13), посвященных созданию финансовых формул. В этой части книги вы найдете множество полезных примеров, которые помогут вам осуществлять практические задания.



Большая часть материала части III была предоставлена Норманом Хакером (Norman Harker). Норман — старший преподаватель Сиднейского Университета (Австралия), специализирующийся в области недвижимости.

Часть IV. Формулы массивов

Эта часть состоит из глав 14 и 15. Большинство пользователей Excel знают о формулах массивов очень мало или вообще ничего о них не знают. Мне же эта тема очень близка. И потому я посвятил часть книги этому чрезвычайно мощному инструменту, который пока применяется крайне редко.

Часть V. Различные методики применения формул

Часть V состоит из глав с 16 по 21. Эти главы посвящены целому ряду задач, причем некоторые из них, на первый взгляд, не имеют никакого отношения к формулам. Глава 16 демонстрирует преимущества циклической ссылки. Прочитав главу 17, вы поймете, как можно применять формулы при построении диаграмм, а глава 18 расскажет о том, какое отношение формулы имеют к сводным таблицам. В главе 19 вы найдете несколько интересных (и полезных) формул, которые можно использовать в Excel при проведении условного форматирования и проверке правильности данных. Глава 20 посвящена так называемым “мегаформулам”. Мегаформула — это формула, включающая несколько промежуточных формул. А что делать, если формулы не функционируют корректно? Ответ на этот вопрос вы найдете в главе 21, где описаны методики устранения неполадок.

Часть VI. Разработка пользовательских функций

Эта часть состоит из глав с 22 по 25. Часть VI посвящена Visual Basic for Application (VBA) — пакету, предназначенному для разработки пользовательских функций. Глава 22 познакомит вас с VBA и редактором VB, а глава 23 предоставит основную информацию о пользовательских функциях. Глава 24 освещает некоторые аспекты программирования, а в главе 25 вы найдете множество примеров пользовательских функций, которые сможете использовать в исходном виде или изменить в соответствии с собственными задачами.

Приложения

Вы когда-нибудь видели компьютерную книгу без единого приложения? В настоящем издании таких приложений пять. В этих приложениях вы найдете ценные советы по импортированию файлов и работе с пользовательскими числовыми форматами, краткий справочник функций Excel, а также список Internet-ресурсов, посвященных Excel.

Как пользоваться книгой

Вы можете читать эту книгу, как вам вздумается. При желании, вы можете прочесть книгу от корки до корки лежа на солнечном пляже где-нибудь на Майами. Но скорее всего вы будете держать эту книгу под рукой, в поте лица трудясь над очередным заданием.

Порядок, в котором вы будете читать эту книгу, большого значения не имеет. Большинство читателей, вероятно, будут читать книгу выборочно, находя материал, нужный для выполнения той или иной задачи. Данное издание содержит множество примеров, которые помогут вам быстро найти формулу, имеющую отношение к интересующей вас теме. Если в данный момент перед вами стоит конкретная задача, вам следует изучить предметный указатель. Так вы узнаете, может ли эта книга быть вам полезна в вашей ситуации.

Power Utility Pak

В конце книги вы найдете купон, предъявив который вы сможете приобрести мой непревзойденный пакет Power Utility Pak со скидкой. Power Utility Pak — это коллекция очень полезных утилит к Excel и множество новых функций, использующихся в рабочей книге. Этот продукт был разработан исключительно с помощью VBA.

Этот купон вы можете использовать и для покупки полного исходного кода VBA приложения по номинальной цене. Разобравшись в коде, вы сможете овладеть приемами программирования, которые могут пригодиться вам в работе.



Power Utility Pak требует использования Excel 97 для Windows или более поздней версии.

Последнюю версию Power Utility Pak вы можете загрузить, посетив мой Web-узел:
<http://www.j-walk.com/ss>

Материалы на Web-узле

Все рабочие книги, описанные в этой книге, вы найдете на Web-узле издательства по адресу <http://www.dialektika.com>. На нем вы также найдете вспомогательные утилиты и надстройки, созданные и подобранные автором для расширения функциональных возможностей программы Excel.

Обратная связь

Я очень заинтересован в ваших отзывах о моих книгах. Обратную связь лучше всего поддерживать с помощью электронной почты. Ваши замечания и предложения направляйте по адресу:
author@j-walk.com

К сожалению, я не смогу ответить на ваши вопросы. Советую выносить возникающие вопросы в сетевые конференции по теме Excel. Скорее всего вам помогут (см. приложение Г).

Путешествуя во Всемирной сети, не забудьте заглянуть и на мой Web-узел:

<http://www.j-walk.com/ss/>

А теперь, не трата времени даром, перевернем страницу и начнем повышать свою квалификацию.

Благодарности

Я благодарен всем, кто приобрел первое издание моей книги. Особую благодарность выражаю тем читателям, которые нашли время для того, чтобы высказать свои замечания и предложения. Многие предложения читателей были учтены мною в настоящем издании.

Я хочу сказать спасибо Норману Хакеру (Norman Harker), старшему преподавателю Сиднейского Университета (Австралия), специализирующемуся в области недвижимости. Прочитав первое издание книги, Норман заметил, что глава, посвященная финансовым формулам, не вполне отвечает уровню книги. В результате эта глава была дополнена, причем большая часть материала глав 11-13 была предоставлена Норманом.

Хочу особо отметить превосходные навыки технического редактирования Билла Мэнвилла (Bill Manville). С Биллом мы работаем над вторым проектом, и я уверен, что он — один из лучших специалистов в сфере научного редактирования. Он исправил мои ошибки, внес множество ценных предложений и отредактировал десятки формул, которые теперь функционируют эффективнее.

И наконец, спасибо ребятам из Hungry Minds за то, что издали эту книгу. Несомненно, она не похожа на другие издания по Excel, и потому опубликовать ее было рискованным предприятием. Но риск был оправдан. Теперь, когда вышло второе издание моей книги, я в этом не сомневаюсь. Хочу выразить особую благодарность Сюзан Кристоферсен (Susan Christophersen), литературному редактору моего проекта. Она очень помогла мне в работе.

Часть I

Основные сведения

ГЛАВА 1

“Коротко об Excel”

ГЛАВА 2

“Основные сведения о формулах”

ГЛАВА 3

“Работаем с именами”

Коротко об Excel

В этой главе...

- ◆ Краткая история Excel
- ◆ Концепция объектной модели
- ◆ Работа с книгами
- ◆ Пользовательский интерфейс Excel
- ◆ Форматирование ячеек
- ◆ Формулы и функции электронных таблиц
- ◆ Объекты графического слоя
- ◆ Настройка Excel в соответствии с требованиями пользователя
- ◆ Средства анализа
- ◆ Уровни защиты
- ◆ Резюме

Microsoft Excel считается “самой лучшей из всех прикладных программ, когда-либо написанных для Windows”. Вы можете соглашаться или не соглашаться с этим утверждением. Однако никто не может отрицать тот факт, что Excel — одно из самых *старых* приложений для Windows, которое за эти годы не раз перевоплощалось, приобретая все более и более современный вид. Внешний вид последней версии этой программы — Excel 2002 — лишь отдаленно напоминает вид исходной версии (которая, между прочим, была написана для Macintosh). Тем не менее, несмотря на значительное расширение возможностей программы, многие основные элементы Excel оставались неизменными на протяжении многих лет.

Эта глава представляет собой краткий обзор основных характеристик, которыми обладают современные версии Excel, причем особое внимание уделено Excel 2002. Материал этой главы служит основой для освоения последующих глав. Эта глава особенно важна для тех пользователей, которые уже имеют опыт работы с программными средствами для создания электронных таблиц и хотят освоить Excel. Например, пользователям Lotus 1-2-3 необходимо лишь немного помочь, чтобы научить мыслить категориями Excel.



Если вы уже имеете достаточный опыт работы в Excel, то можете пропустить или бегло просмотреть эту главу.

Краткая история Excel

Вы, наверное, не ожидали обнаружить в этой книге исторический экскурс. Но эта информация может вас заинтересовать. По крайней мере, у вас будет возможность блеснуть эрудицией перед коллегами.

Электронная таблица — вещь непростая, хотя мы воспринимаем существование подобных приложений как нечто само собой разумеющееся. В те времена, когда подобных программ не существовало, люди использовали громоздкие вычислительные машины или калькуляторы, часами работая над заданиями, выполнение которых сегодня занимает всего несколько минут.

Сначала была VisiCalc

В конце 70-х годов, когда люди еще не знали, что такое персональный компьютер в офисе, Ден Бриклин (Dan Bricklin) и Боб Фрэнкстон (Bob Frankston) создали VisiCalc — первую в мире электронную таблицу. Программа VisiCalc была написана для компьютеров типа Apple II, которые по сравнению с современными машинами кажутся просто игрушечными. Очень скоро VisiCalc стала известной и особо дальновидные компании приобретали компьютеры Apple II с единственной целью — планировать бюджет с помощью программы VisiCalc. Поэтому многие считают, что своим успехом в то время компьютеры Apple II в значительной мере обязаны VisiCalc.

Потом появился Lotus

В 1982 году компания IBM громко заявила о себе, способствуя в дальнейшем утверждению персональных компьютеров на мировом рынке. Не тратя времени даром, VisiCorp адаптировала VisiCalc к новой аппаратной среде. Стремясь превзойти успех VisiCalc, небольшая группа программистов Кембриджа (штат Массачусетс) решила создать более совершенную программу управления электронными таблицами. Объединившись в небольшую компанию во главе с Митчем Капором (Mitch Kapor) и Джонатаном Заксом (Jonathon Sachs), эти энтузиасты создали новый программный продукт. Кроме того, впервые в истории создания программных средств была проведена широкомасштабная рекламная компания по продвижению нового продукта на рынок. Lotus, появившийся в 1983 году, пользовался неизменным успехом в течение многих лет. Несмотря на высокую цену — \$495 (да, в те времена люди действительно платили за программу такие деньги), новая программа быстро превзошла VisiCalc по объему продаж. Долгое время Lotus имел одни из самых высоких показателей торговой активности. Возможно, Lotus 1-2-3 — самая популярная прикладная программа из всех существующих.

Программные продукты компании Microsoft

Не многие знают о том, что компания Microsoft выпустила свой первый процессор электронных таблиц в начале 80-х. Первый программный продукт такого типа — MultiPlan — вышел в 1982 году. MultiPlan, созданный для компьютеров с операционной системой CP/M, был в последствии адаптирован и для таких платформ, как Apple II, Apple III, XENIX и MS-DOS. При создании программы MultiPlan не были соблюдены общепринятые стандарты пользовательского интерфейса. Кроме того, научиться работать с MultiPlan было нелегко, да и пользоваться неудобно. Поэтому, программа не стала популярной в Соединенных Штатах. И не удивительно, что Lotus 1-2-3 оставил своего конкурента MultiPlan далеко позади.

В некоторой степени MultiPlan можно считать предшественником Excel. Выпущенный для Macintosh в 1985 году, Excel, подобно другим приложениям Mac, имел графический интерфейс (в отличие от текстового MultiPlan). Первая версия Excel для Windows (названная Excel 2, поскольку первая версия была разработана для Macintosh) вышла в свет в ноябре 1987 года. Excel получил широкое применение не сразу. Однако популярность новой программы постепенно возрастала с распространением Windows. Вскоре появилась версия Lotus 1-2-3 для Windows. Еще одним конкурентом для Excel стал Quattro Pro — программа, созданная компанией Borland International, которая была продана Novell, а затем перешла в собственность Corel (теперешнего владельца Quattro Pro).

Версии Excel

Excel 2002 — это на самом деле Excel 10. Вы, конечно, думаете, что это — десятая версия программы. Подумайте хорошенько. Microsoft — очень успешная компания, однако у них всегда были проблемы с нумерацией программных продуктов. В действительности, Excel 2002 — восьмая версия Excel для Windows. В следующих разделах я вкратце расскажу вам об основных версиях Excel для Windows.

Excel 2

Исходная версия Excel для Windows — Excel 2 — появилась в конце 1987 года. Эта версия программы носила название Excel 2, поскольку первая версия была разработана для Macintosh. В то время Windows еще не была широко распространена. Поэтому к Excel прилагалась *оперативная* версия Windows — операционная система, обладавшая функциями, достаточными для работы в Excel. По сегодняшним стандартам эта версия Excel кажется недоработанной. Обратите внимание на рис. 1.1.

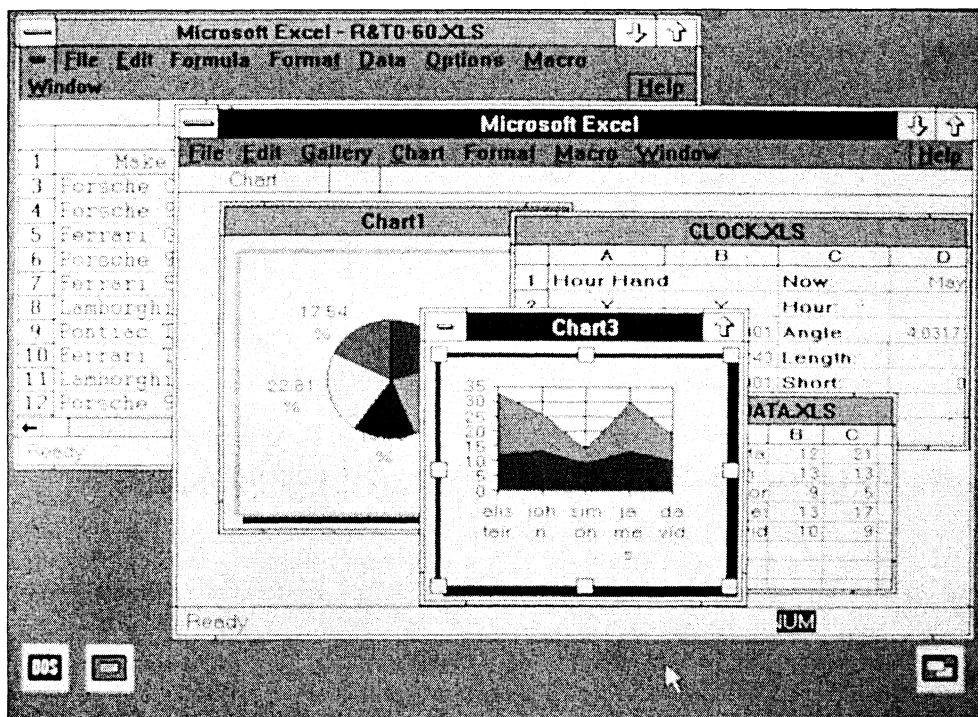


Рис. 1.1. Исходная версия Excel 2 для Windows. С тех пор программа претерпела значительные изменения (фотография предоставлена Microsoft Corporation)

Excel 3

В 1990 году компания Microsoft выпустила Excel 3 для Windows. Эта версия обладала более совершенными инструментами и внешним видом. В Excel 3 появились панели инструментов, средства рисования, режим структуры рабочей книги, надстройки, трехмерные диаграммы, функция совместного редактирования документов и многое другое.

Excel 4

Excel 4 вышла в свет весной 1992. Появление этой версии программы оказало значительное влияние на рост популярности Windows. Excel 4 обладала множеством новых функций. Кроме того, эта версия была значительно удобнее в использовании, и теперь начинающие пользователи могли достаточно быстро освоить незнакомую для них программу.

Excel 5

В начале 1994 года на рынке появилась Excel 5. В этой версии было огромное количество новых средств, включая многолистные книги и новый макроязык Visual Basic for Application (VBA). Как и предшествующая версия, Excel 5 получала наилучшие отзывы во всех отраслевых изданиях.

Excel 95

Excel 95 (также известная как Excel 7) выпущена летом 1995 года. Внешне эта версия напоминала предыдущую (в Excel 95 появилось лишь несколько новых средств). Однако появление этой версии все же имело большое значение, поскольку в Excel 95 впервые был использован более современный 32-битовый код. В Excel 95 и Excel 5 используется один и тот же формат файлов.

Excel 97

Excel 97 (также известная как Excel 8) значительно усовершенствована по сравнению с предыдущими версиями. Изменился внешний вид панелей инструментов и меню, справочная система теперь организована на качественно новом уровне, количество строк рабочей книги было увеличено в четыре раза. Если вы занимаетесь программированием на макроуровне, то, вероятно, заметили, что среда программирования Excel (VBA) значительно усовершенствована. В Excel 97 появился новый формат файлов.

Excel 2000

Excel 2000 (также известная как Excel 9) появилась в июне 1999 года. Эта версия характеризовалась незначительным расширением возможностей. Немаловажным преимуществом новой версии стала возможность использования HTML в качестве универсального формата файлов. В Excel 2000 конечно же поддерживался и стандартный двоичный формат файлов, совместимый с Excel 97.

Excel 2002

Самую современную версию программы Excel 2002 (также известную как Excel 10) выпустили в июне 2001 года. Она составляет часть пакета Microsoft Office XP. В этой версии присутствует несколько новых функций, предназначенных прежде всего для начинающих пользователей Excel. Основным нововведением Excel 2002 является возможность сохранения результатов работы в случае аварийного завершения программы и восстановления файлов поврежденной рабочей книги, над которой вы когда-то работали. В современной версии Excel появляется также фоновая проверка формул, а также новые инструментальные средства отладки. Обе эти функции напрямую связаны с темой данной книги.

Многие из этих версий Excel имели несколько выпусков. Например, компания Microsoft создала два сервисных пакета для Excel 97 (SR-1 и SR-2). Эти выпуски помогли решить многие проблемы, возникшие при эксплуатации рассматриваемого приложения.

Концепция объектной модели

Если вы не первый день знакомы с компьютером, вам, вероятно, приходилось встречать термин *объектно-ориентированное программирование*. Под словом *объект* подразумевают элемент программы, которым программист может манипулировать. Работая в Excel, иногда удобно мыслить именно категориями объектов. При этом совсем не обязательно быть программистом. Объектно-ориентированный подход поможет вам увидеть рассматриваемые элементы в перспективе.

Excel включает в себя следующие объекты:

- ◆ программу Excel;
- ◆ рабочую книгу Excel;
- ◆ электронную таблицу в рабочей книге;
- ◆ диапазоны в таблицах;
- ◆ кнопку в таблице;
- ◆ элемент управления типа список на пользовательской форме;
- ◆ лист диаграммы;
- ◆ диаграмму на листе;
- ◆ ряды диаграммы.

Обратите внимание на *иерархию объектов*. Объект Excel содержит объекты рабочих книг, которые содержат объекты таблиц. Объекты таблиц в свою очередь содержат объекты диапазонов. Эта иерархия называется *объектной моделью* Excel. Другие программные продукты Microsoft имеют другие объектные модели. Концепция объектной модели имеет большое значение при разработке макросов VBA. Даже если вы не собираетесь работать с макросами, навыки мышления на уровне объектов могут вам пригодиться.

Работа с книгами

Одним из основных объектов Excel является *книга*. Все, что вы делаете в Excel происходит в книге, которая хранится в файле с расширением `.xls`.

Хранение листов модулей VBA?

В Excel 5 и Excel 95 модуль VBA располагался на отдельном листе рабочей книги. Как вы, вероятно, знаете, в модуле VBA содержится код VBA. В Excel 97 и более современных версиях модули VBA все еще хранятся в книгах, однако они уже не располагаются на отдельном листе. С модулями VBA теперь можно работать в редакторе Visual Basic (VB Editor). Если вы хотите просмотреть или отредактировать модуль VBA, вызовите VB Editor нажатием комбинации клавиш `<Alt+F11>`. Подробную информацию о VBA вы найдете в части VI этой книги.



Начиная с версии Excel 2000, вы можете использовать HTML в качестве универсального формата файлов. В таких файлах должно храниться очень большое количество информации, необходимое для восстановления рабочей книги. Поэтому файлы HTML, созданные в Excel, могут показаться вам слишком "раздутыми". Следовательно, сохранять файл в формате HTML стоит только в том случае, если это действительно необходимо. Предпочтительнее использовать обычный для Excel формат файлов.

Книга Excel может содержать любое количество листов (ограниченное только объемом памяти). Существует четыре типа листов:

- ◆ Таблицы
- ◆ Листы диаграмм
- ◆ Листы макросов XLM (устаревшие, но все еще поддерживаемые современными версиями)
- ◆ Листы диалоговых окон (устаревшие, но все еще поддерживаемые современными версиями)

Вы можете открыть любое количество книг (каждую в отдельном окне), но *активной книгой* в данный момент может быть только одна из них. Соответственно, только один лист книги может быть *активным листом*. Для того чтобы активизировать другой лист, щелкните на соответствующей закладке, которая расположена в нижней части окна, или нажмите <Ctrl+PgUp> (чтобы активизировать следующий лист) или <Ctrl+PgDn> (чтобы активизировать предыдущий лист). Если вы хотите изменить имя листа, щелкните на соответствующей закладке дважды и введите новое название. Щелкнув на закладке правой кнопкой мыши, вы вызовете контекстное меню, с помощью которого можно осуществлять некоторые дополнительные операции с листами.

Вы можете скрыть окно книги, применив команду **Окно⇒Скрыть**. Скрытое окно книги остается открытым, но невидимым. Одна и та же книга может отображаться в нескольких окнах. Для этого выберите **Окно⇒Новое**. В новом окне будет отображен другой лист книги.

Таблицы

Наиболее распространенный тип листа — это таблица. Ее мы чаще всего подразумеваем, говоря об электронных таблицах. Каждая таблица Excel имеет 256 столбцов и 65 536 строк. Сразу ответчу на вопрос, который возникает у всех пользователей, — количество столбцов и строк строго фиксировано. Несмотря на многочисленные просьбы, Microsoft отказывается увеличить количество столбцов и строк рабочей книги. Вы можете скрыть лишние столбцы и строки, но увеличить их количество невозможно.



Электронные таблицы версий программы, предшествующих Excel 97, поддерживают только 16 384 строки.

Основное преимущество использования нескольких рабочих листов состоит не в том, чтобы предоставить на ваше распоряжение больше ячеек, а скорее в повышении эффективности организации рабочего процесса. В те далекие времена, когда файл электронной таблицы состоял из одного листа, разработчикам приходилось тратить много времени и сил на то, чтобы должным образом разместить информацию в ней. Сегодня вы можете хранить информацию в любом количестве листов и в любой момент получать доступ к этой информации.

Вы полностью контролируете ширину столбцов и высоту строк. Кроме того, вы можете скрыть столбцы и строки (а также целые таблицы). Содержимое ячеек может отображаться вертикально или под углом, более того, последнее можно перемещать так, что оно будет располагаться в нескольких строках.

Размеры таблицы

Давайте подумаем, какие же размеры имеет таблица. Умножим 256 на 65 536. Итак, таблица состоит из 16 777 216 ячеек. Напомню, что такое количество ячеек содержит лишь одна таблица, а рабочая книга может состоять из нескольких таблиц.

Если вы работаете в стандартном режиме представления видеоизображений VGA, в котором ширина столбцов и высота строк заданы по умолчанию, вы можете одновременно видеть 9 столбцов и 18 строк (или же 162 ячейки). Это составляет менее 0,001 процента полной таблицы. Иными словами, в одной таблице могло бы уместиться почти 104 000 экранов VGA.

Если бы вы вводили в каждую ячейку таблицы по одному символу, причем делали бы это достаточно быстро — заполняя одну ячейку в секунду —, вам пришлось бы непрерывно работать около 194 дней для того, чтобы заполнить всю таблицу. Для того чтобы распечатать результаты вашего труда, вам потребовалось бы 36 000 листов бумаги — стопка высотой более двух метров.



По умолчанию каждая новая рабочая книга состоит из трех таблиц. Добавить новую таблицу вы можете в любой момент, поэтому начинать новую книгу с трех таблиц нет необходимости. Вы можете изменить значение по умолчанию таким образом, чтобы каждая новая книга состояла из одной таблицы. Для этого выберите команду Сервис⇒Параметры, перейдите на вкладку Общие и измените опцию Листов в новой книге для новой книги.

Листы диаграмм

Лист диаграмм обычно включает в себя одну диаграмму. Многие пользователи предпочитают использовать вместо листов диаграмм “встроенные диаграммы”, которые хранятся на графическом слое обычного рабочего листа с данными. Разумеется, вам вовсе не обязательно использовать листы диаграмм. Однако с их помощью проще распечатать диаграмму на отдельной странице, что особенно важно при подготовке презентаций. Вложенные диаграммы (или *плавающие диаграммы*) описаны далее в этой главе.

Листы макросов XLM

Лист макросов XLM (также известный как лист макросов MS Excel 4) на самом деле представляет собой таблицу, но некоторые значения по умолчанию такого листа отличаются. В частности, лист макросов XLM отображает формулы, а не результаты применения этих формул. Кроме того, столбец листа макросов по умолчанию имеет большую ширину, чем столбец обычной таблицы.

Название этого типа листов говорит о том, что лист макросов XLM должен содержать макрос XLM. Как вы, наверное, знаете, система макросов XLM — пережиток предыдущих версий Excel (версии Excel 4.0 и предшествующих ей). Excel 2002 поддерживает макросы XLM из соображений совместимости, однако эта версия приложения уже не предоставляет возможности записывать макросы XLM. Эта книга не содержит информацию о системе макросов XLM. Пристальное внимание здесь уделено более мощной системе макросов VBA.

Листы диалоговых окон

В Excel 5 и Excel 95 вы можете создать пользовательское диалоговое окно, вставив в книгу специальный лист. Когда вы открываете книгу, содержащую лист диалогового окна Excel 5/95, этот лист представляется как один из листов книги. Excel 97 и более позд-

ние версии программы все еще поддерживают листы диалоговых окон. Однако в современных версиях существует хорошая альтернатива — пользовательские формы. С пользовательскими формами можно работать в VB Editor.



Если из соображений совместимости вам необходимо вставить лист диалогового окна Excel 5/95 в документ современной версии Excel, в меню Вставка вы не найдете для этого подходящей команды. Лист диалогового окна Excel 5/95 можно вставить следующим образом: щелкните на закладке любого листа правой кнопкой мыши и выберите Добавить в контекстном меню. Затем в диалоговом окне Вставка выберите Окно диалога Excel 5.0.

Пользовательский интерфейс Excel

Пользовательский интерфейс (UI) — это совокупность средств, с помощью которых конечный пользователь общается с программой. Пользовательский интерфейс состоит из таких элементов, как меню, диалоговые окна, панели инструментов, комбинации клавиш, а также функций, обеспечивающих операции типа “перетащить и оставить”. В основном в Excel используются стандартные для Windows элементы интерфейса.

Меню

Начиная с Excel 97, пользовательский интерфейс Excel начал отличаться от стандартного для Windows, поскольку в этой версии появились нестандартные меню. Меню в Excel 2000 и Excel 97 — это те же панели инструментов — их выдают значки, представляющие некоторые отдельные элементы.

Система меню Excel достаточно проста. В программе существуют две строки меню: одна — для активного рабочего листа, другая — для активного листа диаграмм или вложенной диаграммы. В соответствии с принятыми в Windows условными обозначениями команды меню, которые в текущий момент недоступны для использования, “затемнены”. Команды, которые приводят к появлению диалогового окна, сопровождаются троеточием. Рядом с некоторыми командами приводятся комбинации клавиш (например, в меню Правка приведена комбинация клавиш <Ctrl+Z> в качестве альтернативы команд Правка⇒Отменить).

Несколько элементов меню являются вложенными, т.е. таковыми, которые открывают подменю с дополнительными командами Правка⇒Заполнить (например, представляет вложенное меню). О том, что данное меню — вложенное, свидетельствует небольшая стрелка справа от названия элемента меню.



Пользователь или разработчик может настроить систему меню по своему усмотрению. Для этого выберите Вид⇒Панели инструментов⇒Настройка. Не забывайте о том, что внесенные вами изменения носят постоянный характер и не будут отменены при перезапуске программы. Возможности редактора меню Excel 5 и Excel 95, который отсутствует в версиях начиная с Excel 97, коренным образом отличаются от возможностей команд современных версий программы.

Контекстные меню

В Excel присутствуют десятки контекстных меню вызова команд. Эти меню появляются, когда пользователь щелкает правой кнопкой мыши, предварительно выделив один или несколько объектов. Такие меню чувствительны к контексту. Иными словами, то,

какое меню появится, зависит от положения указателя мыши в момент щелчка правой кнопкой. Вы можете навести указатель на все, что угодно: ячейку, границу столбца или строки, строку заголовка рабочей книги, панель инструментов и так далее.

Смарт-тэги

Смарт-тэг — это небольшой значок, который автоматически появляется на рабочем листе. При щелчке на нем появляется несколько опций, которые можно выбрать с помощью мыши.



Смарт-тэги доступны только в Excel 2002.

Например, когда вы копируете и вставляете некоторый диапазон ячеек, Excel автоматически отображает смарт-тэг, и располагает его ниже вставляемого диапазона (рис. 1.2). В Excel 2002 представлены и другие типы смарт-тэгов. Кроме того, к ним можно добавить смарт-тэги сторонних производителей.

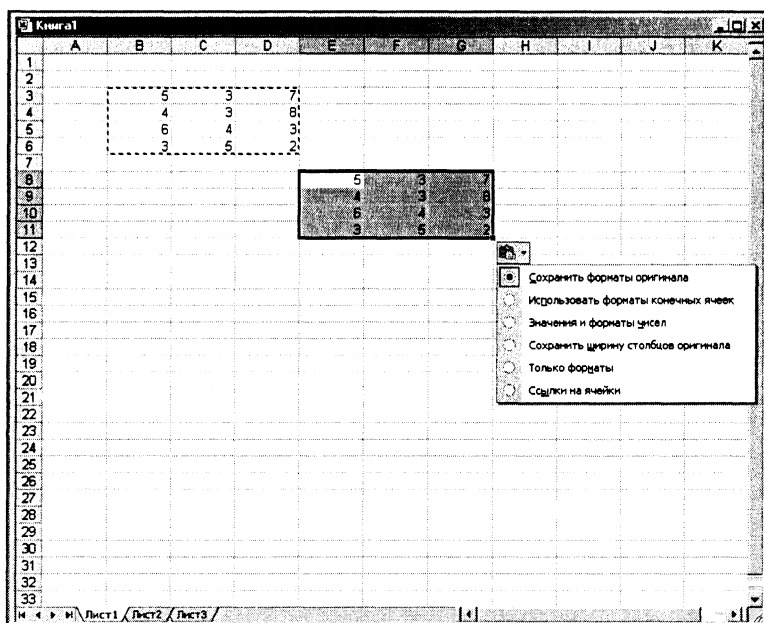


Рис. 1.2. Этот смарт-тэг появляется, когда вы вставляете скопированный ранее диапазон ячеек

Диалоговые окна

Большинство команд меню программы Excel имеют соответствующие диалоговые окна, с помощью которых пользователь может уточнить, что именно он намерен сделать. Диалоговые окна функционируют, в основном, по одному и тому же принципу. В Excel некоторые диалоговые окна состоят из нескольких вкладок. Вкладки позволяют одному окну функцио-

нирывать в качестве нескольких диалоговых окон. Диалоговое окно с вкладками предоставляет доступ сразу ко многим опциям и при этом не дает пользователю запутаться в многочисленных вариантах выбора. Примером такого диалогового окна может служить Параметры (рис. 1.3). Для вызова окна выберите Сервис⇒Параметры.

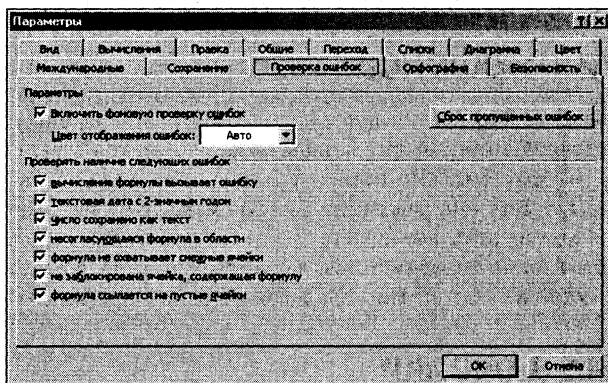


Рис. 1.3. Диалоговое окно *Параметры* содержит несколько вкладок

Панели инструментов

Excel 2002 имеет 54 панели инструментов (включая две панели, которые содержат строки меню). Обычно панели инструментов появляются автоматически по мере необходимости. Например, после выделения диаграммы появляется панель инструментов *Диаграммы*.

Вы можете прикрепить панель инструментов к любой из границ окна, либо оставить ее плавающей. По умолчанию в Excel отображены две панели инструментов: *Стандартная* и *Форматирование*. Эти панели находятся непосредственно под строкой меню.

Перетаскивание данных

Функция “перетаскивания” данных графического интерфейса программы позволяет свободно перемещать объекты, расположенные на рабочем листе. Нажатие клавиши <Ctrl> при перемещении объекта приводит к дублированию этого объекта.

Excel позволяет “перетаскивать” ячейки и диапазоны. Нажатие клавиши <Ctrl> при перемещении ячейки или диапазона приводит к дублированию выделенного диапазона.



Функция “перетаскивания” ячеек дополнительная. Вы можете отменить ее на вкладке *Правка* диалогового окна *Параметры*.

Комбинации клавиш

Excel предоставляет возможность использования целого ряда комбинаций клавиш для выполнения тех или иных команд. Например, скопировать выделенные элементы можно, нажав <Ctrl+C>. Если вы — начинающий пользователь Excel или же просто хотите научиться

работать эффективнее, воспользуйтесь справочной системой (ищите раздел *Сочетания клавиш*). Файл справочной системы содержит полезные сочетания клавиш, необходимые для быстрого выполнения команд в таблице.

Настройка отображения дополнительных элементов интерфейса

Excel дает пользователю возможность по своему усмотрению решать, какие элементы интерфейса (строка состояния, строка формул, панели инструментов и т.д.) следует отображать на экране. Например, выбрав Вид⇒Во весь экран, вы избавитесь от всех дополнительных элементов, что позволит вам одновременно видеть максимум рабочего листа. Останется только строка меню. Кроме того, вы можете задать те элементы, которые должны отображаться в окне рабочего листа (например, вы можете скрыть полосы прокрутки или линии сетки). Сделать это можно на вкладке Вид диалогового окна Параметры.

Ввод данных в ячейки

Данные в ячейках Excel задаются достаточно просто. Программа интерпретирует каждый вводимый элемент как:

- ◆ Значение (включая дату и время)
- ◆ Текст
- ◆ Булево значение (TRUE (“истина”) FALSE (“лож”))
- ◆ Формулу

Формулы всегда начинаются со знака равенства (=). Excel воспринимает и символы, привычные для пользователей Lotus 1-2-3, такие, как “коммерческое at” (@), знаки “плюс” (+) и “минус” (-). Все эти знаки могут быть первым символом формулы. Программа автоматически преобразует вводимые элементы после того, как вы нажмете клавишу <Enter>.

Выделение ячеек и других объектов

В целом, выделение объектов в Excel происходит так же, как и в других приложениях Windows. Выделить диапазон ячеек можно с помощью клавиатуры (нажимая клавишу <Shift> и клавиши с изображением стрелок) или мыши (щелкнув кнопкой мыши и перемещая указатель). Для того чтобы выделить большой диапазон, щелкните в любом из углов диапазона, затем нажмите клавишу <Shift> и щелкните на ячейке в противоположном углу диапазона.

Для выделения всей таблицы используйте комбинацию клавиш <Ctrl+*> (<Ctrl+звездочка>). При выделении большого диапазона используйте <Ctrl+.> (<Ctrl+точка>). Это дает возможность указывать четыре угла диапазона.

Для выделения объекта, находящегося на рабочем листе, просто наведите на него указатель и щелкните мышью. Единственным исключением может быть только объект, к которому добавлен макрос. В этом случае щелчок на объекте вызывает макрос. При выделении нескольких объектов или независимых ячеек удерживайте нажатой клавишу <Ctrl>.

Советы по вводу данных в ячейки

Ниже приведен список советов, которые помогут пользователям, имеющим опыт работы с другими процессорами электронных таблиц, научиться правильно вводить данные в ячейки Excel.

- ◆ Если вы хотите вводить данные и при этом не пользоваться клавишами со стрелками, активизируйте опцию Переход к другой ячейке после ввода на вкладке Правка диалогового окна Параметры (вызвать которое можно командой Сервис⇒Параметры). Вы можете задать направление перехода к другой ячейке.
- ◆ Иногда удобно выделить диапазон ячеек перед тем, как вводить данные. Если вы поступите таким образом, то сможете перемещаться в пределах выделенного диапазона с помощью клавиши табуляции.
- ◆ Для того чтобы ввести одни и те же данные во все ячейки диапазона, выделите диапазон, введите данные в активную ячейку, а затем нажмите комбинацию клавиш <Ctrl+Enter>.
- ◆ Если необходимо скопировать содержимое активной ячейки во все ячейки выделенного диапазона, нажмите <F2> и затем <Ctrl+Enter>.
- ◆ Для того чтобы все последующие значения диапазона прирастить на единицу, нажмите <Ctrl> во время перетаскивания маркера выделения диапазона, расположенного в нижнем правом углу текущей ячейки.
- ◆ Если вы хотите создать свой собственный самозаполняющийся список, сделать это вы сможете на вкладке Списки диалогового окна Параметры.
- ◆ Для того чтобы скопировать ячейки без приращения, переместите маркер выделения диапазона в правом нижнем углу ячейки, или нажмите <Ctrl+D> для копирования в соседнюю ячейку снизу, или <Ctrl+R> — справа.
- ◆ Если необходимо, чтобы сделать вводимый вами более удобочитаемым, можно использовать клавишу возврата каретки. Для того чтобы ввести в ячейку символ возврата каретки, нажмите <Alt+Enter>. Таким образом содержимое ячейки будет переноситься на новую строку.
- ◆ Для того чтобы задать дробное число, введите 0, пробел, а затем дробь (с использованием косой черты). Excel преобразует значение в формат дробного числа.
- ◆ Для того чтобы использовать в ячейке денежный формат, введите соответствующий символ валюты перед или после числового значения.
- ◆ Если необходимо задать величину в процентах, введите символ процента после числового значения. Вы можете также использовать принятый в вашей стране разделитель разрядов (например, 123,434).
- ◆ Для того чтобы ввести текущую дату, используйте комбинацию клавиш <Ctrl+>. Если нужно ввести время, нажмите <Ctrl+Shift+>.
- ◆ Если вы хотите, чтобы ячейка или диапазон принимали только данные определенного типа (или значения, находящиеся в определенных пределах), примените команду Данные⇒Проверка.

Форматирование ячеек

В Excel существует два типа форматирования ячеек — числовое форматирование и стилевое форматирование.

Числовое форматирование

От *числового форматирования* зависит форма, в которой значения будут представлены в ячейке. В Excel существует целый ряд стандартных числовых форматов. Кроме того, вы мо-

жете создавать и собственные числовые форматы на вкладке Число диалогового окна Формат ячеек. Для вызова этого диалогового окна выполните команду Формат⇒Ячейки.

В зависимости от вводимого элемента, Excel иногда применяет числовое форматирование автоматически. Например, если перед числовым значением вы введете символ валюты (например, символ доллара), программа применит денежный числовой формат.



Дополнительную информацию о том, как создавать свои собственные числовые форматы, вы найдете в приложении В.

Числовой формат не влияет на само значение, которое находится в ячейке. Например, ячейка содержит значение 3,14159. Если вы зададите такой формат, при котором отображаются только два десятичных разряда, в ячейке окажется следующая запись — 3,14. Однако в формуле будет использовано действительное значение ячейки (3,14159).

Стилевое форматирование

Стилевое форматирование используется исключительно для придания содержимому ячеек желаемого внешнего вида (цвета, начертания, границы и т.д.). Диалоговое окно Формат ячеек (рис. 1.4) поможет вам отформатировать ячейки и отдельные диапазоны.

Некоторые кнопки на панели инструментов предоставляют доступ к опциям форматирования, которые могут быть применены к ячейкам, графическим объектам или диаграммам. Например, с помощью кнопки Цвет заливки вы можете изменить цвет фона ячейки, цвет заливки рисованного текстового поля или цвет одной из областей диаграммы. Все опции форматирования вы найдете в меню Формат.

Для каждого типа объектов предусмотрено свое диалоговое окно. Вызвать соответствующее объекту диалоговое окно очень просто — выделите объект, щелкните правой кнопкой мыши, а затем выберите в контекстном меню команду **Format xxx** (где **xxx** — выделенный объект). Существует еще один способ: нажмите комбинацию клавиш **<Ctrl+1>**. Любое из описанных действий приводит к появлению диалогового окна с набором вкладок, в котором представлены все опции форматирования данного объекта.

Стоит обратить внимание и на функцию условного форматирования Excel. Этот инструмент позволяет назначить объекту определенный стиль форматирования, который проявляется только при определенных условиях. Например, вы можете назначить красный цвет заливки ячейке, в которую вводится отрицательное число.

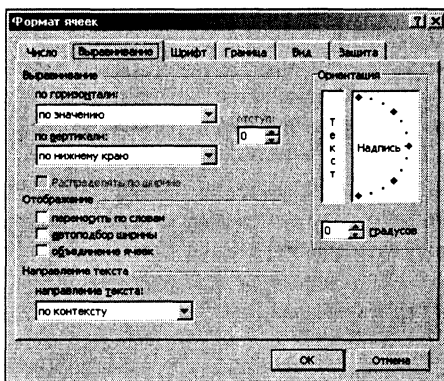


Рис. 1.4. Стилевое форматирование объектов осуществляется с помощью диалогового окна Формат ячеек



Прочитав главу 19, вы узнаете, как создавать формулы условного форматирования, которые делают эту функцию программы намного эффективнее.

Формулы и функции электронных таблиц

Формула — один из основополагающих элементов электронной таблицы. Значительным преимуществом программы Excel является возможность создания формул. А как это сделать, вы узнаете прочитав следующие главы этой книги.

Функции электронных таблиц позволяют проводить вычисления или иные операции, которые невозможно осуществить без них. Excel обладает множеством встроенных функций. Кроме того, подключив надстройку “Пакет анализа”, вы можете получить доступ ко многим дополнительным функциям.



Функции электронных таблиц подробно описаны в главе 4.

Все электронные таблицы позволяют назначать имена ячейкам и диапазонам. Работа с именами в Excel имеет свои особенности. *Имя* представляет собой идентификатор, который позволяет обращаться к ячейке, диапазону, значению или формуле. Использование имен значительно упрощает работу с формулами.



Глава 3 полностью посвящена именам.

Объекты графического слоя

Как я уже говорил, каждый рабочий лист имеет невидимый графический слой, в котором содержатся все фигуры, диаграммы, графики, карты, картинки и элементы управления (такие, как кнопки и списки). В следующих разделах я расскажу об основных объектах графического слоя.

Фигуры

Вставка автофигуры осуществляется с помощью панели инструментов Рисование. Вам предоставляется широкий выбор фигур. Разместив фигуру на рабочем листе, вы можете видоизменить ее. Для этого выделите фигуру, а затем перетащите соответствующие маркеры.

Кроме того, вы можете добавить тень, создать трехмерный текст или другие эффекты. Несколько фигур можно объединить в один графический объект, который составит единое целое.

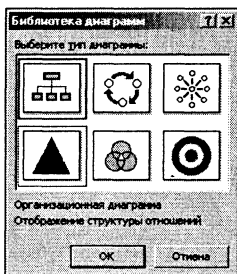


Рис. 1.5. Excel 2002 поддерживает несколько типов диаграмм

Диаграммы

Команда Вставка⇒Организационная диаграмма открывает диалоговое окно Библиотека диаграмм, показанное на рис. 1.5. В этом окне вы можете выбрать один из шести типов диаграмм, каждый из которых при желании можно изменить.



Библиотеку диаграмм вы найдете только в версии Excel 2002.

Ссылки на графические объекты

Создать *ссылку на графический объект* достаточно сложно. Для этого необходимо скопировать объект, нажать клавишу <Shift>, а затем выбрать команду Правка⇒Вставить связь с рисунком. Эта команда появляется в меню Edit только при нажатии <Shift>. Исходно команда предназначена для пользователей, которым необходимо распечатать на одном листе набор диапазонов, не связанных между собой. С помощью этой команды пользователи могли скопировать нужные диапазоны, разместить их в одном наборе и затем распечатать.

Карты

Если вы работаете с географическими данными, то, вероятно, воспользуетесь средством добавления карты на рабочий лист с помощью команды Вставка⇒Карта. К сожалению, эта функция отсутствует в Excel 2002. Карты, созданные в предыдущих версиях программы, поддерживаются в Excel 2002, однако изменить их в последней версии невозможно.

Элементы управления диалоговыми окнами

Многие элементы управления диалоговыми окнами могут размещаться на графическом слое рабочего листа. Это придает таблицам удобство и простоту в использовании, избавляет от необходимости создания собственных диалоговых окон. На рис. 1.6 изображена таблица, на графический слой которой добавлены некоторые элементы управления диалоговыми окнами.

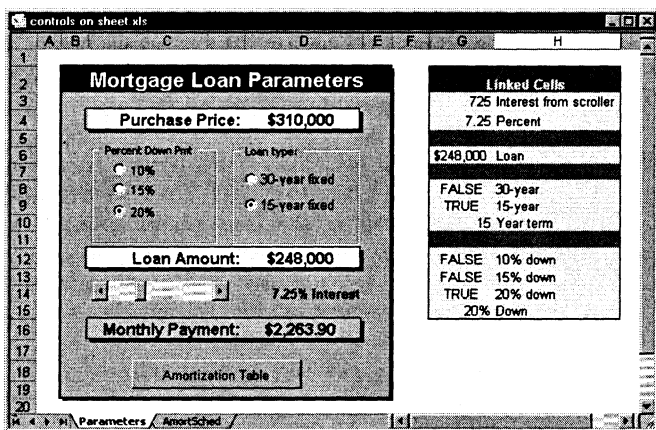


Рис. 1.6. Excel позволяет размещать элементы управления диалоговыми окнами на графическом слое рабочего листа



Элементы управления диалоговыми окнами находятся на панелях инструментов Формы или Элементы управления. Элементы управления на панели Элементы управления представляют собой элементы ActiveX, которые доступны только в Excel 97 и более современных версиях.

Диаграммы

Excel — превосходная среда для построения диаграмм. Как я уже говорил, диаграммы могут храниться на листе диаграмм или располагаться на текущем рабочем листе в качестве плавающих объектов.

В Excel любая диаграмма может изменяться, чтобы максимально соответствовать требованиям пользователя. Если диаграмма — плавающий объект, выделить его можно, расположив указатель над любым ее элементом и щелкнув мышью (если вы щелкните дважды, появится диалоговое окно форматирования диаграмм). Щелчок правой кнопкой мыши активизирует контекстное меню.

Создать плавающую диаграмму нетрудно. Выделите данные, на основе которых строится диаграмма. Создать подходящую диаграмму вам поможет мастер диаграмм.



Глава 17 содержит дополнительную информацию о диаграммах.

Настройка Excel в соответствии с требованиями пользователя

В этом разделе описаны средства, позволяющие настроить Excel в соответствии с вашими требованиями. Среди них основными являются макросы, панели инструментов и программные надстройки.

Макросы

Язык программирования VBA — мощный инструмент, позволяющий Excel выполнять такие задачи, осуществить которые иными средствами чрезвычайно трудно или невозможно. С помощью VBA создаются следующие основные типы макросов:

- ◆ макросы, автоматизирующие операции в Excel;
- ◆ макросы, функционирующие в качестве пользовательских функций, которые используются при записи формул таблиц.



Часть VI этой книги посвящена созданию пользовательских функций с помощью VBA.

Панели инструментов

Excel располагает большим количеством панелей инструментов. При желании, вы можете создать новые панели инструментов, содержащие уже существующие или новые кнопки, щелчок на которых запускает макросы.

Настроить стандартные панели инструментов или создать новые можно, выбрав команду Вид⇒Панели инструментов⇒Настройка. Вы можете написать специальный код VBA для управления панелями инструментов.

Надстройки

Надстройка — это программа, подключаемая к Excel. Программные надстройки расширяют возможности основной программы. Например, в надстройке вы можете хранить пользовательские функции таблиц. Для подключения надстройки выберите Сервис⇒Надстройки.

К Excel прилагается только несколько надстроек (включая “Пакет анализа”). Кроме прилагаемых к Excel, вы можете приобрести или загрузить большое количество иных надстроек. Одной из таких надстроек является мой пакет Power Utility Pak.



Прочитав главу 23, вы узнаете, как создавать свои собственные надстройки, содержащие пользовательские функции.

Средства анализа

Excel — большой специалист в области анализа. Большинство людей применяют электронные таблицы именно с целью проведения анализа. С помощью формул решаются многие задачи. Но в арсенале Excel есть еще целый ряд других средств, предназначенных для проведения анализа. Об этих средствах я расскажу в последующих разделах.

Доступ к базам данных

До настоящего времени большинство электронных таблиц предоставляли пользователю возможность работать только с плоскими таблицами баз данных (даже в исходной версии Lotus 1-2-3 не было альтернативы). Базы данных Excel бывают двух основных типов:

- ◆ **Табличные базы данных.** База данных хранится на рабочем листе. Это строго ограничивает ее размеры. Табличная база данных Excel может иметь не более 65 535 записей (поскольку в таблице Excel 65 536 строк, а верхняя строка предназначена для названий полей) и 256 полей (т.к. в таблице Excel 256 столбцов).
- ◆ **Внешние базы данных.** В этом случае данные хранятся в одном или нескольких файлах на диске, к которым можно обратиться в любой момент.

Когда выделена ячейка рабочего листа табличной базы данных, Excel распознает эту базу и по возможности отображает названия полей. Например, если вы примените команду Данные⇒Сортировка, программа позволит задавать параметры сортировки. Для этого выберите имена полей из раскрывающегося списка.

В Excel есть очень полезная функция — Автофильтр, которая позволяет отображать только те записи, которые используются вами в данный момент. В режиме автофильтра вы можете представлять данные, выбирая значения из раскрывающихся списков (эти списки являются вместо имен полей, когда вы выбираете Данные⇒Фильтр⇒Автофильтр). Строки, которые не соответствуют параметрам фильтра, временно скрываются. На рис. 1.7 приведен пример, иллюстрирующий работу фильтра.

Вы можете использовать и традиционный способ работы с базами данных электронных таблиц, при котором используются условные диапазоны. Для этого выполните команду Данные⇒Фильтр⇒Расширенный фильтр.

	A	B	C	D	E	F
1	SalesRep	Region	Month	Sales		
2	Amy	North	(All)	13040		
3	Amy	North	(Top 10...)	14131		
4	Amy	North	(Custom...)	14646		
5	Amy	North	Apr	12047		
6	Amy	North	Dec	14971		
7	Amy	North	Feb	14218		
8	Amy	North	Jan	15735		
9	Amy	North	Jul	13638		
10	Amy	North	Jun	15749		
11	Amy	North	Mar	14437		
12	Amy	North	May	15355		
13	Amy	North	Nov	15899		
14	Bob	North	Oct	10024		
15	Bob	North	Jan	13822		
16	Bob	North	Feb	14854		
17	Bob	North	Mar	12838		
18	Bob	North	Apr	15320		
19	Bob	North	May	14733		
20	Bob	North	Jun	11184		
21	Bob	North	Jul			

Рис. 1.7. Функция Excel Автофильтр дает возможность отображать только те записи базы данных, которые необходимы вам в данный момент



Дополнительную информацию о списках и базах данных таблиц вы найдете в главе 9.

Excel может автоматически вставлять (или удалять) формулы промежуточных сумм в таблицах, которые именуются базами данных. В программе есть возможность отдельного просмотра промежуточных сумм или любых других вспомогательных сведений.

Структура

Режим структуры таблиц особенно удобен при работе с иерархически упорядоченными данными, такими как, например, бюджетные сведения. Excel может создать подходящую структуру автоматически, исходя из формул таблицы. После того как структура создана, вы можете изменить ее размеры таким образом, чтобы обеспечить необходимый уровень детализации информации. На рис. 1.8 показан пример структуры таблицы.

	A	B	C	D	E	F	G	H	I
1	State	Jan	Feb	Mar	Q1 Total	Q2 Total	Q3 Total	Q4 Total	C
6	West Total	5,170	6,527	5,081	16,778	18,242	18,314	19,138	
7	New York	1,429	1,316	1,993	4,738	4,763	3,915	4,289	
8	New Jersey	1,735	1,406	1,224	4,365	4,316	4,051	4,526	
9	Massachusetts	1,099	1,233	1,110	3,442	4,155	4,882	5,376	
10	Florida	1,705	1,792	1,225	4,722	4,630	5,062	4,734	
11	East Total	5,968	5,747	5,552	17,267	17,864	17,910	18,925	
12	Kentucky	1,109	1,078	1,155	3,342	4,626	4,306	4,526	
13	Oklahoma	1,309	1,045	1,641	3,995	5,364	4,338	4,449	
14	Missouri	1,511	1,744	1,414	4,669	4,556	4,203	4,279	
15	Illinois	1,539	1,493	1,211	4,243	3,623	4,889	4,330	
16	Kansas	1,973	1,560	1,243	4,776	4,007	4,322	4,299	
17	Central Total	7,441	6,920	6,664	17,683	17,550	17,752	17,357	
18	Grand Total	18,579	19,194	17,297	51,728	53,656	53,976	55,420	

Рис. 1.8. Excel может автоматически вставлять промежуточные формулы и создавать структуру данных

Управление сценариями

Управление сценариями — это процесс сохранения входных величин, которые составляют определенную модель. Например, вы подготовили прогноз сбыта продукции. Этот прогноз может включать сценарии наиболее благоприятного, наименее благоприятного и наиболее вероятного случаев развития событий.

Эффективнее остальных управление сценариями осуществляет менеджер сценариев. Менеджер сценариев Excel может выполнять только самые простые задачи управления сценариями. Но использовать функцию управления сценариями Excel гораздо удобнее, чем пытаться согласовать несколько сценариев вручную.

Надстройка “Пакет анализа”

“Пакет анализа” предоставляет 19 инструментов анализа данных специального назначения (в основном статистических) и множество дополнительных функций таблиц. Эти средства позволяют Excel осуществлять статистический анализ небольшого и среднего масштаба.

Сводные таблицы

Сводные таблицы — одно из самых мощных инструментальных средств Excel. *Сводная таблица* позволяет отображать сводные данные в любой форме. Данные для сводной таблицы берутся из табличной или внешней базы данных и хранятся в специальном кэше. Это позволяет программе быстро производить пересчет данных в случае изменения сводной таблицы.



Глава 18 содержит дополнительную информацию о сводных таблицах.

Excel 2002, как наиболее современная версия программы, поддерживает функцию создания сводной диаграммы. Сводные диаграммы позволяют присоединить диаграмму к сводной таблице.

Средства проверки данных

Excel предоставляет возможность проверки данных. Проверка поможет выявить ошибки или разобраться в неизвестной таблице. Воспользоваться этой функцией можно, выбрав команду Сервис⇒Проверка формул. В версиях, предшествующих Excel 2002, — Сервис⇒Проверка.



В Excel 2002 существует функция фоновой проверки формул. Об этой и многих других средствах проверки читайте в главе 21.

Надстройка “Поиск решения”

Для выполнения особых линейных и нелинейных задач служит программа-надстройка “Поиск решения”. Она занимается разрешением вопросов, связанных с анализом ситуаций типа “что-если”, основываясь на изменяемых ячейках, целевых ячейках и, возможно, ячейках, которые содержат максимальные или минимальные данные.

Уровни защиты

В Excel существует несколько уровней защиты. Например, вы можете защитить формулы от перезаписи или изменения, защитить структуру рабочей книги, а также защитить код VBA.

Защита формулы от перезаписи

Нередко возникает необходимость защитить формулы от перезаписи или изменения. Для того чтобы защитить формулы, сделайте следующие действия:

1. Выделите ячейки, которые могут подвергнуться перезаписи.
2. Выполните команду **Формат**⇒**Ячейки** и перейдите на вкладку **Защита** диалогового окна **Формат ячеек**.
3. На вкладке **Защита** снимите флажок опции **Защищаемая ячейка**.
4. Закройте диалоговое окно **Формат ячеек**, щелкнув на кнопке **ОК**.
5. Выполните команду **Сервис**⇒**Защита**⇒**Защитить лист**. Появится диалоговое окно **Защита листа**, показанное на рис. 1.9. Если вы работаете с версией программы, предыдущей Excel 2002, это диалоговое окно будет выглядеть иначе.
6. В диалоговом окне **Защита листа** задайте пароль (не обязательно) и щелкните на кнопке **ОК**.



По умолчанию все ячейки являются защищаемыми. Но защита действует только в том случае, если защищен весь рабочий лист.



Excel 2002 располагает более гибкими функциями защиты. Когда вы защищаете рабочий лист, диалоговое окно **Защита листа** позволяет вам выбрать и те элементы, которые не должны защищаться. Например, вы можете разрешить пользователям сортировать данные или использовать автофильтр при работе с защищенным листом (в предыдущих версиях программы это было невозможно).

Вы можете скрыть формулы так, чтобы они не отображались в строке формул Excel, когда ячейка активна. Для этого выделите ячейки и выставьте флажок опции **Скрыть формулы** на вкладке **Защита** диалогового окна **Формат ячеек**.

Защита структуры рабочей книги

Защитить структуру рабочей книги — означает запретить добавление и удаление листов. Выполните команду **Сервис**⇒**Защита**⇒**Защитить книгу**. Появится диалоговое окно **Защита книги**, показанное на рис. 1.10. Убедитесь в том, что выставлен флажок опции **структура**. Если вы поставите метку и в окошке, расположенном рядом с **Windows** (окна), окно нельзя будет переместить или изменить его размеры.



Не забывайте о том, что функции защиты Excel несовершенны. Даже если вы используете пароль, защитить компоненты книги вам удастся только от рядовых пользователей. Тот, кто действительно заинтересован в том, чтобы взломать защиту вашей книги, скорее всего добьется желаемого результата с помощью существующих программ взлома паролей.

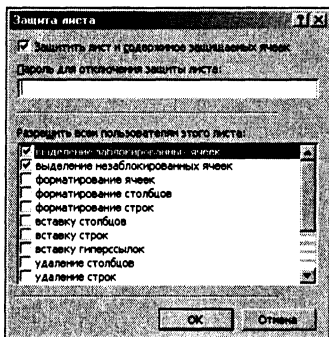


Рис. 1.9. Диалоговое окно Защита листа в Excel 2002

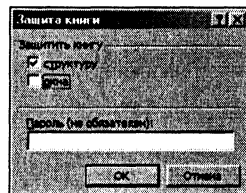


Рис. 1.10. Диалоговое окно Защита книги

Резюме

Эта глава содержит краткий обзор средств Excel и адресована в первую очередь начинающим пользователям. Следующая глава отвечает теме данной книги и содержит вводную информацию о формулах Excel.

Глава 2

Основные сведения о формулах

В этой главе...

- ◆ Ввод и редактирование формул
- ◆ Применение операторов в формулах
- ◆ Вычисление формул
- ◆ Ссылки на ячейки и диапазоны ячеек
- ◆ Создание точной копии формулы
- ◆ Преобразование формул в значения
- ◆ Скрытие формул
- ◆ Ошибки в формулах
- ◆ Работа с циклическими ссылками
- ◆ Функция подбора параметра
- ◆ Резюме

Эта глава служит введением к теме “Формулы в Excel”. Информация, содержащаяся в этой главе, в первую очередь адресована начинающим пользователям Excel. Но некоторые сведения могут быть полезны даже тем, кто уже имеет большой опыт работы с программой.

Ввод и редактирование формул

В этом разделе описаны основные элементы формул. Кроме того, здесь перечислены различные способы введения и редактирования формул.

Элементы формул

Формула, вводимая в ячейку, может состоять из пяти типов элементов:

- ◆ **Операторы.** Символы, например “+” (сложение) и “*” (умножение).
- ◆ **Ссылки на ячейки.** В эту категорию входят именованные ячейки и диапазоны, относящиеся к текущему рабочему листу, ячейкам другого листа текущей книги и даже ячейкам листа другой рабочей книги.
- ◆ **Значения или строки.** Например, 7,5 или “*Результаты на конец года*”.
- ◆ **Функции и их аргументы.** Сюда относятся функции, такие как СУММ или СРЗНАЧ и их аргументы.
- ◆ **Скобки.** Задают порядок выполнения действий в формуле.

Ввод формул

Когда вы вводите в ячейку знак равенства, программа воспринимает следующую за этим знаком запись как формулу (формулы в Excel всегда начинаются со знака равенства). В начале формулы допустимо также использование знаков “плюс” и “минус”. Однако Excel автоматически заменит любой из них на знак равенства, как только вы закончите вводить формулу.

Excel также воспринимает символы, привычные для пользователей Lotus 1-2-3, такие, как “коммерческое at” (@). Этот символ может открывать формулу, которая начинается с функции. Например, программа распознает обе приведенные ниже формулы:

=СУММ(A1:A200)

@СУММ(A1:A200)

Однако, после того как вы введете вторую формулу, программа автоматически заменит символ “коммерческое at” на знак равенства. Ввести формулу в ячейку можно двумя способами: вручную или указав ссылки на ячейки. Оба метода описаны в следующих разделах.

Ручной ввод формул

Ручной ввод формул означает, что вы просто активизируете ячейку и вводите в ней знак равенства (=), а за ним — саму формулу. Вводимые вами символы одновременно появляются в ячейке и в строке формул. При вводе формул вы, конечно же, можете использовать клавиши, предназначенные для редактирования. Закончив вводить формулу, нажмите <Enter>.



Исключение составляют формулы массивов. По окончании ввода формул массива следует нажать комбинацию клавиш <Ctrl+Shift+Enter>. О формулах массивов речь пойдет в части IV.

После того как вы нажмете <Enter>, в ячейке будет отображен результат выполнения формулы. Сама же формула будет появляться в строке формул, когда соответствующая ячейка будет активна.

Ввод формул с указанием ссылок на ячейки

Этот способ также предполагает ручной ввод некоторых элементов. Вы можете указать ссылки на ячейки, вместо того чтобы задавать их вручную. Например, чтобы ввести формулу =A1+A2 в ячейке A3, необходимо выполнить следующее:

1. Выделите ячейку A3.
2. Введите знак равенства (=). Обратите внимание, Excel отображает слово *Ввод* в левой части строки состояния.
3. Дважды нажмите на клавишу со стрелкой вверх. Нажав на эту клавишу, вы заметите вокруг ячейки рамку. Ссылка на ячейку (A1) появится в ячейке A3 и в строке формул. Обратите внимание и на слово *Укажите* в строке состояния.

Если вы предпочитаете использовать мышь, наведите указатель на ячейку A1 и щелкните мышью.

4. Введите знак плюс (+). Движущаяся рамка исчезнет, а в строке состояния снова появится *Ввод*.
5. Еще раз нажмите клавишу со стрелкой вверх. К формуле добавится A2.

Если вы предпочитаете использовать мышь, наведите указатель на ячейку A2 и щелкните кнопкой мыши.

6. Закончив ввод формулы, нажмите <Enter>. Точно так же, как и при ручном вводе формул, результат выполнения формулы отобразится в ячейке. Формула появится в строке формул, когда соответствующая ячейка будет активна.

Вы также можете навести указатель на значок с изображением “галочки”, расположенный рядом со строкой формул, и щелкнуть на нем.

Задавать ссылки на ячейки — менее утомительно, чем вводить формулу вручную. Кроме того, это помогает избежать ошибок при вводе формул.



Когда вы задаете формулу, относящуюся к другим ячейкам, ячейка, содержащая эту формулу, имеет тот же числовой формат, что и ячейки, к которым она относится. Существует только одно исключение, ссылка на первую ячейку имеет формат процентных величин.

В Excel 97 и Excel 2000 существует панель формул, которая поможет вам при вводе и редактировании формул. Для того, чтобы вызвать панель формул, щелкните на кнопке Изменить формулу, расположенной в строке формул (на этой кнопке изображен знак равенства). Панель формул позволяет обращаться к формулам вручную или применять технику задания ссылок, описанную выше.



В Excel 2002 панель формул отсутствует.

Вставка имен

В главе 3 я расскажу о том, как присваивать имя ячейке или диапазону. Если в формуле используются именованные ячейки или диапазоны, вы можете задать имя вместо адреса. Имя можно выбрать из списка, тогда программа автоматически вставит его в формулу.

Для вставки имени в формулу выберите Вставка⇒Имя⇒Вставить или нажмите клавишу <F3>. Появится диалоговое окно Вставка имени. В этом окне перечислены все имена, как показано на рис. 2.1. Выберите имя и щелкните на кнопке ОК. Вы можете навести указатель на имя и дважды щелкнуть кнопкой мыши. Excel вставит имя в формулу и закроет диалоговое окно.

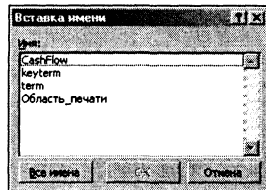


Рис. 2.1. Диалоговое окно Вставка имени позволяет вставлять имя при вводе формулы

Пробел и разрыв строки

Чаще всего формулы не имеют пробелов. Однако при вводе формул вы все же можете использовать пробелы и даже разрыв строки. Пробелы и разрывы строк упростят чтение данных и, при этом, не повлияют на результат формулы. Для того чтобы задать разрыв строки, нажмите комбинацию клавиш <Alt+Enter>. На рис. 2.2 изображена формула с использованием пробелов и разрывов строки.

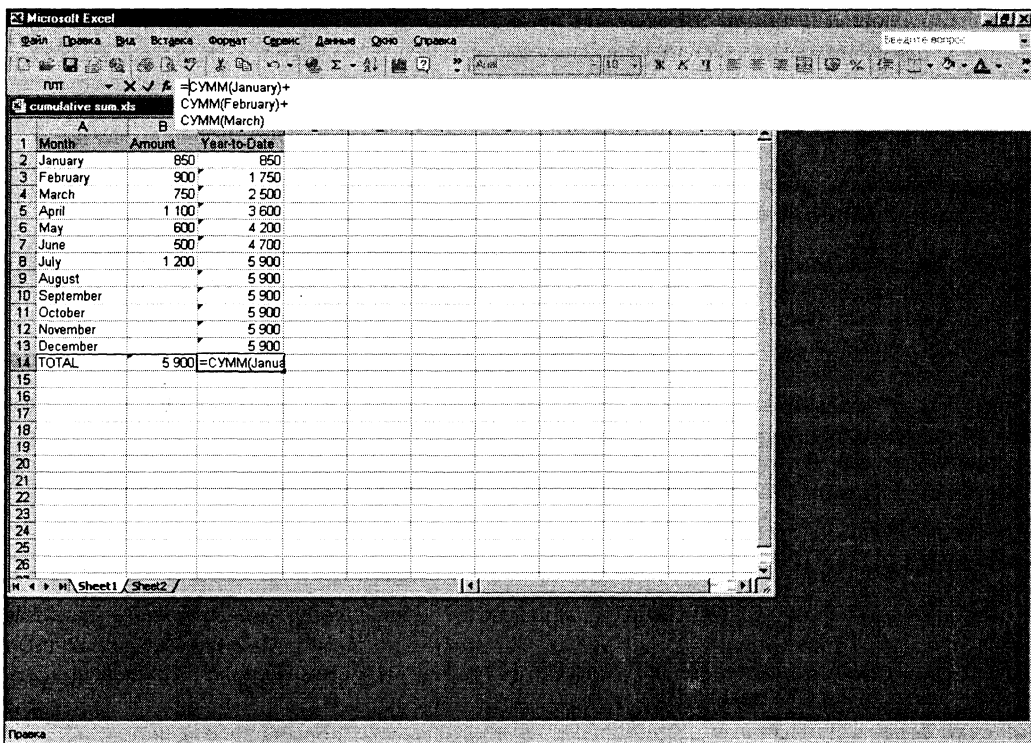


Рис. 2.2. В этой формуле использованы пробелы и разрывы строки

Предельный размер формул

Формула не может содержать более, чем 1024 символа. Если вам необходима формула, количество символов в которой превышало бы предельно допустимое, разделите ее на несколько формул. При желании вы можете создать пользовательскую функцию (с помощью VBA).



О том, как создавать пользовательские функции, читайте в части IV.

Примеры формул

Следуя приведенным выше инструкциям, вы сможете создать разнообразные формулы. В этом разделе я приведу несколько примеров формул.

- ♦ Следующая формула служит для умножения 150 на 0,01. Результат вычисления — 1,5. В этой формуле использованы только алгебраические значения, поэтому особой пользы она не приносит (вместо формулы можно ввести значение 1,5).

=150*.01

- ♦ С помощью этой формулы складываются значения ячеек A1 и A2:

=A1+A2

- ◆ Следующая формула предназначена для вычитания значения ячейки с именем *Expenses* из значения ячейки с именем *Income*.
=Income-Expenses
- ◆ В следующей формуле используется функция СУМ, с помощью которой складываются значения ячеек диапазона A1 : A12.
=СУММ(A1 : A12)
- ◆ А эта формула сравнивает значения ячеек A1 и C12 с использованием оператора =. Если значения обеих ячеек равны, формула выдаст результат ИСТИНА; в противном случае результат будет — ЛОЖЬ.
=A1=C12
- ◆ Последняя формула вычитает значение ячейки B3 из значения ячейки B2, а затем умножает результат вычисления на значение ячейки B4:
=(B2-B3) * B4

Редактирование формул

В случае необходимости внесения изменений в рабочий лист, вам, возможно, придется редактировать формулы. Если же формула возвращает одно из значений ошибки (описанных далее в этой главе), то ее необходимо исправить, чтобы устранить ошибки. Отредактировать формулу можно так же, как и значение любой другой ячейки.

Режим редактирования ячеек можно активизировать несколькими способами:

1. Дважды щелкните кнопкой мыши, расположив курсор внутри ячейки. Это позволит внести изменения непосредственно в ячейке. Данный способ работает только в том случае, если включена опция *Правка прямо в ячейке*. Включить эту опцию можно на вкладке *Правка* диалогового окна *Параметры*.
2. Нажмите клавишу <F2>. Это позволит вам редактировать содержимое непосредственно в ней. Если же опция *Правка прямо в ячейке* отключена, редактирование будет осуществляться в строке формул.
3. Выделите ячейку с формулой, подлежащей редактированию, и щелкните кнопкой мыши, расположив курсор в строке формул. Это позволит вам редактировать формулу в строке формул.
4. Щелкните на кнопке *Изменить формулу* (она имеет значок с изображением знака равенства) в строке формул. Появится панель формул.



В Excel 2002 кнопка *Изменить формулу* отсутствует.

При редактировании формул вы можете выделять несколько символов перетаскивая указатель мыши или с помощью клавиш со стрелками, удерживая <Shift> в нажатом состоянии. Кроме того, можно использовать клавиши <Home> и <End> для выделения части формулы от текущего места расположения курсора до начала или конца формулы. Использование клавиш со стрелками с комбинацией <Ctrl+Shift> позволяет выделять отдельные элементы формулы.

Использование строки формул в качестве калькулятора

Если вам необходимо произвести вычисления, используйте строку формул в качестве калькулятора. Например, введите в ячейку следующую формулу:

`= (145*1,05) / 12`

Такая формула всегда возвращает один и тот же результат, поэтому имеет смысл сохранить результат вычисления, а не формулу. Нажмите <F2> и отредактируйте содержимое ячейки. Затем нажмите <F9> и <Enter>. Excel сохранит результат вычисления (12,6875), а не саму формулу. Эта техника работает и в том случае, если в формуле используются ссылки на ячейки.

Использование этой техники особенно эффективно при работе с функциями рабочего листа. Например, для того чтобы поместить в ячейку квадратный корень из 221, введите `=КОРЕНЬ(221)`, нажмите <F9> и <Enter>. Excel возвратит результат: 14,8660687473185. Таким способом вы можете вычислить значение определенной части формулы. Рассмотрим такой пример:

`= (145*1,05) / A1`

Если вы хотите заменить выражение в скобках определенным значением, войдите в режим редактирования ячеек и выделите ту часть формулы, значение которой необходимо вычислить. В данном случае следует выделить `145*1,05`. Затем нажмите <F9> и <Enter>. Excel изменит формулу таким образом:

`= (152,25) / A1`



Предположим, вы задали длинную формулу, в которой присутствует ошибка. Программа не позволяет вам работать с данной формулой из-за ошибки. В таком случае рекомендую преобразовать формулу в текст и вернуться к ней позже. Для того чтобы преобразовать формулу в текст, просто удалите знак равенства (=). Когда вы все же захотите разобраться с этой формулой, вставьте знак равенства. Таким образом содержимое ячейки будет снова преобразовано в формулу.

Применение операторов в формулах

Как я уже говорил, оператор — один из основных элементов формулы. *Оператор* — это символ, обозначающий операцию. Excel поддерживает следующие операторы:

+	Сложение
-	Вычитание
/	Деление
*	Умножение
%	Процент
&	Объединение последовательностей символов в одну строку
^	Возведение в степень
=	Логическое сравнение (равно)
>	Логическое сравнение (больше)
<	Логическое сравнение (меньше)
>=	Логическое сравнение (больше или равно)
<=	Логическое сравнение (меньше или равно)
<>	Логическое сравнение (не равно)

Вы, конечно же, можете применять любое количество операторов. При использовании множества операторов формулы могут получиться достаточно сложными.

Операторы ссылки

Excel поддерживает еще один класс операторов, так называемые *операторы ссылки*. Операторы ссылки, приведенные в следующем списке, применяются к ссылкам на ячейки.

- : (двоеточие) Оператор диапазона. Используется для ссылки на все ячейки между крайними ячейками диапазона включительно.
- , (запятая) Оператор объединения. Объединяет несколько ссылок на ячейки или диапазоны в одну ссылку.
- (одиночный пробел) Оператор пересечения. Создает ссылку на ячейки, общие для двух диапазонов.

Примеры формул с операторами

В этих примерах формул присутствуют различные операторы:

- ◆ Приведенная ниже формула объединяет две строки текста в новую строку — *Part-23A*:

= "Part-" & "23A"

- ◆ Следующая формула объединяет содержимое ячеек A1 и A2:

=A1&A2

Обычно объединение применяется при управлении текстовыми строками. Однако этот оператор может быть применен и для объединения цифровых значений. Например, если ячейка A1 содержит значение 123, а ячейка A2 — значение 456, приведенная выше формула возвратит значение 123456. Обратите внимание: формально результат представляет собой текстовую строку, и тем не менее эта строка воспринимается как цифровое значение.

- ◆ В следующей формуле оператор возведения в степень используется для возведения 6 в третью степень. Результат операции — 216.

=6^3

- ◆ Приведенная выше формула имеет и более удобный вариант, в котором вместо цифрового значения используется ссылка на ячейку. В этом примере значения ячейки A1 возводятся в третью степень:

=A1^3

- ◆ А эта формула возвращает кубический корень из 216 (т.е. 6):

=216^(1/3)

- ◆ Следующая формула возвращает ИСТИНА, если значение ячейки A1 меньше значения ячейки A2. В противном случае результат — ЛОЖЬ.

=A1<A2

Операторы логического сравнения могут быть использованы и при работе с текстом. Если A1 содержит значение *Alpha*, а A2 — значение *Gamma*, формула возвращает ИСТИНА, поскольку в алфавитном порядке *Alpha* стоит раньше *Gamma*.

- ◆ Следующая формула возвращает ИСТИНА, если значение ячейки A1 меньше или равно значению ячейки A2. В противном случае результат будет — ЛОЖЬ.

=A1<=A2

- ◆ Следующая формула возвращает ИСТИНА, если значение ячейки A1 не равно значению ячейки A2. В противном случае результат будет — ЛОЖЬ.

=A1<>A2

- ◆ В отличие от некоторых других электронных таблиц (например, Lotus 1-2-3), в Excel нет логических операторов И или ИЛИ. Вместо них используются функции. Следующая формула, например, возвращает ИСТИНА, если ячейка A1 содержит значение 100 или 1000:

=ИЛИ (A1=100 ; A1=1000)

Последняя формула возвращает ИСТИНА только в том случае, если обе ячейки — A1 и A2 — содержат значения, меньше 100:

=И (A1<100 ; A2<100)

Приоритет операторов

В формулах можно (и нужно) использовать скобки. Скобки позволяют контролировать порядок выполнения вычислений. В качестве примера рассмотрим следующую формулу, которая содержит ссылки на именованные ячейки.

=Income-Expenses*TaxRate

Эта формула предназначена для того, чтобы вычесть значение расходов из значения доходов и затем умножить результат на величину налоговой ставки. Если вы введете формулу в том виде, в котором она приведена выше, Excel выдаст неправильный результат. Программа умножит расходы на процентную ставку, а результат вычитет из дохода. Верный результат возвратит формула, записанная следующим образом:

=(Income-Expenses)*TaxRate

Для того чтобы понять принцип, в соответствии с которым следует расставлять скобки, необходимо ознакомиться с таким понятием, как приоритет операторов. *Приоритет операторов* — это свод правил, согласно которым Excel производит вычисления. Рассмотрев табл. 2.1, вы узнаете, какой приоритет имеет каждый из операторов Excel. Вначале выполняются операции с более высоким приоритетом, затем — с менее высоким.

Таблица 2.1. Приоритет операторов в формулах Excel

Символ	Оператор	Приоритет
-	Отрицание	1
%	Процент	2
^	Возведение в степень	3
* и /	Умножение и деление	4
+ и -	Сложение и вычитание	5
&	Объединение двух текстовых строк в одну	6
=, <, >, <=, >=, и <>	Сравнение	7

Для изменения порядка выполнения операций используются скобки. Возвращаясь к приведенному выше примеру, заметим, что вычисления в формуле без скобок производятся согласно правилу приоритета операторов Excel. Операция умножения имеет более высокий приоритет, чем опе-

рация вычитания, программа сначала умножает расходы на процентную ставку. Затем результат вычитается из дохода. Таким образом формула возвращает неправильный результат.

В правильном варианте формулы используются скобки, что и позволяет контролировать порядок выполнения действий. Вначале всегда выполняются те части формулы, которые заключены в скобки. В данном случае, значение расходов вычитается из значения доходов и затем результат умножается на величину налоговой ставки.

Вложенные скобки

Скобки в формулах можно вкладывать. Вложенные скобки — это скобки, находящиеся внутри других скобок. Если в формуле используются вложенные скобки, Excel вычислит сначала выражение, находящееся внутри скобок с наибольшей глубиной вложения, а затем — все остальные. Ниже приведен пример формулы со вложенными скобками.

```
= ( (B2*C2) + (B3*C3) + (B4*C4) ) *B6
```

В этой формуле — четыре пары скобок, три из которых вложены в четвертую. Программа вычисляет значения выражений, находящихся во вложенных скобках, суммирует эти значения, а затем умножает результат на значение ячейки B6.

Скобки в формулах рекомендуется использовать даже в том случае, когда нет необходимости изменять приоритетный порядок вычислений. Скобки разъясняют порядок выполнения операций, делая формулу более читабельной. Например, если вы хотите прибавить единицу к результату умножения значений двух ячеек, следующая формула, конечно же, справится с задачей:

```
=A1*A2+1
```

Однако, порядок выполнения действий в таком варианте записи формулы (с лишними скобками) будет более очевиден:

```
= (A1*A2) +1
```

Каждая открывающая скобка, естественно, должна иметь парную ей закрывающую скобку. Если в формуле присутствует множество скобок разной глубины вложения, контролировать пары скобок нелегко. В случае, если какая-либо из скобок не имеет парной скобки, Excel выдаст соответствующее сообщение и не позволит вам ввести формулу в ячейку.



К счастью, Excel поможет вам проследить за скобками в формулах. При вводе или редактировании формул внимательно следите за текстом. Когда курсор располагается возле скобки, программа на мгновение выделяет ее и парную ей скобку полужирным шрифтом. Будьте внимательны — это продлится менее секунды.

Не задавайте строгих значений

Создавая формулу, старайтесь не использовать строгих цифровых значений. Например, если формула предназначена для вычисления суммы 7,5-процентного налога с оборота, вы, вероятно, захотите записать формулу так:

```
=A1* ,075
```

Однако разумнее ввести значение процентной ставки в ячейку, и вместо цифрового значения использовать ссылку на ячейку. Это упростит корректировку и дальнейшую работу с рабочим листом. Например, если процентная ставка изменится до 7,75, вам не придется изменять каждую формулу, в которой было использовано устаревшее значение. Если значение процентной ставки хранится в ячейке, все, что вы должны будете сделать, — это изменить значение одной единственной ячейки. В результате обновлены будут все формулы, в которых используется ссылка на эту ячейку.

Если в формуле имеются непарные скобки, Excel, в некоторых случаях, может предложить варианты исправления ошибки (в Excel 97 появилась функция автоматического исправления ошибок в формулах).



Вы, конечно же, захотите последовать совету, предложенному программой в диалоговом окне. Будьте осторожны. В некоторых случаях исправленная по совету программы формула будет синтаксически правильной, но эта формула будет функционировать совсем не так, как вы ожидаете. На рис. 2.3, я пропустил закрывающую скобку после Январь. Excel предложил такое исправление:

=СУММ (Январь / СУММ (Всего))

В действительности формула должна иметь следующий вид:

=СУММ (Январь) / СУММ (Всего)

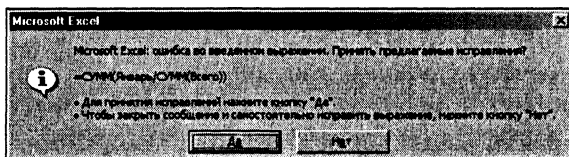


Рис. 2.3. Функция автоматического исправления ошибок Excel предлагает пути устранения ошибок в формулах

Вычисление формул

Вы, вероятно, заметили, что вычисление формулы происходит сразу же, после ее ввода. Если вы измените значения ячеек, используемых в формуле, формула выдаст новый результат без малейших усилий с вашей стороны. Так происходит только в том случае, если на вкладке Вычисление активна опция Автоматически. При вычислении формул в этом режиме (установленном по умолчанию) Excel следует таким правилам:

- ◆ Когда вы вносите изменения (например, вводите новые или редактируете данные или формулы), Excel сразу же пересчитывает те формулы, которые зависят от новых или измененных данных.
- ◆ Производя длительные расчеты, Excel временно приостанавливает вычисление в том случае, если вам необходимо выполнить какие-либо действия в программе; вычисление возобновляется, когда вы закончите свои действия.
- ◆ Формулы вычисляются в соответствии с последовательностью натуральных чисел. Например, если формула, расположенная в ячейке D12 зависит от результата вычисления формулы ячейки D11, ячейка D11 будет вычисляться прежде ячейки D12.

Иногда полезно контролировать, когда программа будет проводить вычисление формул. Например, если вы создаете рабочий лист с тысячами сложнейших формул, вам придется ждать целую вечность, пока Excel вычислит значения всех формул. В этом случае целесообразно включить режим вычисления Вручную. Сделать это можно на вкладке Вычисления диалогового окна Параметры. (Для вызова диалогового окна выберите Сервис⇒Параметры.)

Когда вы работаете в ручном режиме вычисления, Excel отображает команду *Вычислить* в строке состояния в том случае, если в рабочем листе остались невычисленные формулы. Для пересчета формул можно использовать следующие комбинации клавиш:

- ◆ <F9>. Вычисляет формулы во всех открытых рабочих книгах.
- ◆ <Shift+F9>. Вычисляет формулы только активной рабочей книги.
- ◆ <Ctrl+Alt+F9>. Проводит полный пересчет во всех открытых рабочих книгах. Используйте эту комбинацию клавиш, если Excel (по той или иной причине) возвращает неправильные значения.
- ◆ <Ctrl+Shift+Alt+F9>. Перепроверяет все зависимые формулы, а затем проводит полный пересчет во всех открытых рабочих книгах.



Комбинация клавиш <Ctrl+Shift+Alt+F9> работает только в Excel 2002.



Режим вычисления Excel относится не только к тому рабочему листу, при работе с которым был применен этот режим. Изменение режима отражается на всех открытых рабочих книгах, а не только на активной книге. Исходным режимом вычисления становится тот режим, который сохранен в первой открытой вами рабочей книге.

Ссылки на ячейки и диапазоны ячеек

Во многих формулах используются ссылки на одну и более ячеек. Для ссылки используется адрес ячейки или диапазона (или имя, если это — именованная ячейка или диапазон). Существует четыре типа ссылок. Различать эти типы помогает знак доллара:

- ◆ **Относительные:** ссылка полностью относительна. При копировании формулы ссылка на ячейку вставляется в новое место. Пример: A1
- ◆ **Абсолютные:** ссылка полностью абсолютна. При копировании формулы в другую ячейку ссылка не изменяется. Пример: \$A\$1
- ◆ **Абсолютная строка:** ссылка частично абсолютна. При копировании формулы изменяется часть ссылки, относящаяся к столбцу. Та часть, которая относится к строке, остается неизменной. Пример: A\$1
- ◆ **Абсолютный столбец:** ссылка частично абсолютна. При копировании формулы изменяется часть ссылки, относящаяся к строке. Та часть, которая относится к столбцу, остается неизменной. Пример: \$A1

Создание абсолютной ссылки

Когда вы создаете формулу, указывая на ячейки, все ссылки на ячейки и диапазоны будут относительными. Для того чтобы ссылка стала абсолютной, вам придется вручную расставить знаки доллара. Есть и еще один способ: когда вы вводите адреса ячеек или диапазонов, используйте клавишу <F4>. С ее помощью можно перебирать все возможные режимы ссылки.

Режим ссылки целесообразно изменять только в том случае, если вы планируете копировать формулу. На рис. 2.4 приведен пример использования абсолютных ссылок в формуле. Обратите внимание на формулу в ячейке C4:

=C\$3*\$B4

	A	B	C	D	E	F	G	H	I	J	K
1											
2											
3			1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
4		1.0	1.00	1.50	2.00	2.50	3.00	3.50	4.00	4.50	5.00
5		1.5	1.50	2.25	3.00	3.75	4.50	5.25	6.00	6.75	7.50
6		2.0	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00
7		2.5	2.50	3.75	5.00	6.25	7.50	8.75	10.00	11.25	12.50
8		3.0	3.00	4.50	6.00	7.50	9.00	10.50	12.00	13.50	15.00
9		3.5	3.50	5.25	7.00	8.75	10.50	12.25	14.00	15.75	17.50
10		4.0	4.00	6.00	8.00	10.00	12.00	14.00	16.00	18.00	20.00
11		4.5	4.50	6.75	9.00	11.25	13.50	15.75	18.00	20.25	22.50
12		5.0	5.00	7.50	10.00	12.50	15.00	17.50	20.00	22.50	25.00
13											

Рис. 2.4. Пример использования абсолютных ссылок в формуле

Эта формула предназначена для вычисления площади для различной ширины (значения перечислены в столбце В) и длины (значения перечислены в строке 3). Введенная формула может затем быть скопирована вниз по столбцу и вдоль строки. А поскольку в формуле использованы абсолютные ссылки на строку 3 и столбец В, каждая скопированная формула возвращает правильный результат. Если бы в формуле были использованы относительные ссылки, ссылки изменялись бы при копировании и возвращали неправильный результат.

Сравнение типов ссылок A1 и R1C1

По умолчанию Excel использует тип ссылок A1. Адрес каждой ячейки состоит из буквы столбца и номера строки. Однако программа поддерживает и другую систему записи — R1C1. В соответствии с этой системой, ячейка A1 имеет адрес R1C1, A2 — R2C1 и т.д.

Если вы хотите использовать систему записи R1C1, выберите Сервис⇒Параметры. В диалоговом окне Параметры перейдите на вкладку Общие и выставьте флажок опции Стиль ссылок R1C1. Изменив тип ссылок, вы заметите, что столбцы в этом режиме обозначаются не буквами, а цифрами. Изменяются и все ссылки на ячейки и диапазоны в формулах.

Обратите внимание на примеры записи формул в стандартной системе и в системе R1C1. Предположим, что формула расположена в ячейке B1 (или же R1C2).

Стандартная

=A1+1
 =\$A\$1+1
 =\$A1+1
 =A\$1+1
 =СУММ(A1:A10)
 =СУММ(\$A\$1:\$A\$10)

R1C1

=RC[-1]+1
 =R1C1+1
 =RC1+1
 =R1C[-1]+1
 =СУММ(RC[-1]:R[9]C[-1])
 =СУММ(R1C1:R10C1)

Система R1C1 окончательно сбита вас с толку? И не только вас. Тип ссылок R1C1 приемлем при работе с абсолютными ссылками. Но когда речь идет об относительных ссылках — в этих скобках и сам черт ногу сломит!

Числа в скобках означают относительное расположение. R[-5]C[-3], например, означает, что ячейка находится на пять строк выше и на три столбца левее. Соответственно, R[5]C[3] ссылается на ячейку, расположенную на пять строк ниже и на три столбца правее. Если вы опустите какую-либо из скобок, запись будет означать ячейку в том же столбце или строке. Например, R[5]C означает ячейку, расположенную на пять строк ниже в том же столбце.

Вы, вероятно, не будете использовать R1C1 в качестве основной системы представления ячеек. Однако у этой системы есть по крайней мере одно преимущество: в R1C1 очень легко обнаружить формулу с ошибкой. При копировании формула в R1C1 остается неизменной вне зависимости от типа ссылок (относительного, абсолютного или смешанного). Поэтому

очень удобно включить режим R1C1 для проверки скопированных формул. Если одна из формул отличается от остальных, она, возможно, содержит ошибку.

Если вы работаете в Excel 2002, воспользуйтесь функцией фоновой проверки формул. Эта функция поможет найти формулы, которые, вероятно, содержат ошибки. О фоновой проверке формул я расскажу в главе 21.

Создание ссылок на ячейки других рабочих листов или рабочих книг

В формуле могут использоваться ссылки на ячейки и диапазоны ячеек, расположенные в других рабочих листах. Для того чтобы задать ссылку на ячейку другого листа, расположите перед адресом ячейки имя листа, за которым следует восклицательный знак. Рассмотрим следующий пример формулы со ссылкой на ячейку другого рабочего листа (Лист2):

```
=Лист2!A1+1
```

Можно создавать формулы и со ссылками на ячейки другой рабочей книги. Для этого перед ссылкой на ячейку введите имя рабочей книги (в квадратных скобках), затем — имя листа и восклицательный знак:

```
=[Budget.xls]Лист1!A1+1
```

Если в имени рабочей книги присутствует один и более пробелов, имя книги (и имя рабочего листа) следует заключить в одинарные кавычки. Например:

```
='[Budget Analysis.xls]Лист1'!A1+1
```

Если рабочая книга, на которую задается ссылка, закрыта, необходимо добавить полный путь к файлу этой книги:

```
= 'C:\MSOffice\Excel\[Budget Analysis.xls]Лист1'!A1+1
```

Создавать внешние ссылки можно и просто указывая на ячейки (об этом методе я уже упоминал). Для этого файл источника должен быть открыт. Обычно создать формулу можно, указав на результаты в относительных ссылках на ячейки. Но когда с помощью этого метода вы обращаетесь к ячейке другой рабочей книги, Excel создает *абсолютные* ссылки (если вы планируете копировать формулу в другие ячейки, ссылки необходимо сделать относительными).



При работе со внешними ссылками могут возникнуть неожиданные трудности. Например, если вы используете команду **Файл**⇒**Сохранить** как для того, чтобы сделать копию книги-источника, связанные формулы автоматически изменяются и ссылка указывает уже на другой файл (зачастую не тот, что вам нужен). Путаница со ссылками может возникнуть и в случае переименования файла книги-источника.

Создание точной копии формулы

При копировании формулы Excel автоматически изменяет ссылки на ячейки, когда вы вставляете формулу в новую ячейку. Именно это чаще всего и нужно пользователю. Но бывают случаи, когда необходимо сделать точную копию формулы. Для этого следует преобразовать относительные ссылки в абсолютные, как было описано ранее. Но этот способ не всегда оптимален.

Удобнее выделить формулу в режиме редактирования, а затем скопировать ее в буфер как обыкновенный текст. Для этого существует несколько способов. Предлагаю вам пошаговую инструкцию, следуя которой можно сделать точную копию формулы, расположенной в ячейке A1, и вставить ее в ячейку A2:

Использование ссылок для восстановления данных поврежденного файла

По тем или иным причинам одна из книг Excel может быть повреждена. Если поврежденную книгу невозможно загрузить, вы можете создать связанную формулу и с ее помощью восстановить всю информацию или ее часть (формулы восстановить невозможно). Сделать это можно, поскольку для создания связанной формулы файл не обязательно должен быть открыт. Например, поврежденный файл называется `Badfile.xls`. Для того чтобы восстановить данные рабочего листа Лист1, откройте пустую рабочую книгу и введите следующую формулу в ячейку A1:

```
=[Badfile.xls]Лист1!A1
```

Скопируйте эту формулу вниз и вправо, чтобы восстановить максимум информации. Разумеется, лучше всего делать резервные копии наиболее важных файлов.

Если вы работаете в Excel 2002, поврежденная рабочая книга не составляет большой проблемы, поскольку современная версия восстанавливает большинство разрушенных файлов.

1. Дважды щелкните кнопкой мыши, расположив курсор в ячейке A1 (или нажмите <F2>); включится режим редактирования.
2. Для того чтобы выделить всю формулу, нажмите клавишу <End>, а затем комбинацию <Shift+Home>. Вы также можете перетащить указатель мыши, выделив таким образом всю формулу.
3. Щелкните на кнопке Копировать, расположенной на панели Стандартная (или нажмите комбинацию клавиш <Ctrl+C>). Выделенный текст будет скопирован в буфер обмена.
4. Для выхода из режима редактирования нажмите <Enter>.
5. Активизируйте ячейку A2.
6. Щелкните на кнопке Вставить, расположенной на панели Стандартная (или нажмите комбинацию клавиш <Ctrl+V>). В результате точная копия формулы будет вставлена в ячейку A2.

Эту технику вы можете использовать и для копирования части формулы, которую можно использовать при создании новой формулы. Просто выделите ту часть формулы, которую вы хотите скопировать, перетащив указатель мыши или используя клавишу <Shift> одновременно с клавишами со стрелками. Затем выделенную часть формулы скопируйте в буфер обмена любым из описанных выше способов. И, наконец, вставьте текст в другую ячейку.

Ссылки в формулах (или частях формул), скопированных таким образом, останутся неизменными при вставке в другую ячейку, поскольку вы копируете формулы как текст.

Существует еще один способ получения точной копии формулы. Формулу необходимо отредактировать, удалив знак равенства. Это преобразует формулу в текст. Этот текст следует скопировать в новую ячейку, а затем добавить знаки равенства в исходную формулу и ее копию.

Преобразование формул в значения

Если на вашем рабочем листе используется набор формул, которые всегда возвращают один и тот же результат (это — так называемые “мертвые формулы”), разумно преобразовать эти формулы в значения. Для этого используется команда Правка⇒Специальная вставка.

Предположим, что диапазон A1:A10 содержит формулы, возвращающие неизменный результат. Для того чтобы преобразовать эти формулы в значения, сделайте следующее:

1. Выделите диапазон A1:A10.
2. Щелкните на кнопке Копировать, расположенной на панели Стандартная (или нажмите комбинацию клавиш <Ctrl+C>).
3. Выполните команду Правка⇒Специальная вставка. Появится диалоговое окно Специальная вставка.
4. Выберите переключатель Значения и щелкните на кнопке ОК.
5. Отмените режим вставки, нажав <Enter> или <Esc>.



Если вы работаете в Excel 2002, воспользуйтесь смарт-тэгом. Третьим пунктом приведенной выше последовательности будет выполнение команды Правка⇒Вставить (или нажатие <Ctrl+V>). В правом нижнем углу диапазона появится смарт-тэг. Щелкните на нем и выберите опцию Только значения (рис. 2.5).

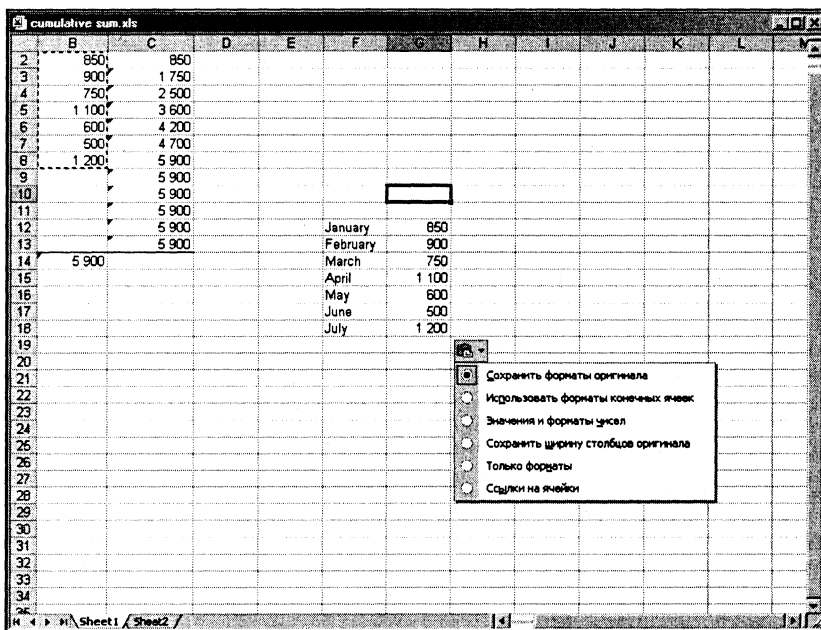


Рис. 2.5. В Excel 2002 после операции вставки данных появляется смарт-тэг

Эта техника особенно эффективна в тех случаях, когда формулы используются для преобразования ячеек. Предположим, у вас есть список имен (в верхнем регистре), расположенный в столбце A. Вы хотите, чтобы эти имена были записаны в нижнем регистре. Для этого необходимо создать формулы в отдельном столбце, затем преобразовать эти формулы в значения и заменить ими исходные значения столбца A. Выполнять это нужно следующим образом:

1. Вставьте новый столбец после столбца A.
2. В ячейку B1 поместите следующую формулу:
=ПРОПИСН (A1)
3. Скопируйте формулу вниз по столбцу B столько раз, сколько элементов содержит столбец A. Теперь столбец B содержит значения столбца A, но в нужном регистре.

4. Выделите все имена столбца В.
5. Щелкните на кнопке Копировать, расположенной на панели Стандартная.
6. Выделите ячейку А1.
7. Выполните команду Правка⇒Специальная вставка. Появится диалоговое окно Специальная вставка.
8. Выставьте переключатель Значения и щелкните на кнопке ОК.
9. Отмените режим вставки, нажав <Enter> или <Esc>.
10. Удалите столбец В.

Формулы против функции автозаполнения

Функция автозаполнения Excel позволяет быстро копировать содержимое ячейки в смежные ячейки. Кроме того, эта функция может быть применена не только для копирования в смежные ячейки, а в некоторых случаях автозаполнение может заменить формулу. Меня удивляет тот факт, что многие опытные пользователи Excel не применяют автозаполнение, хотя эта функция помогает сэкономить массу времени.

Например, расположить список значений от 1 до 100 в диапазоне А1:А100 можно, применив формулу. В ячейку А1 вы вводите 1, в ячейку А2 — формулу =А1+1, а затем копируете эту формулу в оставшиеся 98 ячеек диапазона.

Используя функцию автозаполнения, вы сможете обойтись без формул. Для того чтобы заполнить упомянутый диапазон последовательностью от 1 до 100, введите 1 в ячейку А1, а в ячейку А2 — 2. Затем выделите диапазон А1:А2 и перемещайте маркер диапазона вниз до ячейки А100. (Маркер диапазона — это небольшой квадратик в правом нижнем углу активной ячейки.) Когда вы проводите автозаполнение таким образом, Excel анализирует содержимое выделенных ячеек и использует полученную информацию для заполнения остальных ячеек диапазона. Если ячейка А1 содержит 1, а ячейка А2 — 3, программа улавливает принцип, по которому составляется последовательность, и заполняет остальные ячейки таким образом: 5, 7, 9 и т.д. Та же техника применяется и для создания убывающих последовательностей (10, 9, 8 и т.д.), и списков дат. Если же программе не удастся уловить принцип, по которому составляется последовательность, Excel выполняет линейную регрессию и вставляет значения в соответствии с полученной тенденцией.

Excel распознает такие общеизвестные списки, как названия месяцев года и дней недели. Если вы введете Понедельник и начнете перемещать манипулятор заполнения, программа автоматически вставит в ячейки дни недели по порядку. Excel позволяет создавать пользовательские списки автозаполнения. Создать их можно на вкладке Списки диалогового окна Параметры. И наконец, если вы будете перемещать маркер, щелкнув правую кнопку мыши, появится контекстное меню, в котором вы сможете выбрать необходимую опцию автозаполнения.

Скрытие формул

В некоторых случаях может возникнуть необходимость скрыть формулы. Вы, вероятно, не захотите, чтобы кто-либо смог увидеть специально разработанную вами формулу, которая является собственностью вашей компании. Скрыть формулы можно с помощью диалогового окна Формат ячеек.

Для того чтобы сделать невозможным просмотр одной и более формул, следуйте приведенной ниже инструкции:

1. Выделите формулу или формулы.
2. В меню Формат выберите команду Ячейки, а затем перейдите на вкладку Защита.

3. Установите флажок Скрыть формулы, как показано на рис. 2.6.
4. В меню Сервис откройте подменю Защита и выберите команду Защитить лист. Для того чтобы никто кроме вас не мог отменить защиту листа, не забудьте установить пароль в диалоговом окне Защитить лист.

По умолчанию все ячейки “закрыты”. Если лист защищен, это значит, что никто не сможет изменить “закрытые” ячейки. Следовательно, перед тем, как защитить лист, вы должны будете “открыть” ячейки, предназначенные для ввода информации пользователями.



Вы должны знать о том, что существуют программы взлома паролей. Следовательно, техника скрывать формул не может гарантировать того, что ваши формулы не увидит абсолютно никто.

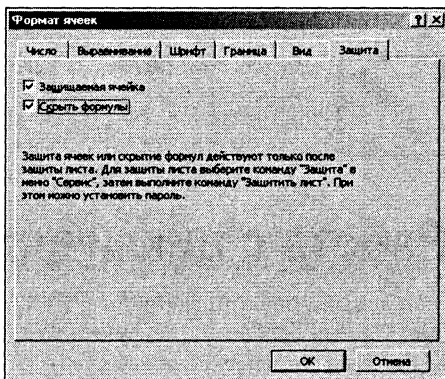


Рис. 2.6. Скрыть формулы можно в диалоговом окне *Формат ячеек*

Ошибки в формулах

Нередко программа обнаруживает ошибку во вводимых формулах. В табл. 2.2. представлены типичные ошибки, которые появляются в ячейке, содержащей формулу. Формулы могут возвращать значения ошибки и в том случае, если в ячейке, ссылку на которую содержит эта формула, присутствует значение ошибки. Это — так называемый волновой эффект: одна единственная ячейка, содержащая ошибку, может повлиять на множество ячеек, в которых расположены зависимые от этой ячейки формулы.

Таблица 2.2. Типичные ошибки в Excel

Значение ошибки	Пояснения
#ДЕЛ/0! (#DIV/0!)	Формула пытается совершить операцию деления на ноль (что на нашей планете строго запрещено). Это значение ошибки возникает и в том случае, если формула пытается разделить значение на пустую ячейку.
#ИМЯ? (#NAME?)	В формуле использовано имя, которое Excel не воспринимает. Это может происходить в случае, если имя, используемое в формуле, было удалено, или в записи имени допущена опечатка.
#Н/Д (#N/A)	Формула обращается (прямо или косвенно) к ячейке, в которой используется функция НД, что служит сигналом обнаружения неопределенных данных. Это значение ошибки может появляться и в том случае, если искомая функция не имеет соответствия.
#ПУСТО! (#NULL!)	Эта ошибка появляется, когда задано пересечение двух диапазонов, которые в действительности не имеют общих ячеек. (Об этом я расскажу далее в этой главе.)
#ЧИСЛО! (#NUM!)	В формуле возникла проблема со значением. Например, вы ввели отрицательное значение там, где должно быть положительное.
#ССЫЛ! (#REF!)	Ошибка появляется, когда используется недопустимая ссылка на ячейку. Происходит это, например, когда ячейка, на которую ссылается формула, удалена из рабочего листа.
#ЗНАЧ! (#VALUE!)	Ошибка появляется, когда используется недопустимый тип аргумента или операнда. Операнд обращается к значению или ссылке на ячейку, которые используются в формуле для вычисления значения.



Если ячейка заполняется символами “решеток” (#####), то это, обычно, означает, что ширина столбца не позволяет разместить значение. Вы можете увеличить ширину столбца или изменить цифровой формат ячейки. Символы “решетки” появляются в ячейке и в том случае, когда формула, расположенная в этой ячейке, возвращает ошибочное время или дату.



В Excel 2002 рядом с формулой, содержащей ошибку, появляется смарт-тэг. Щелкнув на нем, вы можете узнать, какая ошибка была допущена и на каком этапе вычислений она появилась. Подробную информацию о смарт-тэге вы найдете в главе 21.

Работа с циклическими ссылками

Когда вы вводите формулы в Excel, программа может выдать сообщение, показанное на рис. 2.7. Это сообщение говорит о том, что формула, которую вы только что ввели, привела к появлению *циклической ссылки*.

Циклическая ссылка появляется в случае, если формула содержит прямую или косвенную ссылку на свое собственное значение. Например, если в ячейку A3 вы введете формулу $=A1+A2+A3$, то создадите циклическую ссылку, поскольку в формуле, расположенной в ячейке A3, присутствует ссылка на эту ячейку. В результате после каждого вычисления значения ячейки A3 это значение должно вычисляться снова и снова, поскольку меняется значение A3. Таким образом вычисление происходило бы бесконечно долго и формула никогда не могла бы вернуть окончательный ответ.

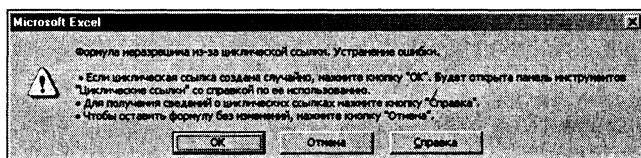


Рис. 2.7. Вот так Excel сообщает пользователю о появлении циклической ссылки

Если вы ввели формулу, содержащую циклическую ссылку, Excel отображает диалоговое окно с тремя вариантами действий:

- ◆ Если циклическая ссылка создана случайно, щелкните на кнопке ОК. Будет открыта панель инструментов Циклические ссылки со справкой по ее использованию.
- ◆ Для получения сведений о циклических ссылках щелкните на кнопке Справка.
- ◆ Чтобы оставить формулу без изменений, щелкните на кнопке Отмена.

Вы, конечно же, предпочтете исправить циклическую ссылку и щелкните на кнопке ОК. Excel откроет панель инструментов Циклические ссылки (рис. 2.8). На панели Циклические ссылки выберите первую ячейку из раскрывающегося списка Найти циклическую ссылку. Затем проанализируйте формулу, которую содержит эта ячейка. Если вы не уверены в том, что именно эта формула привела к появлению циклической ссылки, выберите в списке Найти циклическую ссылку следующую ячейку. Продолжайте перебирать варианты формул, пока в строке состояния не исчезнет сообщение о цикличности.

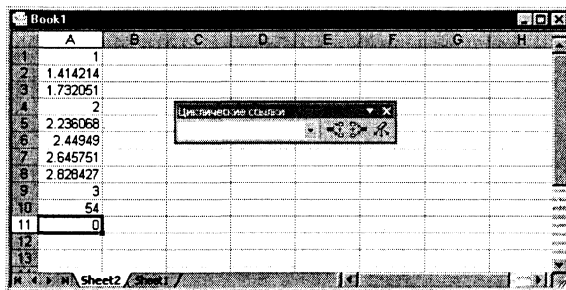


Рис. 2.8. Панель инструментов Циклические ссылки



В редких случаях пользователь намеренно вводит циклическую ссылку. Некоторые примеры использования циклических ссылок приведены в главе 16.

Если вы не отреагируете на замечание программы о появлении циклической ссылки и щелкнете на кнопке Отмена, Excel позволит вам ввести эту формулу. В строке состояния будет присутствовать сообщение, напоминающее о существовании циклической ссылки. Если вы начнете работу в другом рабочем листе или книге, в напоминании будет отсутствовать ссылка на ячейку.



Excel не оповещает пользователя о появлении циклической ссылки, если активна опция Итерации. Проверить состояние этой опции можно на вкладке Вычисления диалогового окна Параметры. Если опция Итерации активна, Excel выполняет циклические вычисления столько раз, сколько задано в поле Предельное число итераций, или же пока значение не изменится на меньше, чем 0,001 (или любое другое значение, указанное в поле Относительная погрешность). Рекомендую держать опцию Итерации отключенной. Тогда программа будет предупреждать вас о появлении циклических ссылок. Ведь чаще всего существование циклической ссылки свидетельствует об ошибке, которую необходимо исправить.

Чаще всего причина циклической ссылки бывает очевидна и исправить такую ошибку достаточно легко. Но иногда вы будете сталкиваться с косвенными циклическими ссылками. Иными словами, формула ссылается на формулу, которая ссылается на другую формулу, содержащую ссылку на исходную формулу. В таких случаях найти виновника циклической ссылки бывает непросто.

Функция подбора параметра

Многие электронные таблицы содержат формулы, которые позволяют задавать вопросы типа: “Какова будет сумма общей прибыли, если объем продаж вырастет на 20 процентов?” Если вы задали формулу правильно, то можете изменить значение одной ячейки и увидеть, как изменится значение объема продаж в другой ячейке.

Функция подбора параметра очень эффективна при совместном применении с формулами. Если вы знаете, какой должен быть результат, программа сама подскажет вам значения одной и более ячеек, необходимые для достижения этого результата. Иными словами, вы можете задать такой вопрос: “На сколько процентов должен вырасти объем продаж, чтобы общая прибыль составила \$1,2 миллиона?”

Подбор параметра для одной ячейки — задача несложная. Excel определяет прямо в ячейке, какое значение должно быть введено в формулу, чтобы получился нужный результат. Ознакомиться с работой этой функции программы лучше всего, рассмотрев конкретный пример.

Пример подбора параметров

На рис. 2.9 представлена таблица, предназначенная для расчета суммы займа под залог недвижимости. В этой таблице есть четыре ячейки для ввода данных (C4 : C7) и четыре ячейки с формулами (C10 : C13). Формулы вычисляют различные значения, используя ячейки для ввода значений. В ячейках расположены следующие формулы:

C10: = (1-C5) * C4

C11: =ПЛТ (C7/12;C6-C10)

C12: =C11*C6

C13: =C12-C10

Input Cells	
Purchase Price	\$325 000
Down Payment	20%
Loan Term (months)	360
Interest Rate	8.00%
Result Cells	
Loan Amount	\$260 000
Monthly Payment	\$1 907.79
Total Payments	\$686 804
Total Interest	\$426 804

Рис. 2.9. Эта таблица — яркий пример использования функции подбора параметра

Выберите Сервис⇒Подбор параметра, как показано на рис. 2.10. Заполнить окна этого диалогового окна — все равно, что составить предложение: Установить в ячейке C11 значение 1200, Изменяя значение ячейки C4. Введите в диалоговое окно всю необходимую информацию, задав ссылки на ячейки или указав их мышью.

Предположим, вы решили купить новый дом. Максимальная сумма, которой вы располагаете, — \$1200 в месяц в виде закладного платежа. Также известно, что заимодавец может выдать заем с фиксированной ставкой — 8%, основываясь на процентном соотношении суммы займа к стоимости заложенного имущества, составляющем 80% (размер займа на 20 процентов меньше стоимости имущества). Вопрос состоит в следующем: “Какова максимальная цена, по которой вы сможете купить новый дом?” Иными словами, какое значение в ячейке C4 приведет к результату \$1200 в ячейке C11? Вы можете вставлять в ячейку C4 значения, пока в ячейке C11 не появится результат \$1200. Однако лучше всего поручить подбор значения программе.

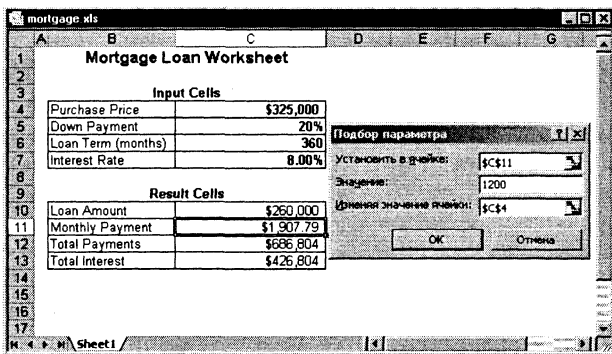


Рис. 2.10. Диалоговое окно Подбор параметра

Excel мгновенно сообщает о том, что решение найдено, отображая окно **Результат подбора параметра**. В окне вы увидите подбираемое значение и текущее значение. В нашем примере программа нашла точное значение. Подбираемое значение было помещено в ячейку C4 (\$204,425). В результате получается желаемая оплата \$1200 в месяц. Окно **Результат подбора параметра** предоставляет следующие варианты выбора:

- ◆ Щелкните на кнопке **ОК**, чтобы заменить исходное значение найденным значением.
- ◆ Щелкните на кнопке **Отмена**, и ваша таблица вернется к своему исходному состоянию, в котором она находилась до применения команды **Сервис**⇒**Подбор параметра**.

Еще немного о подборе параметра

Excel не всегда находит параметр, удовлетворяющий заданным условиям, — иногда подходящего значения просто не существует. В таком случае окно **Результат подбора параметра** информирует вас о невозможности подбора параметра. Впрочем, Excel иногда выдает сообщение о невозможности подбора параметра и в том случае, когда вы уверены в том, что такой параметр существует. Тогда, вы можете заменить текущее значение в изменяемой ячейке на значение, больше приближенное к решению, а затем применить команду повторно. Если подобрать параметр не удастся со второй попытки, то, скорее всего, вы ошиблись в своих рассуждениях. Проверьте, действительно ли формула зависит от ячейки с изменяемым значением.

Как и любая другая программа, Excel имеет ограниченную точность вычислений. Для того чтобы в этом убедиться, введите в ячейку A2 формулу $=A1^2$. Затем выберите **Сервис**⇒**Подбор параметра** и найдите такое значение ячейки A1, чтобы результатом вычисления стало число 16. Excel возвратит параметр 4,00002269, что лишь приблизительно равно квадратному корню из 16. Изменить точность вычислений можно на вкладке **Вычисления** диалогового окна **Параметры** (следует уменьшить значение в поле **Относительная погрешность**).

В некоторых случаях желаемый результат дают несколько значений изменяемой ячейки. Например, формула $=A1^2$ возвращает 16, если ячейка A1 содержит -4 или +4. В том случае, если существует несколько подходящих значений параметра, Excel отображает то из них, которое имеет тот же знак, что и текущее значение в ячейке, или то, которое наиболее приближено к этому значению.

Значительным ограничением команды **Сервис**⇒**Подбор параметра** является то, что с ее помощью можно подобрать значение только для одной ячейки. Например, программа не сможет рассчитать и максимальную цену и процентную ставку при заданной месячной оплате. Если вы хотите одновременно подобрать значения для двух переменных, используйте надстройку **“Поиск решения”**.

Резюме

Эта глава служит введением к теме **“Формулы в Excel”**. В ней описаны основные составляющие элементы формулы. В главе 2 рассмотрены такие темы, как относительные и абсолютные ссылки, преобразование формул в значения, ошибки в формулах и циклические ссылки.

В следующей главе я расскажу о том, как работать с именами в Excel.

Работаем с именами

В этой главе...

- ◆ Что такое Имя?
- ◆ Способы присвоения имен ячейкам и диапазонам
- ◆ Присваивание имен в нескольких листах
- ◆ Область действия имен
- ◆ Работа с именами диапазонов и ячеек
- ◆ Поддержка имен ячеек и диапазонов
- ◆ Проблемы при работе с именами
- ◆ Ключ к пониманию термина “имя”
- ◆ Решение сложных задач с помощью формул
- ◆ Резюме

Большинство пользователей Excel знакомы с такими понятиями, как именованные ячейки и диапазоны. Работа с именованными ячейками и диапазонами дает значительные преимущества. Прочитав эту главу, вы узнаете о различных типах имен, которые могут быть использованы в Excel. Возможно, вы и не подозреваете о том, как именованные ячейки смогут помочь вам в работе.

Что такое Имя?

Имя можно рассматривать как идентификатор какого-либо элемента рабочей книги. Имя может быть присвоено таким элементам, как ячейка, диапазон, диаграмма, фигура и т.д. Имя, присвоенное, например, диапазону, может затем использоваться в формулах. Предположим, что диапазон B2 : B200 рабочего листа содержит информацию о дневном обороте компании. В ячейке C1 задана ставка комиссионного сбора с продажи. В следующей формуле сумма дневного оборота умножается на ставку комиссионного сбора:

```
=СУММ(B2 : B200) * C1
```

Такая формула будет работать, однако смысл ее не очевиден. Для того чтобы формула стала понятной, можно назначить описательные имена диапазону, содержащему информацию о дневном обороте, и ячейке, в которой указана ставка комиссионного сбора. Например, диапазону B2 : B200 можно присвоить имя *DailySales*, а ячейке C1 — имя *ComissionRate*. Теперь формулу можно переписать, используя имена вместо адресов диапазона и ячейки:

```
=СУММ(DailySales) * ComissionRate
```

Как видите, теперь формула говорит сама за себя. Формулу, в которой вместо ссылок на ячейки используются имена, понять значительно легче.

Использование именованных ячеек и диапазонов имеет ряд преимуществ:

- ◆ Имена позволяют легче понять смысл формулы и использовать ее. Это особенно важно для тех, кто не участвовал в создании рабочего листа. Очевидно, что такую формулу как, например, =Income-Taxes, понять значительно легче, чем =D20-D40.
- ◆ Описательное имя диапазона (например, *Total_Income*) запоминается легче, чем адреса ячеек (например, A21). Кроме того, вероятность опечатки при написании имени меньше, чем при вводе адреса ячейки или диапазона.
- ◆ Вы можете легко перемещаться по рабочему листу, используя поле Имя, расположенное в левой части строки формул (щелкните на кнопке со стрелкой, и появится список всех назначенных имен). Перейти от одного элемента листа к другому можно и выбрав Правка⇒Перейти или нажав клавишу <F5>. Появится диалоговое окно, в которое следует ввести имя диапазона или ячейки.
- ◆ Когда вы выделяете именованную ячейку или диапазон, имя этого элемента появляется в поле Имя.
- ◆ Возможно, вам будет легче записывать формулы, используя имена ячеек и диапазонов. Эти имена вы можете вставлять в формулу, применяя команду Вставка⇒Имя⇒Вставить или нажимая клавишу <F3>.
- ◆ Использование имен вместо адресов ячеек и диапазонов облегчит создание макросов и дальнейшую работу с ними.

Способы присвоения имен ячейкам и диапазонам

В Excel есть несколько способов присвоения имен ячейкам и диапазонам. В данном разделе я расскажу вам об этих методах. Кроме того, здесь вы найдете и другую полезную информацию, имеющую отношение к именам.

Присвоение имен в окне Присвоение имени

Для того чтобы присвоить имя ячейке или диапазону, сначала выделите нужную ячейку или диапазон. Затем выберите Вставка⇒Имя⇒Присвоить или нажмите комбинацию клавиш <Ctrl+F3>. Появится диалоговое окно Присвоение имени, показанное на рис. 3.1.

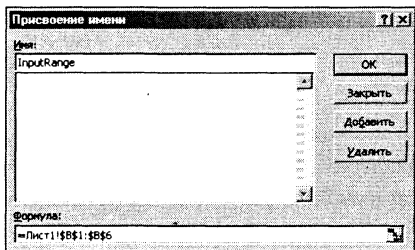


Рис. 3.1. Используйте диалоговое окно Присвоение имени для создания имен ячеек и диапазонов

Введите имя в текстовом поле Имя или используйте имя, предложенное программой (если таковое будет предложено). Адрес выделенной ячейки или диапазона появится в поле Формула. Убедитесь в том, что в поле указан правильный адрес. Затем щелкните на кнопке ОК, и на рабочем листе появится заданное вами имя. Диалоговое окно будет закрыто.

Если вы хотите продолжить присвоение имен ячейкам и диапазонам рабочего листа, щелкните на кнопке Добавить. В этом случае необходимо либо ввести адрес в поле Формула (адрес должен начинаться со знака равенства), либо указать на нужную ячейку рабочего листа.



Количество имен ячейки и диапазона не ограничено. Excel позволяет присваивать одному и тому же элементу более одного имени, однако делать это вряд ли имеет смысл. Если выделенная ячейка или диапазон имеет несколько имен, в поле Имя всегда будет появляться первое из них.

Имя может задаваться и прерывному диапазону ячеек. Выделить все ячейки и диапазоны прерывистого диапазона можно с помощью мыши, нажав клавишу <Ctrl>.



Редактируя содержимое поля Формула вручную, убедитесь в том, что это поле работает в режиме “указания”. Это означает, что при редактировании нельзя использовать клавиши <End> и <Home>. Для того чтобы переключиться из режима “указания” в стандартный режим редактирования, нажмите <F2>. Активизировав поле Формула в режиме редактирования, вы сможете использовать все клавиши редактирования.

Присвоение имени с помощью поля Имя

В поле Имя вы сможете присвоить имя ячейке или диапазону значительно быстрее. Поле Имя — это раскрывающееся меню, расположенное в левой части строки формул. Для того чтобы присвоить имя, выделите нужную ячейку или диапазон, затем щелкните на поле Имя и введите имя. Для подтверждения выбранного имени нажмите <Enter>. Если такое имя уже существует, поле Имя нельзя использовать для переназначения этого имени другому диапазону или ячейке. Если вы попытаетесь сделать это, программа выделит исходный диапазон. Назначить существующее имя другому диапазону или ячейке можно только с помощью диалогового окна Присвоение имени.



Создавая имя в поле Имя, не забудьте нажать <Enter>. Если вы введете имя, а затем щелкните в области рабочего листа, новое имя не будет создано.

Правила создания имен

Что касается назначаемых вами имен, Excel достаточно неприхотлив. И все же есть некоторые правила создания имен:

- ◆ Имена не могут содержать пробелов. Для разделения слов внутри имени можно использовать символ подчеркивания или точку (например, *Annual_Total* или *Annual.Total*).
- ◆ В качестве имени можно использовать любую комбинацию букв и цифр. Однако любое имя должно начинаться с буквы или символа подчеркивания. Имя не может начинаться с цифры (как, например, *3rdQuarter*) или выглядеть как ссылка на ячейку (как, например, *Q3*).
- ◆ Для разделения слов внутри имени нельзя использовать иные символы, кроме символа подчеркивания и точки. Хотя это и не указано в документации к программе, Excel воспринимает и такие символы, как обратная косая черта (\) и вопросительный знак (?). Заметьте — имя не может начинаться с этих символов.
- ◆ Количество символов имени не может превышать 255. Поверьте моему опыту — использовать имена такой длины не имеет смысла. Использование чрезмерно длинных имен лишает всякого смысла операцию присвоения имени диапазону.
- ◆ В качестве имени вы можете использовать отдельные буквы (кроме R и C). Однако я бы не рекомендовал давать такие имена, поскольку имя должно нести информацию о содержимом ячейки или диапазона.

- ◆ При вводе имен регистр значения не имеет. *AnnualTotal* значит то же самое, что и *annualtotal*. Excel сохраняет имя точно в таком же виде, в каком вы вводите его первоначально. Но записывая имя в формуле, вы можете использовать любой регистр.

Excel использует несколько так называемых внутренних имен. Вы можете присваивать внутренние имена Excel ячейкам и диапазонам. Но я бы не рекомендовал вам этого делать. Итак, присваивая имя, вам следует избегать следующих имен: *Print_Area*, *Print_Titles*, *Consolidate_Area*, *Database*, *Criteria*, *Extract*, *FilterDatabase* и *Sheet_Title*.

Поле Имя имеет и еще одну функцию. С его помощью вы можете быстро активизировать именованную ячейку или диапазон, как показано на рис. 3.2. Для того чтобы выделить ячейку или диапазон, щелкните в поле Имя и выберите нужное имя из списка. Будет выделена соответствующая ячейка или диапазон. Как ни странно, для активизации поля Имя не предусмотрена комбинация клавиш. Иными словами, вы не сможете получить доступ к этому полю с помощью клавиатуры; сделать это можно только используя мышь. А вот после того, как вы щелкните в поле Имя, для указания имени достаточно использовать клавиши со стрелками и <Enter>.

Автоматическое присвоение имени

Рабочий лист может содержать некий текст, который вы хотите использовать в качестве имени смежных ячеек или диапазонов. На рис. 3.3 вы видите пример такого листа. В данном примере текст, находящийся в столбце А, может быть использован для создания имен ячеек с соответствующими значениями столбца В. Сделать это в Excel достаточно просто.

	B	C	D	E
Amount	Washington	\$125.50	06 май	06 май
DateDue	Washington	\$3 000.00	12 май	06 май
Difference	Oregon	\$2 100.00	25 май	06 май
InvoiceNum	Oregon	\$335.39	25 май	06 май
Office	Washington	\$65.00	30 май	06 май
Today	California	\$250.00	01 июн	06 май
11 AG-0633				
12 TOTAL		\$14 767.45		

Рис. 3.2. Используя поле Имя, вы сможете быстро активизировать именованную ячейку или диапазон

	A	B	C	D	E
1	Month	Sales			
2	Jan	25,984			
3	Feb	28,973			
4	Mar	21,983			
5	Apr	32,744			
6	May	31,982			
7	Jun	27,006			
8	Jul	31,982			
9	Aug	33,321			
10	Sep	27,440			
11	Oct	26,123			
12	Nov	29,831			
13	Dec	18,732			
14					

Рис. 3.3. Excel позволяет создавать имена на основе текста, содержащегося в смежных ячейках

Для того чтобы присвоить имена, используя текст смежных ячеек, выделите этот текст и ячейки, которым должно быть присвоено имя (это могут быть как отдельные ячейки, так и диапазоны ячеек). Текст имен должен находиться в смежных ячейках относительно ячеек, которым вы хотите назначить имена (вы можете выделить несколько ячеек). Выберите Вставка⇒Имя⇒Создать или нажмите комбинацию клавиш <Ctrl+Shift+F3>. Появится диалоговое окно Создать имена, показанное на рис. 3.4.

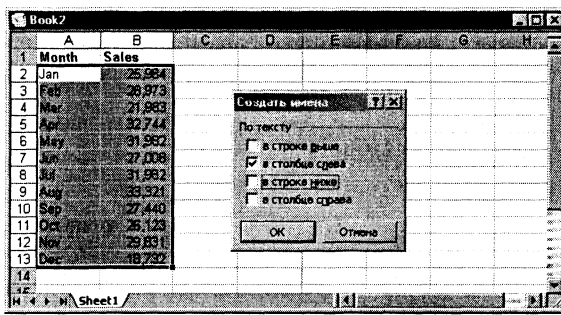


Рис. 3.4. Диалоговое окно *Создать имена*

Опции в диалоговом окне расставлены на основе проведенного программой анализа выделенного диапазона. Например, если Excel обнаруживает текст в первой строке выделенного диапазона, программа предлагает вам создать имена в строке выше. Если Excel задал опции неправильно, то вы можете изменить предложенный вариант вручную. Щелкните на кнопке ОК, и Excel создаст имена. Обратите внимание на то, что когда Excel создает имена, используя текст смежных ячеек, ячейки с текстом не включаются в именованный диапазон.

Если в результате использования текста смежной ячейки получится некорректное имя, Excel изменит это имя так, чтобы оно было корректным. Например, если ячейка содержит текст *Net Income* (использование такого текста в качестве имени некорректно, поскольку в нем присутствует пробел), Excel преобразует пробел в символ подчеркивания и создаст имя *Net_Income*. Если же вместо текста смежная ячейка содержит значение или формулу, программа не сможет создать корректное имя. В таком случае Excel просто не создаст никакого имени.

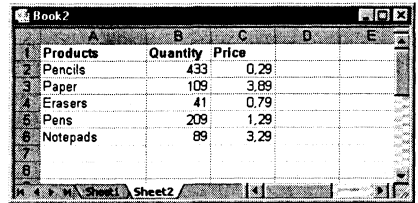


Рис. 3.5. Создание имен на основе данных такой таблицы может привести к неожиданным результатам



Внимательно проверяйте имена, автоматически создаваемые программой. Иногда применение команды Вставка⇒Имя⇒Создать дает вовсе не тот результат, который вы ожидали. На рис. 3.5 представлена небольшая таблица, состоящая из текста и значений. Выделите всю таблицу, выберите Вставка⇒Имя⇒Создать и примите предложения программы (опции в строке выше и в столбце слева). В результате имя *Products* будет относиться не к диапазону A2:A6, как вы ожидали, а к диапазону B2:C6. Если верхняя левая ячейка выделенного диапазона содержит текст, и вы выберете опции в строке выше и в столбце слева, Excel использует это имя для всего набора данных, включая верхнюю строку и левый столбец. Поэтому, прежде, чем принять опции, предлагаемые командой, убедитесь, что этот выбор соответствует вашим замыслам.

Имя столбца или строки

Иногда имеет смысл назначить имя целому столбцу или строке. Нередко рабочий лист используется для хранения информации, которую вы вводите время от времени. На рис. 3.6 приведен пример такого листа. Если вы создаете имя для данных столбца B, вам придется изменять ссылку на диапазон, которому назначено это имя, каждый раз, когда вы вводите новые данные. В таком случае разумно присвоить имя целому столбцу.

Столбец В вы можете назвать, например, *DailySales*. Если этот диапазон расположен на рабочем листе Лист3, ссылка будет выглядеть следующим образом:

=Лист3!\$B:\$B

Назначив имя, вы можете использовать его в формуле. Следующая формула, например, возвращает сумму всех значений столбца В:

=СУММ(DailySales)

Имена, созданные программой Excel

Excel создает и свои собственные имена. Например, если вы зададите область печати для листа, Excel создаст имя *Print Area*. Если же вы назначите повторяющиеся строки или столбцы, которые должны быть напечатаны, программа создаст имя *Print Titles* на уровне рабочего листа. Когда вы выполняете запрос, который возвращает данные в рабочий лист, Excel назначает этим данным свои имена. Множество надстроек, входящих в пакет Excel, создают скрытые имена (см. врезку “Скрытые имена”).

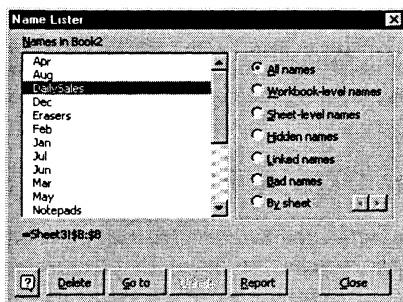
Вы можете изменить ссылки на любые имена, автоматически создаваемые программой. Однако, вы должны ясно представлять себе, к чему эти изменения приведут.

Скрытые имена

Некоторые макросы и надстройки Excel создают скрытые имена. Эти имена присутствуют в рабочей книге, но не появляются в диалоговом окне Присвоение имени или в поле Имя. Надстройка “Поиск решения”, например, создает целый ряд скрытых имен. Чаще всего скрытые имена можно просто игнорировать. Но они могут стать причиной появления некоторых проблем. Если вы копируете лист из одной рабочей книги в другую, вместе с листом копируются и скрытые имена. При этом скрытые имена могут образовать ссылку, которую очень трудно обнаружить.

Работать с именами в Excel нелегко. Например, вы не сможете просмотреть полный список имен, присутствующих в рабочей книге. В диалоговом окне Присвоение имени вы найдете только имена уровня листа, присутствующие в активном рабочем листе. В этом окне вы не найдете скрытых имен.

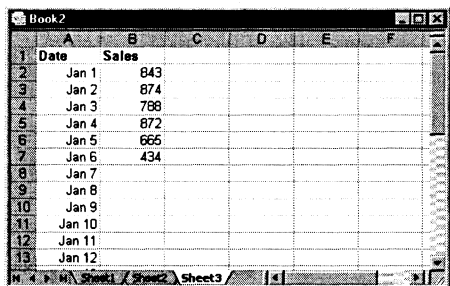
Если вам необходим эффективный инструмент для работы с именами, используйте утилиту Name Lister, которая является частью пакета Power Utility Pak. Эта утилита отображает полный список имен. Кроме того, Name Lister позволяет фильтровать имена несколькими способами. Вы можете просмотреть только имена уровня рабочего листа или же только связанные имена. Данная утилита поможет найти и удалить имена, присвоенные неправильно диапазону.



Присваивание имен в нескольких листах

Имена ячейкам и диапазонам могут быть присвоены в нескольких рабочих листах с помощью “трехмерной” ссылки. Но вы не можете просто выделить диапазон и ввести имя в поле **Имя**. Для того чтобы назначить имя в нескольких листах, необходимо использовать диалоговое окно **Присвоение имени**. “Трехмерная” ссылка будет выглядеть следующим образом: **ПервыйЛист: ПоследнийЛист! СсылкаНаДиапазон**

На рис. 3.7 продемонстрирован пример, в котором имя (*DataCube*), назначенное диапазону A1 : C3, распространяется на листы Лист1, Лист2 и Лист3.



Date	Sales
Jan 1	843
Jan 2	874
Jan 3	788
Jan 4	872
Jan 5	665
Jan 6	434
Jan 7	
Jan 8	
Jan 9	
Jan 10	
Jan 11	
Jan 12	

Рис. 3.6. В этой таблице, содержащей данные о дневном объеме продаж, используется именованный ряд, который состоит из целого столбца

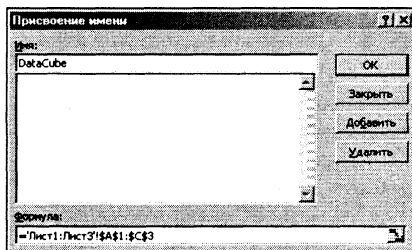


Рис. 3.7. Присвоение имени в нескольких листах

Разумеется, вы можете просто ввести трехмерную ссылку на диапазон в поле **Формула**. Но если вы хотите создать имя, указав на диапазон, тут могут возникнуть некоторые трудности. Даже если вы попытаетесь выделить диапазон, распространяющийся на несколько рабочих листов, Excel не сможет использовать адрес выделенного вами диапазона в диалоговом окне **Присвоение имени**.

Для того чтобы назначить имя *DataCube* диапазону A1 : C3, который распространяется на листы Лист1, Лист2 и Лист3, следуйте приведенным ниже инструкциям:

1. Активируйте Лист1.
2. Выберите **Вставка**⇒**Имя**⇒**Присвоить** или нажмите комбинацию клавиш <Ctrl+F3>. Появится диалоговое окно **Присвоение имени**.
3. Введите *DataCube* в текстовое поле **Имя**.
4. Активируйте поле **Формула** и нажмите клавишу , чтобы удалить ссылку на диапазон.
5. Выделите диапазон A1 : C3 на листе Лист1. В поле **Формула** появится следующая запись:
`=Лист1!A1:C3`
6. Удерживая нажатой клавишу <Shift>, щелкните на Лист3. Вы заметите, что программа по непонятным причинам меняет ссылку на диапазон ссылкой на одну ячейку. На данном этапе запись будет выглядеть так:
`= 'Лист1:Лист3' ! A1`

7. Выделите диапазон A1:C3 на листе Лист1 еще раз. В поле Формула появится следующая запись:

```
= 'Лист1:Лист3' ! $A$1:$C$3
```

8. Теперь, когда поле Формула содержит нужную ссылку на диапазон, щелкните на кнопке ОК. Диалоговое окно Присвоение имени будет закрыто.

Задав имя, вы сможете использовать его в формулах. Например, следующая формула возвращает сумму значений диапазона, который носит имя *DataCube*.

```
=СУММ(DataCube)
```



Имена, распространяющиеся на несколько листов, не появляются в поле Имя или в диалоговом окне Переход (это окно появляется при выборе Правка⇒Перейти). Иными словами, Excel позволяет назначать такие имена, но вы не сможете автоматически выделить ячейки, к которым относится это имя.

Если вы вставите новый лист в рабочую книгу, в которой используются имена, относящиеся к нескольким листам, эти имена будут распространяться и на новый лист в том случае, если этот лист находится между первым и последним листом, заданным в определении имени. В предыдущем примере рабочий лист, вставленный между Лист1 и Лист2, будет включен в диапазон *DataCube*. Лист, вставленный перед Лист1 или после Лист3, не будет включен в диапазон.

Если вы удалите первый или последний лист, включенный в определение имени, Excel автоматически изменит запись в поле Формула. В рассматриваемом нами примере удаление Лист1 приведет к следующим изменениям в поле Формула имени *DataCube*:

```
= 'Лист2:Лист3' ! $A$1:$C$3
```

Область действия имен

Имя, которое вы присваиваете ячейке или диапазону, обычно может использоваться в любом из листов рабочей книги. Например, если вы присвоили имя *RegionTotal* ячейке A1 листа Лист1, это имя может использоваться в любой формуле на любом рабочем листе. Такое имя называется именем уровня рабочей книги (или глобальным именем). По умолчанию все создаваемые имена ячеек и диапазонов являются именами уровня рабочей книги.

Создание имен уровня рабочего листа

А что, если вам нужно использовать какое-либо имя (например, *RegionTotal*) в каждом из листов рабочей книги? В таком случае вы должны создать имена уровня рабочего листа (которые иногда называют локальными именами).

Для того чтобы создать имя *RegionTotal* на уровне рабочего листа, активизируйте лист, в котором должно задаваться это имя, и выберите Вставка⇒Имя⇒Присвоить. Появится диалоговое окно Присвоение имени. В поле Имя перед именем уровня рабочего листа следует задать имя листа, за которым следует восклицательный знак. Например, для того чтобы назначить имя *RegionTotal* на рабочем листе Лист2, необходимо активизировать Лист2 и ввести в поле Имя диалогового окна Присвоение имени следующую запись:

```
Лист2!RegionTotal
```

Если в имени рабочего листа присутствует хотя бы один пробел, имя необходимо заключить в кавычки:

```
'Marketing Dept'!RegionTotal
```

Создать имя на уровне рабочего листа можно и с помощью поля **Имя**. Для этого выделите ячейку или диапазон, которому вы хотите дать имя, щелкните кнопкой мыши, расположив курсор в поле **Имя**, и введите имя. Перед именем ячейки или диапазона должно стоять имя рабочего листа и восклицательный знак (как показано выше). Для подтверждения имени нажмите <Enter>.

Когда вы используете имя уровня рабочего листа в формуле того же листа, в котором это имя задано, нет необходимости указывать имя рабочего листа перед именем диапазона (имя рабочего листа не будет отображено и в поле **Имя**). Однако, если такое имя вы используете в формуле на другом рабочем листе, имя рабочего листа с восклицательным знаком перед именем диапазона задавать необходимо.



В поле **Имя** отображаются только те имена уровня рабочего листа, которые заданы в текущем листе. Аналогично только имена текущего листа появляются в списках диалоговых окон **Присвоение имени** и **Вставка имени**.

Комбинирование имен уровня рабочего листа с именами уровня рабочей книги

Используя имена уровня рабочего листа, можно легко запутаться. Дело в том, что Excel позволяет создавать имя уровня рабочего листа даже в том случае, если такое же имя задано на уровне рабочей книги. В таком случае приоритетным является имя уровня рабочего листа, но только в том листе, в котором это имя задано.

Например, вы можете задать имя *Сумма* на уровне рабочей книги для ячейки листа *Лист1* и имя *Лист2!Сумма* на уровне рабочего листа. Когда *Лист2* активен, *Сумма* — это имя на уровне рабочего листа. Когда активны другие листы книги, *Сумма* является именем, заданным на уровне рабочей книги. Совсем запутались? Неудивительно. Дабы избежать путаницы, не применяйте одно и то же имя на уровне листа и рабочей книги.

Ссылка на имя из другой рабочей книги

В главе 2 рассказано о том, как создавать ссылки на ячейки и диапазоны других рабочих книг. Те же правила работают и при ссылке на имена, заданные в другой рабочей книге.

Например, в следующей формуле используется имя *MonthlySales*, назначенное в рабочей книге под именем *Budget.xls* (предполагается, что эта рабочая книга открыта):

```
=СРЗНАЧ(Budget.xls!MonthlySales)
```

Работа с именами диапазонов и ячеек

Создав имена диапазонов и ячеек, вы можете выполнять над ними целый ряд операций. В этом разделе я расскажу вам о том, как выполнять основные операции над именами диапазонов и ячеек.

Создание списка имен

Если вы создали достаточно большое количество имен, может возникнуть необходимость зафиксировать, к какому диапазону относится каждое из этих имен. В дальнейшем это поможет вам выявить ошибки или документировать свою работу.

Вероятно, вы захотите создать список всех имен (и соответствующих им адресов) рабочей книги. Для того чтобы создать такой список, сначала выделите ячейку в пустой области рабочего листа (если вы создадите список имен, состоящий из двух столбцов, то он отображается поверх находящейся в ячейках информации). Затем выберите **Вставка**⇒**Имя**⇒**Вставить** (или нажмите клавишу <F3>). Появится диалоговое окно **Вставка имени** (рис. 3.8), в котором будут перечислены все заданные имена. Для того чтобы вставить список имен, щелкните на кнопке **Все имена**.

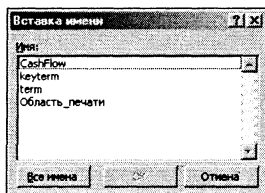


Рис. 3.8. Диалоговое окно **Вставка имени**

Список имен, вставленный в рабочий лист, состоит из двух столбцов. В первом столбце перечислены имена, а во втором — адреса соответствующих диапазонов. Адреса диапазонов, расположенные во втором столбце списка, представляют собой строки текста, которые напоминают формулы. Преобразовать такую строку в формулу можно, отредактировав ячейку (нажмите <F2>, а затем <Enter>). В результате строка становится формулой. Если имя относится к отдельной ячейке, формула отображает текущее значение этой ячейки. Если же имя относится к диапазону, формула возвращает ошибку #ЗНАЧ!



В список имен не входят имена уровня рабочего листа, кроме имен активного листа.

Использование имен в формулах

Задав имя ячейки или диапазона, вы можете использовать это имя в формулах. Если имя задано на уровне рабочей книги (по умолчанию), вы можете использовать это имя в формуле любого рабочего листа книги. Просто введите имя вместо ссылки на ячейку. Следующая формула, например, предназначена для вычисления суммы значений диапазона, который носит имя *UnitsSold*:

```
=СУММ(UnitsSold)
```

Когда имя, заданное на уровне рабочего листа, используется в формуле того листа, в котором это имя было задано, имя рабочего листа не следует указывать перед именем диапазона. Если же такое имя используется в другом листе рабочей книги, необходимо задавать полное имя (имя рабочего листа, восклицательный знак и имя диапазона). Например, если *UnitsSold* — имя, заданное на уровне рабочего листа *Лист1*, формула для вычисления суммы диапазона на любом другом рабочем листе будет выглядеть так:

```
=СУММ(Лист1!UnitsSold)
```

Вводя формулу, вы можете выбрать **Вставка**⇒**Имя**⇒**Вставить** (или просто нажать клавишу <F3>). Появится диалоговое окно **Вставка имени**. Выберите имя из списка, щелкните на кнопке **ОК**, и Excel вставит это имя в формулу. Как я уже говорил, в списке диалогового окна **Вставка имени** представлены все имена уровня рабочей книги и имена уровня рабочего листа для активного листа.

Если в формуле вы использовали имя, которого не существует, Excel выдаст ошибку #ИМЯ?, тем самым сообщая вам, что не может найти имя, которое вы хотите использовать в формуле. Чаще всего появление такого сообщения свидетельствует о допущенной вами ошибке при написании имени.

Использование операторов пересечения

В качестве оператора пересечения диапазонов в Excel служит единичный символ пробела. Например, следующая формула служит для вычисления суммы ячеек на пересечении двух диапазонов — B1:C20 и A8:D8:

```
=СУММ(B1:C20 A8:D8)
```

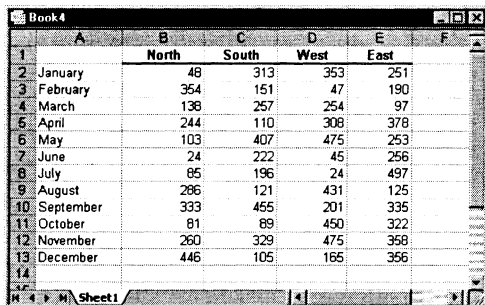
Данные диапазоны пересекаются в ячейках B8 и C8.

Оператор пересечения работает и для именованных диапазонов. На рис. 3.9 представлена таблица, содержащая именованные диапазоны, в которых заданы названия для столбцов и строк таблицы. Например, имя *January* относится к диапазону B2:E2, а имя *North* — к диапазону B2:B13. Следующая формула возвращает содержимое ячейки на пересечении диапазонов *January* и *North*:

```
=January North
```

Используя символ пробела для разделения двух ссылок на диапазоны, вы задаете так называемое *явное пересечение*. Это означает, что вы просите Excel явно определить пересечение диапазонов. Программа же может находить и *неявные пересечения*. Неявное пересечение происходит в том случае, когда Excel выбирает значение из многоячеечного диапазона, основываясь на строке или столбце формулы, содержащей ссылку. Для того чтобы разобраться с пересечениями такого типа, рассмотрим конкретный пример. На рис. 3.10 изображена таблица, содержащая именованный диапазон (B3:B8). В ячейке D5 содержится простая формула, приведенная ниже:

```
=MyData
```

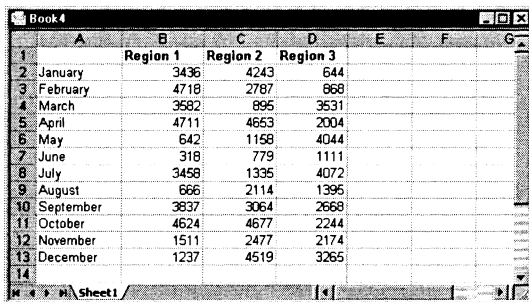


	A	B	C	D	E	F
1		North	South	West	East	
2	January	48	313	353	251	
3	February	354	151	47	190	
4	March	138	257	254	97	
5	April	244	110	308	378	
6	May	103	407	475	253	
7	June	24	222	45	256	
8	July	85	196	24	497	
9	August	286	121	431	125	
10	September	333	455	201	335	
11	October	81	89	450	322	
12	November	260	329	475	358	
13	December	446	105	165	356	
14						

Рис. 3.9. Эта таблица содержит именованные диапазоны, в которых заданы названия для столбцов и строк

Интуитивно понятные формулы? Почему бы и нет!

В Excel 97 и всех последующих версиях при записи формул вы можете применять названия диапазонов даже в том случае, если эти названия не были официально заданы как имена этих диапазонов. Такие формулы называются “интуитивно понятными”. Например, в таблице, представленной на рисунке, не задано ни одного имени.



	A	B	C	D	E	F	G
1		Region 1	Region 2	Region 3			
2	January	3436	4243	644			
3	February	4718	2787	868			
4	March	3582	895	3531			
5	April	4711	4653	2004			
6	May	642	1158	4044			
7	June	318	779	1111			
8	July	3458	1335	4072			
9	August	666	2114	1395			
10	September	3837	3064	2668			
11	October	4624	4677	2244			
12	November	1511	2477	2174			
13	December	1237	4519	3265			
14							

И тем не менее, Excel воспринимает названия строк и столбцов. Следующая формула, например, возвращает сумму значений строки, которая носит название *January*:

```
=СУММ(January)
```


Можно использовать и названия столбцов. Приведенная ниже формула возвращает сумму значений в столбце *Region 1*:

```
=СУММ(Region 1)
```

В формуле можно использовать и несколько таких названий одновременно. Следующая формула возвращает 2787 — значение на пересечении диапазонов *February* и *Region 2*:

```
=February Region 2
```

Работая с интуитивно понятными формулами, вы не тратите времени и сил на то, чтобы назначить диапазонам имена. Но эта функция не всегда работает так, как требуется. Так, формулы с “псевдоименами” иногда не вычисляются при изменении данных. Более того, случается, что две совершенно одинаковые формулы возвращают разные результаты. Существует и еще одна проблема: невозможно предсказать наверняка, как Excel интерпретирует конкретное название. И наконец, в Excel нельзя использовать более 32 764 интуитивных формул. Попробуйте превысить допустимый предел — и программа, вероятно, выйдет из строя.

Я настоятельно рекомендую не прибегать к этой функции программы и всегда использовать действительные имена ячеек и диапазонов. Для того чтобы отключить функцию использования названий диапазонов, выберите Сервис⇒Параметры. Затем перейдите на вкладку Вычисления диалогового окна Параметры и отключите параметр допускать названия диапазонов.

Эта настройка сохраняется отдельно для каждой рабочей книги. Значит, открыв новый файл, в котором разрешено использование псевдоимен, вы можете отключить соответствующий параметр для данного файла. После того как этот параметр будет отключен, Excel проанализирует все ваши формулы и автоматически преобразует все названия в ссылки на ячейки.

Обратите внимание на то, что в ячейке D5 отображено значение из диапазона *MyData*, которое соответствует строке формулы. Точно так же, если вы введете ту же формулу в строки с 3 по 8, формула будет отображать соответствующее значение из диапазона *MyData*. Excel производит неявное пересечение, используя диапазон *MyData* и строку, содержащую формулу. Это то же самое, как если бы вычислялась следующая формула:

```
=MyData 5:5
```

Если вы введете формулу в строку, не занятую диапазоном *MyData*, эта формула возвратит ошибку, поскольку неявное пересечение в таком случае произойти не может.

Между прочим, неявные пересечения работают не только с именованными диапазонами. В рассматриваемом нами примере мы получим

тот же результат, если в ячейке D5, будет находиться следующая формула (в которой не используется имя диапазона):

```
=$B$2 : $B$8
```

Использование операторов диапазона с именами

Работая с именованными диапазонами, вы можете использовать оператор диапазона, в качестве которого в Excel используется символ двоеточия (:). Вернемся к примеру на рис. 3.9. Следующая формула, например, возвращает сумму значений ячеек столбцов от *North* до *West* и строк от *January* до *March*:

```
=СУММ((North January):(West March))
```

	A	B	C	D	E	F
1						
2			6			
3			12			
4			18			
5			24	18		
6			36			
7			48			
8						
9						
10						

Рис. 3.10. Диапазон B3 : B8 этой таблицы носит имя *MyData*. Ячейка D5 демонстрирует неявное пересечение

Ссылка на отдельную ячейку именованного диапазона

Для того чтобы вернуть значение ячейки диапазона, можно использовать функцию Excel ИНДЕКС. Предположим, некий диапазон A1:A50 носит имя *DataRange*. Следующая формула возвращает второе значение (значение ячейки A1) диапазона *DataRange*:

```
=ИНДЕКС(DataRange; 2)
```

Второй и третий аргументы функции ИНДЕКС — дополнительные. Однако один из этих аргументов всегда должен быть задан. Второй аргумент (используемый в приведенной формуле) обозначает смещение строки в диапазоне *DataRange*.

Если диапазон *DataRange* состоит из нескольких ячеек, расположенных в одной строке, используйте примерно следующую формулу, в которой пропущен второй аргумент функции ИНДЕКС, но задан третий аргумент, определяющий смещение столбца в диапазоне *DataRange*:

```
=ИНДЕКС(DataRange; ; 2)
```

Если же диапазон состоит из нескольких строк и столбцов, необходимо использовать оба аргумента функции ИНДЕКС. Следующая формула, например, возвращает значение ячейки, расположенной в четвертой строке и пятом столбце диапазона *DataRange*:

```
=ИНДЕКС(DataRange; 4; 5)
```

Применение имен в уже существующих формулах

После того как вы создаете имя для ячейки или диапазона, Excel не заменяет имена ссылки на эту ячейку или диапазон в формулах, созданных ранее. Вы можете дать программе команду применить новое имя в тех или иных формулах.

Для этого выделите диапазон, содержащий формулы, которые должны быть преобразованы. Затем выберите Вставка⇒Имя⇒Применить. Появится диалоговое окно Применение имени, показанное на рис. 3.11. В диалоговом окне Применение имени выберите те имена, которые вы хотите применить в формулах. В формулах будут применены только те имена, которые вы указали.

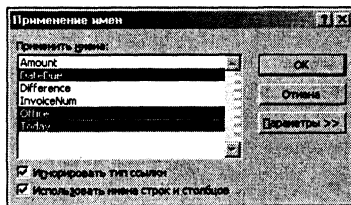


Рис. 3.11. Диалоговое окно Применение имени



Для того чтобы применить новое имя ко всем формулам рабочего листа, выделите одну ячейку перед тем, как выполнить команду Вставка⇒Имя⇒Применить.

Параметр Игнорировать тип ссылки отвечает за то, как Excel заменяет адрес диапазона именем. Обычно, имя ячейки или диапазона задается как абсолютная ссылка. Если параметр Игнорировать тип ссылки включен, Excel применит имя только в том случае, если ссылка в формуле полностью соответствует типу диапазона. В большинстве случаев, когда вы применяете имена, удобнее игнорировать тип ссылки.

Если параметр Использовать имена строк и столбцов включен, применяя имена в формулах, Excel использует оператор пересечения. Не имея возможности использовать точные имена ячеек, Excel применяет имена строк и столбцов для того, чтобы сослаться

на ячейки. Для того чтобы объединить имена, Excel использует оператор пересечения. Щелчком на кнопке **Параметры** можно задать несколько дополнительных параметров, но только при условии, что параметр **Использовать имена строк и столбцов** включен.

Автоматическое применение имен при создании формул

Если вы вставляете в формулу ссылку на ячейку или диапазон, указывая этот элемент таблицы мышью, Excel автоматически заменяет адрес ячейки или диапазона именем (в том случае, если данному элементу присвоено имя).



Такое поведение характерно только для Excel 97 и более современных версий программы.

Иногда такое свойство очень помогает в работе. Но в некоторых случаях, когда вы хотите использовать адрес ячейки, а не ее имя, это свойство только мешает. К сожалению, отключить эту функцию невозможно. Если вы хотите использовать адрес ячейки или диапазона, вам придется ввести его вручную (не указывая элемент таблицы мышью).

Отмена имени

В Excel не предусмотрен специальный метод отмены имени. Иными словами, вы не сможете автоматически заменить имя ячейки или диапазона обычной ссылкой на этот элемент таблицы. Однако есть способ, с помощью которого можно заменить имя ссылкой. Для этого выберите **Сервис**⇒**Параметры**. Перейдите на вкладку **Переход** диалогового окна **Параметры** и выставьте флажок опции **Производить вычисления по правилам Lotus 1-2-3**. Затем нажмите <F2> для того, чтобы отредактировать формулу, содержащую одно или несколько имен ячеек или диапазонов. В результате формула будет содержать ссылки, а не имена ячеек или диапазонов (однако в строке формул будут все же отображены имена). Нажмите <Enter>, и редактирование ячеек будет окончено. Затем вернитесь в диалоговое окно **Параметры** и снимите флажок, отключив параметр **Производить вычисления по правилам Lotus 1-2-3**. Отредактированная таким образом ячейка не будет содержать имен.

Удаление имен

Если вам не нужно ранее назначенное имя, вы можете удалить его. При удалении имени диапазона, содержимое этого диапазона остается без изменений. Выберите **Вставка**⇒**Имя**⇒**Присвоить**. Появится диалоговое окно **Присвоение имени**. Выберите в списке имя, которое следует удалить, и щелкните на кнопке **Удалить**.



Удаляя имена, будьте предельно внимательны. Если имя используется в формуле, его удаление повлечет за собой ошибку в формуле (появится сообщение об ошибке #ИМЯ?). Было бы очень удобно, если бы в формуле программа заменяла удаленные имена ссылками на ячейки и диапазоны. Но Excel, к сожалению, не делает этого. Обнаружив сообщение об ошибке #ИМЯ?, вы можете отменить удаление имени. Для того чтобы восстановить удаленное вами имя, выберите **Правка**⇒**Отменить удаление**.

Удаление именованных ячеек и диапазонов

При удалении строк и столбцов, содержащих именованные ячейки или диапазоны, имена не удаляются (вопреки вашим ожиданиям). Каждое имя в таком случае будет иметь некорректную ссылку. Предположим, ячейка A1 на листе Лист1 носит имя *Interest*. Если вы удалите строку 1 или столбец A, *Interest* будет относиться к =Лист1!#ССЫЛКА! (ошибочная ссылка). Если имя *Interest* было использовано в формуле, появится сообщение об ошибке #ССЫЛКА!.

Для того чтобы избавиться от имени, относящегося к несуществующему диапазону или ячейке, необходимо удалить это имя вручную с помощью команды Вставка⇒Имя⇒Присвоить. Вы также можете переназначить это имя другой ячейке или диапазону.

Переназначение имени

Иногда бывает необходимо переназначить ранее заданное вами имя другой ячейке или диапазону. Для этого выберите Вставка⇒Имя⇒Присвоить. Появится диалоговое окно Присвоение имени. В списке диалогового окна выберите имя, которое следует переназначить, а затем отредактируйте запись в поле Формула. Вы можете также активизировать поле Формула, а затем выбрать любую ячейку или диапазон, указав этот элемент рабочего листа мышью.

Изменение имени

В Excel не предусмотрен специальный метод изменения имени. Если, создав имя, вы поняли, что его необходимо изменить (возможно, вы допустили ошибку при написании), вам придется создать новое имя, а старое — удалить. В диалоговом окне Присвоение имени выберите старое имя в списке имен, введите в поле ИМЯ новое имя и щелкните на кнопке Добавить. Затем выберите старое имя еще раз и щелкните на кнопке Удалить.

Если вы изменили имя в формулах, Excel не заменит старое имя новым автоматически. Для того чтобы найти все случаи использования старого имени в формулах и заменить старое имя новым, используйте команду Правка⇒Заменить.

Просмотр именованных диапазонов

Если вы измените масштаб изображения до 39% и меньше, вокруг именованных диапазонов вы увидите границы и имена, отображенные синим цветом (рис. 3.12). При печати границы имена не видны. Они предназначены только для того, чтобы вы могли видеть именованные диапазоны на экране.

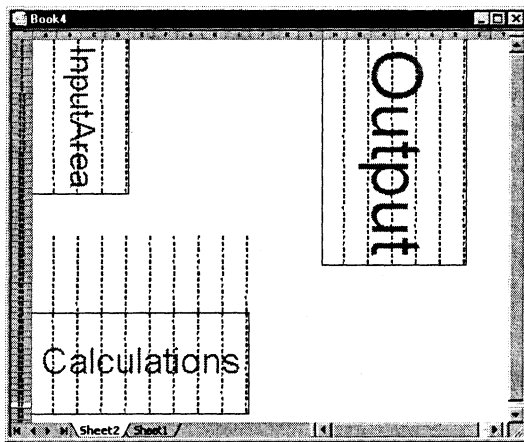


Рис. 3.12. Если вы измените масштаб изображения до 39% и меньше, Excel отобразит имена диапазонов



Это свойство характерно только в Excel 97 и последующих версиях.

Использование имен в диаграммах

При создании диаграммы каждая последовательность данных имеет соответствующую формулу РЯД. Формула РЯД содержит ссылки на диапазоны, используемые в диаграмме. Если в диаграмме используется именованный диапазон, вы можете отредактировать формулу РЯД, заменив ссылку именем диапазона.



Дополнительную информацию о диаграммах вы найдете в главе 17.

Поддержка имен ячеек и диапазонов

Excel автоматически поддерживает созданные вами имена ячеек и диапазонов по мере того, как вы редактируете или изменяете рабочий лист. В следующих примерах мы будем рассматривать рабочий лист Лист1, содержащий имя *MyRange*, заданное на уровне рабочей книги. Предположим, что имя *MyRange* относится к диапазону =Лист1!\$C\$3:\$E\$5 (диапазон состоит из девяти ячеек).

Вставка строки или столбца

Когда вы вставляете строку выше именованного диапазона или столбец слева от именованного диапазона, Excel автоматически изменяет ссылку на этот диапазон так, чтобы она соответствовала новому адресу диапазона. Например, если вы вставите новую строку 1, имя *MyRange* будет относиться к =Лист1!\$C\$4:\$E\$6.

Если же вы вставите новую строку или столбец внутрь именованного диапазона, этот диапазон расширится и будет включать в себя новые строки или столбцы. Например, если вы вставите новый столбец слева от столбца E, имя *MyRange* будет относиться к =Лист1!\$C\$3:\$F\$5.

Удаление строки или столбца

Когда вы удаляете строку выше именованного диапазона или столбец слева именованного диапазона, Excel автоматически изменяет ссылку на этот диапазон так, чтобы она соответствовала новому адресу диапазона. Например, если вы удалите строку 1, имя *MyRange* будет относиться к =Лист1!\$B\$3:\$D\$5.

Если же вы удалите строку или столбец внутри именованного диапазона, этот диапазон изменится соответствующим образом. Например, если вы удалите столбец D, имя *MyRange* будет относиться к =Лист1!\$C\$3:\$D\$5.

Если вы удалите все строки и ячейки, составляющие именованный диапазон, именованный диапазон все же будет существовать, однако ссылка на него будет содержать ошибку. Например, если вы удалите столбцы C, D и E, имя *MyRange* будет относиться к =Лист1!#ССЫЛКА!. Все формулы, в которых использовано это имя, также будут возвращать ошибки.

Вырезание и вставка именованного диапазона

Если вы вырезаете и вставляете именованный диапазон целиком, Excel изменяет ссылку соответствующим образом. Например, если вы переместите диапазон *MyRange* в другой участок листа, который начинается с ячейки A1, имя *MyRange* будет относиться к =Лист1!\$A\$1:\$C\$3. Если же вы вырежете и вставите только часть именованного диапазона, ссылка не изменится.

Проблемы при работе с именами

Имена необходимы для создания таблиц. Однако при работе с ними могут возникать некоторые проблемы. В этом разделе я расскажу вам о том, о чем вы не должны забывать, работая с именами.

Проблемы, возникающие при копировании рабочих листов

Как вы, вероятно, знаете, Excel позволяет копировать рабочие листы в пределах рабочей книги, а также в другую рабочую книгу. Прежде всего, давайте рассмотрим пример копирования рабочего листа в пределах рабочей книги. Если копируемый лист содержит имена, заданные на уровне рабочего листа, те же имена будут присутствовать и в копии этого листа, причем в ссылках будет использовано имя нового рабочего листа. Это именно то, чего вы ожидали. Если же рабочая книга содержит имя, заданное на уровне рабочей книги, и это имя относится к ячейке или диапазону копируемого рабочего листа, имя также будет присутствовать и в копии. Но это имя будет преобразовано в имя, заданное на уровне рабочего листа. А это вам, скорее всего, совсем не нужно.

Рассмотрим рабочую книгу, содержащую один лист (Лист1). В этой книге есть имя — *BookName*, — назначенное ячейке A1 на уровне рабочей книги. Кроме того, есть еще одно

имя — *Лист1!LocalName*, — назначенное ячейке A2 на уровне рабочего листа. Если вы сделаете копию листа Лист1 в пределах рабочей книги, новый лист будет называться Лист1 (2). После того как лист будет скопирован, вы обнаружите, что теперь рабочая книга содержит четыре имени, перечисленных в табл. 3.1.

Таблица 3.1. Имена в рабочей книге после копирования листа

Имя	Формула	Тип
BookName	=Лист1!\$A\$1	Уровень рабочей книги
Лист1!LocalName	=Лист1!\$A\$2	Уровень рабочего листа
Лист1 (2)!BookName	= 'Лист1 (2) ' ! \$A\$1	Уровень рабочего листа
Лист1 (2)!LocalName	= 'Лист1 (2) ' ! \$A\$2	Уровень рабочего листа

Такое увеличение количества имен при копировании листа не только сбивает с толку, но и может привести к ошибкам, найти причину возникновения которых бывает очень сложно. Если в рассматриваемом нами примере мы введем следующую формулу в скопированный лист, формула возвратит содержимое ячейки A1 скопированного листа:

=BookName

Иными словами, используется вновь созданное на уровне рабочего листа имя (а не исходное имя, заданное на уровне рабочей книги).

Если вы скопируете лист рабочей книги, содержащей имя диапазона, относящегося к нескольким листам, вместе с этим рабочим листом будет скопировано и имя диапазона. В поле Формула появится сообщение об ошибке #ССЫЛКА!.

Если же вы копируете лист в другую рабочую книгу, все имена исходной рабочей книги, относящиеся к ячейкам копируемого листа, будут скопированы в новую книгу. Это касается как имен, заданных на уровне рабочей книги, так и имен, заданных на уровне рабочего листа.



Копирование и вставка ячеек из одного рабочего листа в другой не приводит к копированию имен даже в том случае, если копируемый диапазон содержит именованные ячейки.

Вывод? Копируя листы рабочей книги, содержащей имена, будьте очень внимательны. Скопировав лист, проверьте все имена и удалите те, которые были скопированы случайно.

Проблемы, возникающие при удалении рабочих листов

При удалении листа, содержащего ячейки, используемые в имени, заданном на уровне рабочей книги, имя не удаляется. Оно остается в рабочей книге, но ссылка на ячейку или диапазон в поле Формула будет содержать ошибку.

На рис. 3.13 показано диалоговое окно Присвоение имени, в котором присутствует имя с ошибкой. Исходная рабочая книга содержала лист Лист1, в котором был именованный диапазон (имя диапазона — *MyRange* — задано на уровне рабочей книги) A1:F12. После удаления листа Лист1 имя *MyRange* осталось в рабочей книге. Однако в поле Формула диалогового окна Присвоение имени появилось следующее выражение:

=#ССЫЛКА!\$A\$1:\$F\$12

Насколько мне известно, имена с ошибочной ссылкой в рабочей книге вреда не приносят, однако такие имена все же лучше удалять.

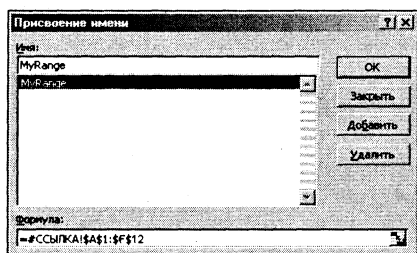


Рис. 3.13. Удаление листа, содержащего ячейку диапазона MyRange, приводит к появлению ошибочной ссылки

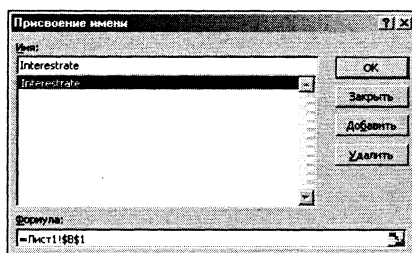


Рис. 3.14. Формально, имя InterestRate представляет собой именованную формулу, а не именованную ячейку

Имена диаграмм и объектов

Когда вы добавляете на рабочий лист диаграмму или любой другой объект, имя этой диаграмме или объекту назначается по умолчанию. Например, первая диаграмма рабочего листа получает имя *Диагр. 1*. Если вы добавляете фигуру (например, прямоугольник), имя отражает тип объекта (например, *Прямоуг. 3*).

Для того чтобы изменить имя объекта, выделите нужный объект, введите новое имя в поле **Имя** и нажмите **<Enter>**. Если же переименованию подлежит диаграмма, выделить необходимо весь объект диаграммы (область, содержащую диаграмму). Для этого нажмите **<Ctrl>** и одновременно щелкните на диаграмме мышью.

Поле **Имя** в Excel устроено не вполне логично в том смысле, что с его помощью можно переименовать объект, а список объектов в этом окне не отображается. Для того чтобы выбрать объект, используя поле **Имя**, вы должны ввести в это окно точное имя объекта. Следует отметить, что и в диалоговом окне **Присвоение имени** вы не найдете списка имен объектов.

Ключ к пониманию термина “имя”

Пользователи Excel часто обращаются к *именованным ячейкам* и *именованным диапазонам*. В этой главе я постоянно использовал эти понятия. Но эти термины не совсем точны.

Вот ключ к пониманию термина “имя”:

Присваивая имя, вы в действительности создаете именованную формулу. В отличие от обыкновенной формулы, именованная формула хранится не в ячейке, а в памяти Excel.

Я, разумеется, не “открыл Америку”, раскрыв вам этот секрет. И все же, если вы будете исходить из такой формулировки, вам будут легче овладеть сложными приемами, о которых я расскажу вам в следующих разделах.

Когда вы работаете в диалоговом окне **Присвоение имени**, поле **Формула** содержит формулу, а поле **Имя** — имя этой формулы. Обратите внимание на то, что содержимое поля **Формула** всегда начинается знаком равенства, что и делает эту запись формулой.

Как видно из рис. 3.14, рабочая книга содержит имя — *InterestRate*, — присвоенное ячейке B1 листа Лист1. В поле **Формула** присутствует следующая запись:

```
=Лист1!$B$1
```


Когда вы используете имя *InterestRate*, Excel, в действительности, оценивает формулу, которая носит это имя, и возвращает результат. Например, вы можете ввести в ячейку следующую формулу:

```
=InterestRate*1,05
```

Оценивая формулу, Excel прежде всего анализирует формулу, которая носит имя *InterestRate* (которое существует только в памяти программы, а не в ячейке). Затем Excel умножает результат именованной формулы на 1,05 и возвращает результат. Разумеется, приведенная выше формула равноценна формуле, в которой вместо имени используется ссылка на ячейку:

```
=Лист1!$B$1*1,05
```

Сейчас вы, вероятно, думаете над тем, можно ли создать такую именованную формулу, которая бы не содержала ссылок на ячейки. Ответ на этот вопрос вы найдете в следующем разделе.

Присвоение имени константе

Рассмотрим таблицу, с помощью которой создается счет и вычисляется сумма налога с оборота на основе объема продаж. Обычно ставка налога с оборота вставляется в ячейку, затем ссылка на эту ячейку используется в различных формулах. Для того чтобы было понятно, что за величина указана в ячейке, вы, вероятно, дадите этой ячейке имя *SalesTax*.

Но есть еще один способ. На рис. 3.15 представлен результат выполнения следующих инструкций:

1. Выберите Вставка⇒Имя⇒Присвоить (или нажмите комбинацию клавиш <Ctrl+F3>). Появится диалоговое окно Присвоение имени.
2. Введите имя (в данном случае *SalesTax*) в поле Имя.
3. Перейдите в поле Формула, удалите содержащуюся в этом поле запись и вместо нее введите простую формулу, например =0,075.
4. Щелкните на кнопке ОК для того, чтобы закрыть диалоговое окно.

В результате выполнения приведенных инструкций создается именованная формула, в которой не используются ссылки на ячейки. Для того чтобы проверить все вышесказанное на практике, введем в любую ячейку такую формулу:

```
=SalesTax
```

Эта простая формула возвращает 0,075 — результат именованной формулы *SalesTax*. Поскольку именованная формула всегда возвращает один и тот же результат, ее можно считать именованной константой. Эта константа может быть использована и в более сложных формулах, как например:

```
=A1*SalesTax
```

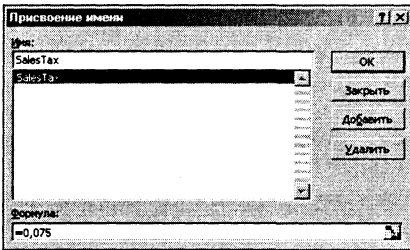


Рис. 3.15. Назначение имени константе

SalesTax — имя, заданное на уровне рабочей книги. Поэтому вы можете использовать его в любом рабочем листе книги.

Присвоение имени текстовым константам

В предыдущем примере константа представляла собой числовое значение. Однако константа может также состоять из текста. Например, вы можете определить константу для на-

звания компании. Для создания следующей формулы, которая носит имя *MS*, используется диалоговое окно **Присвоение имени**:

```
"Microsoft Corporation"
```

Теперь в любой ячейке можно использовать такую формулу, как:

```
"Annual Report:"&MS
```

Эта формула возвращает текст *Annual Report: Microsoft Corporation*.



Имена, не относящиеся к диапазонам, не появляются в поле **Имя** или диалоговом окне **Переход** (которое можно вызвать нажатием клавиши <F5>). И это логично, поскольку константы не присутствуют в ячейках рабочего листа. Константы отображаются в диалоговом окне **Вставка имени**, и это тоже логично, поскольку имена констант используются в формулах.

Как вы, вероятно, догадываетесь, значение константы может быть изменено в любой момент в диалоговом окне **Присвоение имени**. Все, что нужно сделать, — это изменить значение в поле **Формула**. После того как диалоговое окно будет закрыто, Excel проведет пересчет всех формул, в которых используется данная константа, с использованием нового значения.

Этот прием очень эффективен во многих ситуациях. Однако изменение значения занимает некоторое время. Если константу расположить в ячейке, изменить ее значение будет значительно легче. Если значение действительно является константой, изменять его вам не придется.

Использование функций рабочего листа в именованных формулах

На рис. 3.16 приведен еще один пример именованной формулы. В этом случае формула носит имя *ThisMonth* и выглядит так:

```
=МЕСЯЦ(СЕГОДНЯ())
```

В формуле, которая демонстрируется на рис. 3.16, используются две функции рабочего листа. Функция **СЕГОДНЯ** возвращает текущую дату, а функция **МЕСЯЦ** — номер месяца аргумента даты. Следовательно, вы можете ввести в ячейку следующую формулу, которая будет возвращать номер текущего месяца. Например, если текущий месяц — апрель, формула будет возвращать 4.

```
=ThisMonth
```

Более эффективная именованная формула будет возвращать название месяца в текстовой форме. Такая формула под именем *MonthName* будет выглядеть следующим образом:

```
=ТЕКСТ(СЕГОДНЯ(); "ММММ")
```

Теперь эту формулу можно ввести в ячейку, и она будет возвращать название текущего месяца в текстовой форме. В апреле формула будет возвращать текст **Апрель**.

```
=MonthName
```

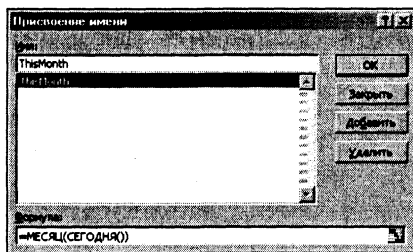


Рис. 3.16. Создание именованной формулы, в которой используются функции рабочего листа

Использование ссылок на ячейки и диапазоны в именованных формулах

На рис. 3.17 приведен очередной пример создания именованной формулы. На этот раз в формуле используется ссылка на ячейку. Эта формула под именем *FirstChar* возвращает первый символ, содержащийся в ячейке A1 листа Лист1. Именованная формула выглядит так:

=ЛЕВСИМВ(Лист1!\$A\$1;1)

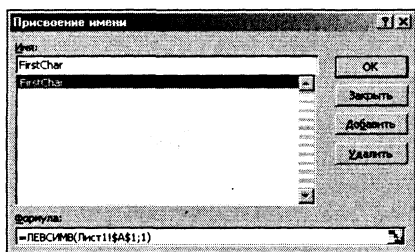


Рис. 3.17. Создание именованной формулы, в которой используется ссылка на ячейку

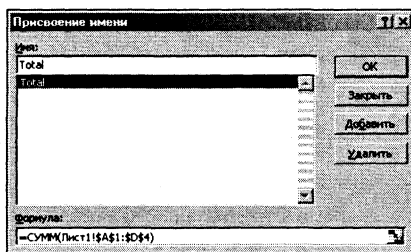


Рис. 3.18. Создание именованной формулы, в которой используется ссылка на диапазон

Создав такую именованную формулу, вы можете ввести эту формулу в ячейку. Формула всегда возвращает первый символ, содержащийся в ячейке A1 листа Лист1.

=FirstChar

В следующей именованной формуле, которую мы рассмотрим, используется ссылка на диапазон. На рис. 3.18 показано диалоговое окно Присвоение имени, в котором задана именованная формула (под именем *Total*).

=СУММ(Лист1!\$A\$1:\$D\$4)

Создав такую именованную формулу, вы можете ввести эту формулу в ячейку любого рабочего листа. Формула возвращает сумму значений ячеек диапазона A1:D4 листа Лист1.

=Total

Обратите внимание на то, что в двух предыдущих формулах использованы абсолютные ссылки. По умолчанию все ссылки на ячейки и диапазоны, используемые в именованных формулах, являются абсолютными и включают название рабочего листа. Прочитав следующий раздел вы узнаете, какие интересные именованные формулы можно создавать, применяя относительные ссылки.

Использование именованных формул с относительными ссылками

Я уже говорил о том, что когда вы используете диалоговое окно Присвоение имени для создания именованной формулы со ссылками на ячейки или диапазоны, ссылки в поле Формула всегда абсолютны и включают название рабочего листа. В этом разделе я расскажу о том, как использовать в именованных формулах относительные ссылки на ячейки и диапазоны.

Использование относительной ссылки на ячейку

Начнем с простого примера. Для того чтобы создать именованную формулу с относительной ссылкой на ячейку, выполните следующее:

1. Откройте пустой рабочий лист.
2. Выделите ячейку A1 (этот шаг очень важен).
3. Выберите Вставка⇒Имя⇒Присвоить. Появится диалоговое окно Присвоение имени.
4. В поле Имя введите **CellToRight**.
5. Удалите содержимое поля **Формула** и введите следующую формулу (не указывая ячейку рабочего листа мышью):

```
=Лист1!B1
```

6. Щелкните на кнопке ОК для того, чтобы закрыть диалоговое окно Присвоение имени.
7. Введите что-либо в ячейку B1.
8. В ячейку A1 введите следующую формулу:

```
=CellToRight
```

Вы обнаружите, что формула, расположенная в ячейке A1, возвращает содержимое ячейки B1.

Затем скопируйте формулу ячейки A1 в несколько ячеек, расположенных под ней. Теперь введите произвольные значения в столбец B. Вы увидите, что формула, расположенная в столбце A, возвращает содержимое ячейки, расположенной справа. Иными словами, именованная формула (*CellToRight*) имеет относительный характер.

Имя *CellToRight* может быть использовано в любой ячейке (а не только в ячейках столбца A). Например, если вы введете **CellToRight** в ячейку D12, эта ячейка будет возвращать содержимое ячейки E12.

Для того чтобы убедиться в том, что в именованной формуле *CellToRight* действительно используется относительная ссылка на ячейку, активизируйте любую ячейку, кроме A1, и откройте диалоговое окно Присвоение имени (рис. 3.19). В списке выберите *CellToRight* и обратите внимание на содержимое поля **Формула**. Вы увидите, что формула изменяется в зависимости от того, какая ячейка активна в данный момент. Например, если активна ячейка E5, формула в диалоговом окне будет выглядеть так:

```
=Лист1!F5
```

Если вы используете имя *CellToRight* в другом рабочем листе, формула все равно будет ссылаться на ячейку, расположенную справа. И это будет ячейка с соответствующим адресом на листе Лист1. Это происходит потому, что именованная формула содержит ссылку на рабочий лист. Для того чтобы формула могла работать на любом рабочем листе, измените ее, следуя инструкциям:

1. Активизируйте ячейку A1 рабочего листа Лист1.
2. Выберите Вставка⇒Имя⇒Присвоить. Появится диалоговое окно Присвоение имени.
3. В списке диалогового окна Присвоение имени выберите пункт *CellToRight*.

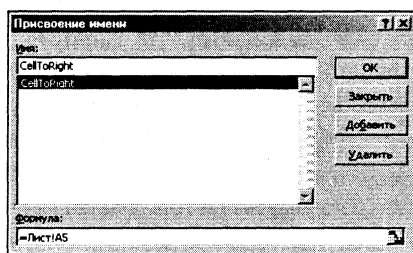


Рис. 3.19. Формула под именем *CellToRight* меняется в зависимости от того, какая ячейка активна в данный момент

- Удалите содержимое поля **Формула** и введите следующую формулу:
=!B1
- Щелкните на кнопке **ОК** для того, чтобы закрыть диалоговое окно **Присвоение имени**.
- Изменив ссылку таким образом, вы убедитесь, что теперь формула под именем *CellToRight* корректно работает в любом листе рабочей книги.



Именованная формула не будет работать, если вы используете ее в формуле, расположенной в столбце IV, поскольку в этом случае формула ссылается на несуществующую ячейку (справа от столбца IV нет столбца).

Использование относительной ссылки на диапазон

Следующий пример основан на предыдущем. Я покажу вам, как создать именованную формулу, которая возвращает сумму 10 ячеек, расположенных справа от определенной ячейки. Для того чтобы создать такую именованную формулу, следуйте инструкциям:

- Активизируйте ячейку A1.
- Выберите **Вставка**⇒**Имя**⇒**Присвоить**. Появится диалоговое окно **Присвоение имени**.
- В поле **Имя** введите **Sum10Cells**.
- Введите следующую формулу в поле **Формула**:

```
=СУММ(!B1:!K1)
```

Создав именованную формулу, вы можете ввести следующую формулу в любую ячейку любого листа, и эта формула будет возвращать сумму 10 ячеек, расположенных справа:

```
=Sum10Cells
```

Например, если вы введете такую формулу в ячейку D12, она будет возвращать сумму значений диапазона E12:N12, состоящего из 10 ячеек.

Не забывайте о том, что когда вы создавали именованную формулу, активной была ячейка A1. Следовательно, относительные ссылки в определении формулы заданы относительно ячейки A1. Обратите внимание и на то, что в формуле не использовано имя рабочего листа. Отсутствие имени рабочего листа (при условии, что в формуле присутствует восклицательный знак) приводит к тому, что формулу можно использовать в любом листе.

Если вы активизируете ячейку D12, а затем откроете диалоговое окно **Присвоение имени** в поле **Формула** для формулы под именем *Sum10Cells* вы увидите следующую запись:

```
=СУММ(!E12:!N12)
```



Именованная формула *Sum10Cells* не будет работать, если вы введете ее в ячейку, расположенную в столбце, следующем за столбцом IL. В таком случае формула становится ошибочной, поскольку в ней появляется ссылка на несуществующую ячейку, расположенную за пределами столбца IV.

Использование смешанных ссылок на диапазоны

Как я уже упоминал в главе 2, ссылка может быть абсолютной, относительной или смешанной. Смешанная ссылка на ячейку может состоять из:

- ♦ Абсолютной ссылки на столбец и относительной ссылки на строку (например, \$A1)
- ♦ Относительной ссылки на столбец и абсолютной ссылки на строку (например, A\$1)

Как вы, вероятно, догадываетесь, смешанные ссылки могут быть использованы в именованных формулах. Давайте попробуем использовать такую ссылку на практике. Активируйте ячейку B1. В диалоговом окне Присвоение имени создайте формулу под именем *FirstInRow*. Определение формулы должно выглядеть так:

```
=!$A1
```

В этой формуле использована абсолютная ссылка на столбец и относительная ссылка на строку. Следовательно, формула всегда возвращает значение, заданное в столбце A. А строка зависит от того, в какой строке вы введете формулу. Например, если вы введете следующую формулу в ячейку F12, эта формула возвратит значение ячейки A12:

```
=FirstInRow
```



Формулу *FirstInRow* нельзя использовать в столбце A, поскольку будет создана циклическая ссылка. Формула в этом случае ссылается на саму себя.

Решение сложных задач с помощью формул

В этом разделе я приведу несколько примеров решения более сложных задач с использованием имен. Все эти примеры предполагают, что вы уже знакомы со всеми приемами, описанными в предыдущих разделах этой главы.

Функция ДВССЫЛ и именованный диапазон

Функция ДВССЫЛ позволяет определять адрес ячейки косвенно. Например, если ячейка A1 содержит текст *C45*, формула будет возвращать содержимое ячейки C45:

```
=ДВССЫЛ(A1)
```

Функция ДВССЫЛ может применяться и в работе с именованными диапазонами. На рис. 3.20 представлена таблица с 12 именами диапазонов, соответствующими названиям Месяцев года. Например, имя *January* относится к диапазону B2 : E2. Ячейка B16 содержит такую формулу:

```
=СУММ(ДВССЫЛ(A16))
```

	A	B	C	D	E	F	G
1		North	South	West	East		
2	January	48	313	353	251		
3	February	354	151	47	190		
4	March	138	257	254	97		
5	April	244	110	308	378		
6	May	103	407	475	253		
7	June	24	222	45	256		
8	July	85	196	24	497		
9	August	286	121	431	125		
10	September	333	455	201	335		
11	October	81	89	450	322		
12	November	260	329	475	358		
13	December	445	105	165	356		
14							
15							
16	March	746					
17							
18							

Рис. 3.20. Использование функции ДВССЫЛ при работе с именованным диапазоном

На практике эта формула возвращает сумму именованного диапазона, который в текстовой форме задан в ячейке A16.



В Excel 97 и последующих версиях программы вы можете использовать команду Данные⇒Проверка. С помощью этой команды ячейку A16 можно преобразовать в раскрывающийся список. Используйте опцию Тип данных диалогового окна Проверка вводимых значений для того, чтобы задать диапазон A2:A13 в качестве источника данных. Это позволит пользователю выбирать название месяца из списка. Сумма для выбранного месяца появляется в ячейке B16.

Используя функцию ДВССЫЛ, вы можете ссылаться и на имена уровня рабочего листа. Предположим, что вы работаете с рабочими листами Region 1, Region 2 и т.д. Каждый лист содержит имя *TotalSales*, заданное на уровне рабочего листа. Следующая формула находит значение в соответствующем рабочем листе при использовании имени, заданного в ячейке A1:

```
=ДВССЫЛ(A1&"!TotalSales")
```

Использование функции ДВССЫЛ для создания именованного диапазона с фиксированным адресом

В Excel можно создать имя, которое бы всегда относилось к одной и той же ячейке или диапазону даже в том случае, если вы вставите новые строки или столбцы. Предположим, вы хотите создать диапазон под именем *UpperLeft*, который бы всегда относился к ячейке A1. Если вы создадите имя обычным способом, вставка новой строки 1 приведет к тому, что диапазон *UpperLeft* будет относиться к ячейке A2. Вставка нового столбца приведет к тому, что диапазон *UpperLeft* будет относиться к ячейке B2. Для создания именованного диапазона, который имеет фиксированный адрес, необходимо задать именованную формулу, в поле Формула которой будет следующее выражение:

```
=ДВССЫЛ (" $A$1 ")
```

Если вы создадите такую формулу, диапазон *UpperLeft* будет относиться к ячейке A1 даже в том случае, если вы вставите новые строки или столбцы. В предыдущей формуле функция ДВССЫЛ позволяет задавать адрес ячейки косвенно с помощью текстового аргумента. Поскольку аргумент заключен в кавычки, он никогда не меняется.



В этой именованной формуле используется функция, и потому формула не появляется ни в диалоговом окне Переход, ни в поле Имя.

Использование массивов в именованных формулах

Массив — это набор элементов. Массив может быть вертикальным и состоять из одного столбца, или горизонтальным и состоять из одной строки. Массивом может быть и совокупность нескольких столбцов и строк.



В части IV данной книги подробно описаны массивы и формулы массивов. Однако эта тема имеет отношение и к именам.

Задать массив можно с помощью скобок. Каждый элемент массива отделяется запятой или точкой с запятой. Запятая используется для разделения элементов, расположенных горизонтально, а точка с запятой — для разделения элементов, расположенных вертикально.

В диалоговом окне Присвоение имени можно создать формулу под именем *MonthNames*, которая имеет такое определение:

```
={"Jan"; "Feb"; "Mar"; "Apr"; "May"; "Jun"; "Jul"; "Aug"; "Sep"; "Oct"; "Nov"; "Dec"}
```

Эта формула определяет массив, состоящий из 12 текстовых строк, расположенных горизонтально.



Вводя такую формулу, не забудьте поставить скобки. В диалоговое окно Присвоение имени формула массива вводится не так, как в ячейку.

Определив формулу *MonthNames*, вы можете использовать ее в других формулах. Однако в конечной формуле необходимо обозначить, какой из элементов массива используется. Для этого лучше всего использовать функцию ИНДЕКС. Следующая формула, например, возвращает текст *Aug*:

```
=ИНДЕКС (MonthNames ; 8)
```

Вы можете отобразить и весь массив, состоящий из 12 элементов, но для этого необходимы 12 смежных ячеек. Для того чтобы ввести 12 элементов массива в диапазон A1:L1, следуйте приведенным ниже инструкциям:

1. В диалоговом окне Присвоение имени создайте формулу под именем *MonthNames*.
2. Выделите диапазон A1:L1.
3. В строку формул введите **=MonthNames**.
4. Нажмите комбинацию клавиш <Ctrl+Shift+Enter>.

Получив команду <Ctrl+Shift+Enter>, Excel вставляет формулу массива в выделенный диапазон. В данном случае формула введена в 12 смежных ячеек (рис. 3.21). Программа заключает формулу массива в скобки, напоминая вам, что это — специфическая формула. Рассмотрев любую из 12 ячеек диапазона A1:L1, вы увидите, что формула выглядит так:

```
{=MonthNames}
```

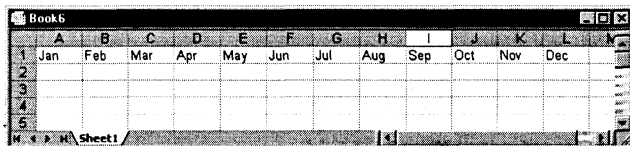


Рис. 3.21. Именованную формулу, которая содержит массив из 12 элементов, можно разместить в 12 смежных ячейках

Создание формулы с динамическим именем

Динамически именованная формула — это формула, относящаяся к диапазону, размер которого не фиксирован. Вероятно, осмыслить новое понятие не просто. Поэтому мы рассмотрим пример.

January	1,043		
February	1,123		
March	1,321		
April	2,244		
May	1,233		
June			
July			
August			
September			
October			
November			
December			

Рис. 3.22. Динамически именованная формула может быть использована для ввода данных в столбце В

Обратите внимание на таблицу, представленную на рис. 3.22. Эта таблица содержит сведения об объеме продаж за период с января по май.

Предположим, вы хотите присвоить имя *SalesData* данным, расположенным в столбце В. При этом вы не хотите, чтобы имя относилось к пустым ячейкам. Иными словами, ссылка на диапазон *SalesData* будет меняться каждый месяц, по мере ввода новых данных. Разумеется, вы могли бы менять ссылку вручную каждый месяц с помощью диалогового окна Присвоение имени. Но гораздо эффективнее применить динамически именованную формулу, которая будет автоматически меняться с каждым последующим вводом данных.

Прежде, чем задать динамически именованную формулу, создайте таблицу, показанную на рис. 3.22. Затем выполните следующие действия:

1. Вызовите диалоговое окно Присвоение имени.
2. В поле Имя введите **SalesData**.
3. В поле Формула введите следующую формулу:
`=СМЕЩ(Лист1!B1;0;0;СЧЕТЗ(Лист1!$B:$B);1)`
4. Щелкните на кнопке ОК для того, чтобы закрыть диалоговое окно Присвоение имени.

В результате выполнения инструкции была создана формула, в которой используются функции СМЕЩ и СЧЕТЗ. Для того чтобы проверить, как функционирует эта формула, введите следующую формулу в любую ячейку, которая не расположена в столбце В:

`=СУММ(SalesData)`

Эта формула возвратит сумму значений столбца В. Обратите внимание на то, что *SalesData* не отображается ни в поле Имя, ни в диалоговом окне Переход. Для того чтобы выбрать диапазон, вы можете открыть диалоговое окно Переход и ввести **SalesData** вручную.

Сейчас вы, наверное, думаете, зачем морочить себе голову. Ведь и простая формула, приведенная ниже, сделает все, что нужно:

`=СУМ(В:В)`

Эффективность работы динамически именованных формул очевидна при создании диаграмм. С помощью таких формул можно создать диаграмму с наборами данных, которые автоматически изменяются при вводе новых данных.



В главе 17 вы найдете пример создания динамической диаграммы.

Резюме

В этой главе мы рассмотрели понятие имени. Я рассказал вам, как создавать и изменять имена. Вы узнали, чем отличается имя, заданное на уровне рабочей книги, от имени, заданного на уровне рабочего листа. В этой главе было приведено множество примеров использования имен в рабочей книге. Кроме того, я дал вам ключ к пониманию термина “имя”: каждое имя, по сути, является именованной формулой.

Глава 4 служит введением к теме функций рабочего листа Excel и содержит обзор основных функций.

Часть II

Использование функций в формулах

ГЛАВА 4

“Введение в функции Excel”

ГЛАВА 5

“Работа с текстом”

ГЛАВА 6

“Работа с данными даты и времени”

ГЛАВА 7

“Способы суммирования
и подсчета данных”

ГЛАВА 8

“Поиск данных”

ГЛАВА 9

“Базы данных и списки”

ГЛАВА 10

“Дополнительные вычисления”

Глава 4

Введение в функции Excel

В этой главе...

- ◆ Что такое функция?
- ◆ Типы аргументов функций
- ◆ Способы введения функции в формулу
- ◆ Категории функций
- ◆ Резюме

Каждый, кто хочет в совершенстве овладеть искусством работы с формулами в Excel, должен досконально изучить тему функций в Excel. Эта глава содержит обзор функций, которые могут использоваться в формулах.

Что такое функция?

Функция — встроенный инструмент, используемый в формуле. В обычной функции (например, СУММ) присутствует один и более аргументов. Функция возвращает результат. Функция СУММ, например, принимает аргумент диапазона, а затем возвращает сумму значений этого диапазона.

Функции вам пригодятся, поскольку они:

- ◆ Упрощают формулы
- ◆ Позволяют производить вычисления, которые невозможно осуществить без них
- ◆ Ускоряют выполнение некоторых задач редактирования
- ◆ Позволяют налагать условия на выполнение формул, что дает последним элементарную способность к принятию решений

Примеры, приведенные в следующих разделах, иллюстрируют эти положения.

Упрощение формулы

Использование встроенных функций позволяет существенно упростить формулу. Например, вам нужно вычислить среднее арифметическое значений десяти ячеек (A1:A10). Без применения функций вам придется создать примерно следующую формулу:

$$=(A1+A2+A3+A4+A5+A6+A7+A8+A9+A10)/10$$

Громоздко, не так ли? Более того, если в последствии диапазон суммируемых значений будет увеличен, вам придется редактировать эту формулу. Такую формулу можно заменить более простой, в которой будет использована одна из встроенных функций рабочего листа Excel. В приведенной ниже формуле используется функция СРЗНАЧ:

$$=СРЗНАЧ(A1:A10)$$

Вычисления, невозможные без функций

Функции позволяют осуществлять такие вычисления, которые без их использования невозможны. Например, вам нужно выявить наибольшее из значений диапазона. Формула, в которой не используются функции, не сможет решить такую задачу. В следующем примере функция МАКС используется для возвращения формулой наибольшего из значений диапазона A1 : D100:

=МАКС (A1 : D100)

Ускорение выполнения задач редактирования

В некоторых случаях функции позволяют избежать редактирования вручную. Предположим, что в ячейках диапазона A1 : A1000 содержится 1000 имен, каждое из которых полностью записано символами верхнего регистра. Увидев такой список имен, ваш начальник заявил, что в письме, которое должно содержать эти имена, запись имен символами верхнего регистра совершенно неприемлема. Например, вместо JOHN F. CRANE, должно быть John F. Crane. Вы конечно можете посвятить остаток рабочего дня редактированию каждого из имен вручную. Однако намного проще создать формулу, в которой используется функция ПРОПНАЧ, которая преобразует текст ячейки A1 в нужный формат:

=ПРОПНАЧ (A1)

Введите эту формулу в ячейку B1, а затем скопируйте ее в следующие 999 строк. Затем выделите диапазон B1 : B1000 и скопируйте его в буфер, применив команду Правка⇒Копировать. Затем активизируйте ячейку A1 и преобразуйте формулы в значения, выбрав Правка⇒Специальная вставка (с опцией Значения). И наконец, удалите столбец B. Таким образом с помощью функции Excel вы сэкономили несколько часов драгоценного времени.

Способность к принятию решений

Функции дают формулам способность к принятию решений. Предположим, у вас есть таблица, предназначенная для вычисления комиссионных с продажи. Если менеджер продает товара на сумму, превышающую \$100 000, ставка комиссионного вознаграждения достигает 7,5%. В противном случае комиссионное вознаграждение составляет 5,0% от суммы продаж. Если вы не примените функцию, вам придется создавать две разные формулы, и для каждого отдельного случая решать, какую формулу следует использовать. Обратите внимание на следующую формулу, в которой используется функция ЕСЛИ. Формула проверяет значение, заданное в ячейке A1, а затем производит соответствующие вычисления:

=IF (A1<100000; A1*7,5%)

Еще немного о функциях

Как я уже говорил, в Excel присутствует более 300 функций. Если вам этого недостаточно, то можете приобрести дополнительные специальные функции у сторонних производителей и даже создать свои собственные функции (используя VBA).



Если вы готовы создать свои пользовательские функции, обратитесь к части IV этой книги.

В таком огромном количестве функций несложно запутаться. А постоянно использовать вы, скорее всего, будете лишь десяток основных формул. На практике вы убедитесь в том, что диалоговое окно **Мастер функций** (описанное далее в этой главе) позволяет легко найти и вставить нужную вам функцию, даже если вы нечасто используете функции.



В приложении Б вы найдете полный список функций Excel с кратким описанием каждой из них.

Типы аргументов функций

Рассмотрев приведенные выше примеры, вы, наверняка, заметили, что в каждой из функций используется пара скобок. Информация, приведенная в скобках, называется *аргументами* функции. Количество аргументов в разных функциях не одинаково. Функция может:

- ◆ Не иметь ни одного аргумента
- ◆ Иметь один аргумент
- ◆ Иметь определенное количество аргументов
- ◆ Иметь неограниченное количество аргументов
- ◆ Иметь аргументы, задавать которые не обязательно

Например, функция СЛЧИС, которая возвращает случайное число между 0 и 1, не имеет аргументов. Но даже при отсутствии аргументов, необходимо ввести пустую пару скобок:

=СЛЧИС ()

Если функция имеет больше одного аргумента, аргументы отделяются друг от друга точками с запятыми. В функции НАИБОЛЬШИЙ, которая возвращает n-ое наибольшее значение из множества данных, используются два аргумента. Первый аргумент задает диапазон значений, а второй — значение для n. Приведенная ниже формула возвращает третье по величине значение ячеек диапазона A1 : A100:

=НАИБОЛЬШИЙ (A1 : A100 ; 3)



Точка с запятой — не единственный символ, который может быть использован для разделения аргументов функции. Для этой цели можно применять, например, запятую. Символ определяется опцией Разделитель апплета Язык и региональные стандарты, вызвать которое можно из папки Панель управления.

В примерах, приведенных в начале этой главы, аргументами служили ссылки на ячейки и диапазоны. Однако, следует заметить, что в отношении аргументов функций Excel достаточно демократична. В следующих разделах я расскажу вам о других типах аргументов функций.

Имена в качестве аргументов

Как вы уже знаете, в качестве аргументов функций могут использоваться ссылки на ячейки и диапазоны. Производя вычисления, Excel просто использует текущее содержимое ячейки или диапазона. Функция СУММ возвращает сумму своих аргументов (или аргумента). Для того чтобы вычислить сумму значений диапазона A1 : A20, можно записать такую формулу:

=СУММ (A1 : A20)

Как вы, вероятно, и ожидали, если диапазон A1 : A20 имеет имя (например, Sales), это имя может быть использовано вместо ссылки на диапазон:

=СУММ(Sales)

В помощь пользователям, работавшим в Lotus 1-2-3

Если вам приходилось работать в любой из версий Lotus 1-2-3 (или Quattro Pro компании Corel), вы, наверное, помните, что в этих электронных таблицах перед именем функции должен стоять символ "коммерческого at" (@). Excel — достаточно сообразительная программа, способная распознать функцию и без отличительного символа.

От старых привычек трудно избавиться, и поэтому Excel воспринимает символ @ при вводе формул. Как только вы введете формулу, символ "коммерческое at" будет удален.

В качестве оператора ссылки на диапазон в Lotus 1-2-3 и Quattro Pro используются две точки (.), например, A1..A10. Excel позволяет задавать такой оператор при вводе формулы. Однако, как только формула будет введена, Excel заменит его собственным оператором ссылки на диапазон — двоеточием (:).

Но на этом уступки пользователям, которые еще не успели адаптироваться к Excel, заканчиваются. В Excel могут применяться только стандартные имена функций; программа не воспринимает и не преобразует имена, принятые в аналогичных средствах для создания электронных таблиц. Например, если вы введете используемую в Lotus 1-2-3 функцию @AVG, Excel выдаст сообщение об ошибке (в Excel аналогичная функция носит название СРЗНАЧ). Дополнительные сведения о совместимости Excel и Lotus 1-2-3 вы найдете в приложении А.



За подробной информацией о том, как назначать и использовать имена, обратитесь к главе 3.

Задание строки или столбца в качестве аргумента

Иногда в качестве аргумента функции удобно использовать строку или столбец полностью. Следующая формула, например, возвращает сумму всех значений столбца B:

=СУММ(B:B)

Ссылки на целую строку или столбец особенно эффективны в тех случаях, если диапазон, сумму значений которого следует вычислить, постоянно изменяется (например, если вы периодически вносите в таблицу новые данные об объемах продаж). Прежде, чем использовать ссылку на целую строку или столбец, убедитесь в том, что эта строка или столбец не содержит данных, которые не должны быть включены в сумму.

Вы, наверное, думаете, что суммирование такого большого диапазона значений (столбец, например, состоит из 65 536 ячеек) может увеличить время вычисления. Не беспокойтесь. Excel помнит, в каком столбце и какой строке располагается последнее значение. Поэтому, вычисляя результат формулы с использованием ссылки на целую строку или столбец, программа не будет использовать ячейки, выходящие за пределы последнего использованного столбца и строки.

Точные значения в качестве аргументов

Точный аргумент — это цифровое значение или текстовая строка, которые задаются непосредственно в функции. Например, функция КОРЕНЬ, которая возвращает значение квадратного корня, имеет один аргумент. В следующем примере в качестве аргумента функции используется точное значение:

=КОРЕНЬ (225)

В некоторых случаях использование точного аргумента лишает формулу всякого смысла. Такая формула всегда возвращает одно и то же значение, поэтому ее можно просто заменить цифрой 15. Применять такую формулу можно разве что в случае, когда вы хотите дать понять, что 15 — это квадратный корень из 225.

Использование точного аргумента оправдано, если формула имеет два и более аргументов. Функция ЛЕВСИМВ (которая имеет два аргумента) возвращает крайние левые символы текстовой строки (первый аргумент функции). Второй аргумент определяет число символов. Если ячейка A1 содержит текст Budget, следующая формула возвращает первую букву, то есть B:

=ЛЕВСИМВ (A1 ; 1)

Математические выражения в качестве аргументов

Excel позволяет использовать в качестве аргументов математические *выражения*. Такие выражения следует рассматривать как формулы в формуле. Если в качестве аргумента используется математическое выражение, Excel сначала оценивает значение этого выражения, а затем использует это значение в качестве значения аргумента. Рассмотрим такой пример:

=КОРЕНЬ ((A1^2) + (A2^2))

В этой формуле используется функция КОРЕНЬ, в качестве единственного аргумента которой выступает следующее математическое выражение:

(A1^2) + (A2^2)

Вычисляя значение формулы, Excel сначала оценивает значение выражения в аргументе функции, а затем извлекает квадратный корень из значения этого выражения.

Функции в качестве аргументов

Мы уже выяснили, что в качестве аргументов функций могут выступать математические выражения. И не удивительно, что эти выражения в свою очередь могут содержать другие функции. Функции, которые используются внутри других функций формулы, называются *вложенными*. Прежде всего Excel оценивает значение выражения с наибольшей глубиной вложения, а затем — все остальные. Рассмотрим следующий пример вложенной функции:

=SIN (РАДИАНЫ (B9))

Функция РАДИАНЫ преобразует градусы в радианы — единицы измерения, которые используются во всех тригонометрических функциях Excel. Если в ячейке B9 задан угол в градусах, функция РАДИАНЫ преобразует градусы в радианы, а затем функция SIN вычисляет синус этого угла.



Формула может содержать не более семи уровней вложенных функций. Если вы превысите допустимый предел, Excel выдаст сообщение об ошибке. В большинстве случаев, использование более семи уровней вложений не имеет смысла. Пользователи часто превышают допустимый предел при попытке создать сложные формулы, состоящие из вложенных функций ЕСЛИ.

Массивы в качестве аргументов

В качестве аргумента функции может быть использован массив. Массив представляет собой ряд значений, разделенных точками с запятыми и заключенных в скобки. В приведенной ниже формуле используется функция ИЛИ, в качестве аргумента которой использован массив. Формула возвращает ИСТИНА, если ячейка A1 содержит значение 1, 3 или 5.

=ИЛИ (A1={1;3;5})



Подробнее о работе с массивами рассказывается в части IV.

Очень часто использование массивов помогает упростить формулу. Следующая формула возвращает тот же результат. Но в качестве аргумента в ней используется не массив, а функции ЕСЛИ:

=ЕСЛИ (A1=1; ИСТИНА; ЕСЛИ (A1=3; ИСТИНА; ЕСЛИ (A1=5; ИСТИНА; ЛОЖЬ)))

Способы введения функции в формулу

Ввести функцию в формулу можно как вручную, так и с помощью диалогового окна Мастер функций.

Ручной ввод функций

Если вы уже знакомы с функцией — знаете, сколько аргументов должна иметь эта функция и к какому типу эти аргументы относятся, — вы можете просто ввести функцию и ее аргументы непосредственно в формулу. Зачастую этот метод оказывается наиболее эффективным.



В Excel 2002 появился полезный инструмент — во всплывающей подсказке отображается список имен аргументов (рис. 4.1). Если это окно мешает вам вводить данные, перетащите его в другую область рабочего листа.

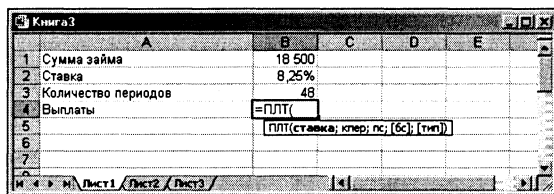


Рис. 4.1. Когда вы вводите функцию, в Excel 2002 появляется список ее аргументов

Если при вводе функции вы пропустите заключительную скобку, Excel поставит скобку автоматически. Например, если вы введете =СУММ(A1:C12 и нажмете <Enter>, программа исправит ошибку, вставив закрывающую скобку.



После того как вы ввели функцию, Excel всегда преобразует записанные символы в верхний регистр. Поэтому разумно записывать все функции символами нижнего регистра. Если после нажатия клавиши <Enter> программа не преобразует текст в верхний регистр, это означает, что Excel не распознал введенную вами запись как формулу. Так происходит в том случае, если при вводе функции была допущена ошибка или данная функция просто недоступна (например, такая функция определена только в надстройке, которая в данный момент не установлена в вашей системе).

Вставка функции с помощью диалогового окна Мастер функций

Диалоговое окно Мастер функций позволяет ввести функцию и ее аргументы полуавтоматически. Использование диалогового окна Мастер функций позволяет избежать ошибок в написании функций. Кроме того, функция, вставленная с помощью этого диалогового окна, будет иметь соответствующее число аргументов, расположенных в правильном порядке.



В версиях программы, предшествующих Excel 2002, это диалоговое окно носит имя Вставка функции.

Для того чтобы вставить функцию, выберите нужный пункт в списке диалогового окна Мастер функций (рис. 4.2). Вызвать диалоговое окно можно несколькими способами:

- ♦ Выберите команду Вставка⇒Функция.
- ♦ Щелкните на кнопке Мастер функций, которая находится рядом со строкой формул. В версиях программы, предшествующих Excel 2002, эта кнопка располагается на стандартной панели инструментов.
- ♦ Нажмите комбинацию клавиш <Shift+F3>.

Когда вы выбираете категорию в раскрывающемся меню, в окне списка отображаются только функции этой категории. В категории 10 недавно использовавшихся представлены функции, которые вы использовали в последнее время. В категории Полный алфавитный перечень представлены функции всех категорий. Если вы знаете имя функции, а ее категория вам не известна, откройте категорию Полный алфавитный перечень.



В Excel 2002 появилась возможность поиска функций. Для поиска используется текстовое поле, расположенное вверху диалогового окна Мастер функций. Введите в окно одно или несколько ключевых слов и щелкните на кнопке Найти. Excel выдаст список функций, удовлетворяющих условиям поиска.

Обратите внимание на то, что после выбора функции в поле списка Функция появляются название этой функции (и аргументов) и ее краткое описание.

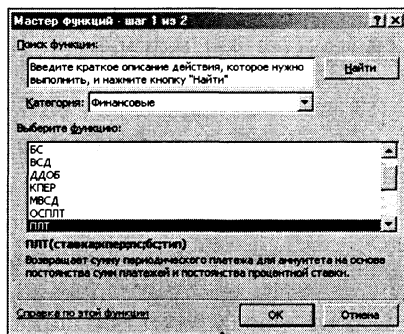


Рис. 4.2. Диалоговое окно Мастер функций

Выбрав нужную функцию, щелкните на кнопке ОК. Появится диалоговое окно Аргументы функции, показанное на рис. 4.3. В этом диалоговом окне задайте аргументы функции. Для того чтобы задать аргумент диапазона удобно нажать на кнопку свернутого окна (значок в правом углу поля каждого аргумента). Excel временно сворачивает диалоговое окно Аргументы функции, что позволяет выделить диапазон на рабочем листе.

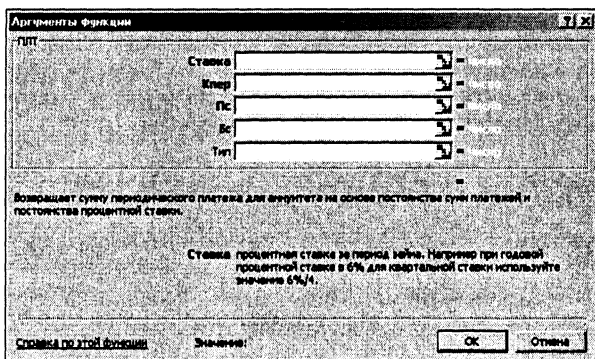


Рис. 4.3. Диалоговое окно Аргументы функции



Диалоговое окно Аргументы функции доступно только в Excel 2002. В предыдущих версиях функции роль этого диалогового окна выполняет Мастер функций (аналогичный диалоговому окну Аргументы функции по внешнему виду и назначению).

Несколько полезных советов по вводу функций

Приведенный ниже список содержит несколько полезных советов, которые помогут вам при вводе функций с помощью диалоговых окон Вставка функции и Мастер функций:

- ◆ Щелкните на ссылке Справка (или клавишу <F1>), выделив название функции, и вы получите информацию о выбранной вами функции (рис. 4.4).

Позвольте Excel вставлять функции за вас

В большинстве случаев вам придется самостоятельно вставлять функции. Однако в некоторых ситуациях Excel вставляет функции автоматически:

- ◆ Когда вы щелкаете на кнопке Автосумма, расположенной на стандартной панели инструментов, Excel производит быструю проверку выделенных и окружающих ячеек. Затем программа предлагает формулу, в которой используется функция СУММ. Если это именно то, что вам нужно, нажмите клавишу <Enter> (или повторно щелкните на кнопке Автосумма), чтобы подтвердить предложенную программой формулу. В Excel 2002 рядом с кнопкой Автосумма приведена стрелка меню, щелкнув на которой вы получите доступ к дополнительным функциям.
- ◆ Выбрав Данные⇒Итоги, вы активизируете диалоговое окно Промежуточные итоги, которое дает возможность задать несколько параметров. После того как вы зададите параметры, Excel вставит строки и автоматически введет формулы. В этих формулах будет использована функция ПРОМЕЖУТОЧНЫЕ.ИТОГИ.

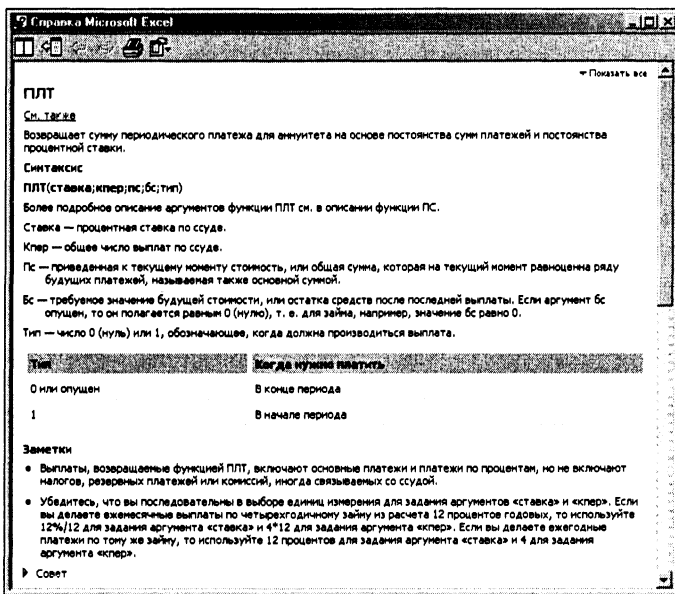


Рис. 4.4. Не забывайте о справочной системе Excel. Обратитесь за помощью к программе, и вы получите исчерпывающую информацию о функциях

- ◆ Если активная ячейка уже содержит формулу, в которой использована функция, щелчком на кнопке **Вставить функцию** можно вызвать диалоговое окно **Аргументы функции**.
- ◆ Диалоговое окно **Мастер функции** может использоваться и для вставки функции в уже существующую формулу. Для этого отредактируйте формулу и переместите курсор туда, куда вы хотите вставить функцию. Затем откройте диалоговое окно **Мастер функции** и выберите нужную функцию.
- ◆ Если вы передумали вставлять функцию, щелкните на кнопке **Отмена**.
- ◆ Количество аргументов той или иной функции определяет число текстовых полей диалогового окна **Аргументы функции**. Если функция не имеет ни одного аргумента, в диалоговом окне не будет ни одного текстового поля. Если функция может иметь различное число аргументов (как, например, функция **СРЗНАЧ**), Excel представляет новое текстовое поле каждый раз, когда вы вводите дополнительный аргумент.
- ◆ В левой части каждого текстового поля диалогового окна **Аргументы функции** указывается текущее значение каждого аргумента.
- ◆ В Excel есть функции, имеющие несколько форм (например, **ИНДЕКС**). Если вы выберете одну из этих функций, Excel откроет диалоговое окно, позволяющее выбрать ту форму функции, которую вы хотите использовать.
- ◆ Если вы не помните, какие аргументы имеет та или иная функция, введите знак равенства и название функции, а затем нажмите комбинацию клавиш <Ctrl+Shift+A>. Excel вставит функцию с “заполнителями”, содержащими описание аргументов этой функции (рис. 4.5). Эти “заполнители” заменяются конкретными аргументами функции.
- ◆ Для того чтобы ускорить поиск нужной функции в списке **Функции** диалогового окна **Вставка функции** (**Мастер функций**), откройте список, введите первую букву имени

функции, а затем прокрутите список до нужного вам пункта. Например, если вы хотите найти функцию SIN в категории Полный алфавитный перечень, щелкните кнопкой мыши в любом месте списка Функции и нажмите S. Excel выберет первую функцию, имя которой начинается с буквы S. Нажимайте S, пока не найдете в списке функцию SIN.

- ◆ Если активная ячейка содержит формулу, в которой использована одна или несколько функций, вы можете отредактировать эти функции с помощью диалогового окна Аргументы функции. Щелкните на функции кнопкой мыши для того, чтобы эта функция появилась в строке формул. Затем щелкните на кнопке Вставить функцию.

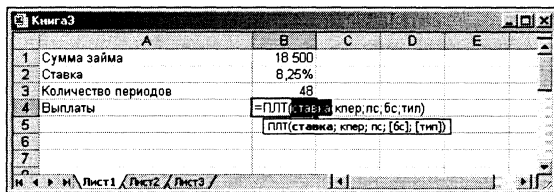


Рис. 4.5. Нажмите <Ctrl+Shift+A>, и Excel вставит функцию с “заполнителями”, содержащими описание аргументов этой функции

Категории функций

Далее я привожу список категорий функций с кратким описанием каждой из них.



В последующих главах приведены некоторые примеры специфического применения функций.

Финансовые функции

Финансовые функции позволяют производить экономические вычисления, связанные с расчетом денежных средств. Функция ПРОЦПЛАТ, например, используется для вычисления величины ежемесячной выплаты по ссуде на покупку автомобиля. Для вычисления необходимо задать следующие аргументы: размер ссуды, процентную ставку по ссуде и срок, на который ссуда предоставлена.

Функции даты и времени

Функции этой категории позволяют анализировать и работать со значениями даты и времени в формулах. Например, функция СЕГОДНЯ возвращает текущую дату (которая указана на системных часах).

Математические функции

В эту категорию входят разнообразные функции, выполняющие математические и тригонометрические вычисления.



Во всех тригонометрических функциях углы измеряются в радианах (а не в градусах). Для того чтобы преобразовать градусы в радианы, используйте функцию РАДИАНЫ.

Статистические функции

Функции этой категории предназначены для проведения статистического анализа диапазонов значений. С помощью статистических функций можно вычислить наименьшее значение набора значений, значение моды множества данных, стандартное отклонение по выборке или дисперсию.



Некоторые функции этой категории требуют установки надстройки “Пакет анализа”.

Функции ссылок и массивов

Функции этой категории используются для нахождения значений в списках или таблицах. Самый распространенный пример — поиск значения в таблице налогов. Функция ПОИСКПОЗ может использоваться для определения налоговой ставки для конкретного уровня дохода.

Функции работы с базами данных

Функции этой категории применяются для вычисления суммы данных списка (также известного как база данных рабочего листа), который удовлетворяет определенным условиям. Предположим, у вас есть список, содержащий информацию о месячном объеме продаж. Функцию БСЧЕТ можно использовать для подсчета записей об объеме продаж в северном регионе, значение которых превышает 10 000.

Текстовые функции

Текстовые функции позволяют манипулировать текстовыми строками в формулах. Функция ПСТР, например, возвращает заданное число знаков из строки текста, начиная с указанной позиции. Другие функции позволяют изменить регистр строки текста (например, преобразовать символы нижнего регистра в символы верхнего регистра).

Логические функции

Эта категория состоит всего из шести функций. Эти функции позволяют проверить условие (для логических ИСТИНА или ЛОЖЬ). Функция ЕСЛИ также может быть полезна, поскольку она предоставляет формулам способность к принятию решений.

Информационные функции

Функции этой категории помогают определить тип данных, хранящихся в ячейке. Например, функция ЕТЕКСТ возвращает ИСТИНА, если значение является текстом. С помощью функции ЕПУСТО можно определить, пуста ли ячейка. Функция ЯЧЕЙКА возвращает полезную информацию о конкретной ячейке.

Инженерные функции

Функции этой категории помогут вам при решении инженерных прикладных задач. Эти функции позволяют работать с комплексными числами, а также выполнять преобразования единиц одной системы исчисления или измерения в единицы исчисления или измерения другой системы.



Для того чтобы использовать функции инженерной категории, необходимо установить дополнение “пакет анализа”.

Функции, созданные пользователем

К этой категории относятся пользовательские функции рабочего листа, созданные с помощью VBA. Эти функции выполняются так же, как и встроенные функции Excel. Отличие состоит в том, что описания аргументов этих функций не появляются в диалоговых окнах Вставка функции и Мастер формул.

Другие категории функций

Кроме описанных выше категорий, в Excel существует еще четыре категории функций, которые не появляются в диалоговом окне Вставка функции. Это такие категории, как Команды, Настраиваемые функции, Управление макросами и DDE/Внешние данные. Эти категории — пережиток устаревших версий Excel. Создав пользовательскую функцию, вы можете определить ее в одну из этих категорий. Кроме того, вы можете обнаружить и другие категории, созданные макросами.

Непостоянные функции

Некоторые функции Excel принадлежат к особому классу так называемых *непостоянных функций*. Программа автоматически пересчитывает изменяемую функцию при пересчете рабочей книги даже в том случае, если формула, в которой эта функция используется, не задействована в пересчете.

Одной из непостоянных функций является функция СЛЧИС, которая возвращает новое случайное число при каждом пересчете рабочего листа. Кроме того, в Excel присутствуют следующие изменяемые функции:

ОБЛАСТИ
ЯЧЕЙКА
СТОЛБЕЦ
ИНДЕКС
ДВССЫЛ
СЕЙЧАС
СМЕЩ
СТРОКА
СЕГОДНЯ

Использование непостоянных функций имеет один побочный эффект: Excel будет всегда напоминать вам о том, что перед тем, как закрыть рабочую книгу, ее необходимо сохранить (даже в том случае, если вы не вносили никаких изменений). Например, вы открыли рабочую книгу, в которой используются непостоянные функции, просмотрели (ничего при этом не меняя) и хотите закрыть файл. Excel обязательно спросит, хотите ли вы сохранить рабочую книгу.

Вы можете “обойти” это поведение, используя режим Вычисления вручную, при этом опцию пересчет перед сохранением необходимо отключить.



Информацию о том, как определить пользовательские функции в одну из категорий, вы найдете в главе 23.

Функции настройки “Пакет анализа”

Разобравшись со встроенными функциями Excel, можете приступить к освоению дополнительных функций, предоставляемых надстройкой “Пакет анализа”. В этой надстройке вы найдете десятки функций рабочего листа.

Загрузив эту надстройку, вы заметите, что в диалоговом окне Мастер функции появилась новая категория — Инженерные функции. В такие категории, как Финансовые, Дата и время, Математические и Информационные, добавились новые функции.

Резюме

Эта глава служит введением к теме “Функции рабочего листа”. Excel предоставляет сотни функций, которые можно применять в формулах. Кроме того, вы можете использовать функции, доступные благодаря надстройкам. В последующих главах этой книги приведено множество примеров использования функций в формулах. В следующей главе будут рассмотрены функции категории Текстовые.

Работа с текстом

В этой главе...

- ◆ Несколько слов о самом тексте
- ◆ Текстовые функции
- ◆ Дополнительные текстовые формулы
- ◆ Создание собственных текстовых функций с помощью языка VBA
- ◆ Резюме

Приложение Excel, прежде всего, известно благодаря способности выполнения любых операции с числами. Тем не менее, его способности в области обработки текста также весьма разнообразны. Как известно, Excel позволяет вводить текстовые данные для создания таких элементов как, например, заголовки строки или столбца, имя пользователя и адрес, часть числа и многое другое. И, как и следовало ожидать, для того чтобы управлять текстом в ячейках, Excel предоставляет широкий набор специальных формул.

В этой главе приведено множество примеров формул, которые используют функции управления текстом. Многие из них способны творить настоящие чудеса, которые, иногда, трудно себе представить.

Несколько слов о самом тексте

Работа Excel начинается сразу же после того, как пользователь вводит данные в ячейку. Приложение определяет, что же именно вводится: формула, числовая информация (то же самое относится к дате и времени) или что-либо еще. Говоря “что-либо еще”, мы как раз и подразумеваем текст.



Возможно, читателю уже приходилось сталкиваться с термином *строка*, используемым в понимании *текстовых данных*. Эти понятия часто взаимозаменяются. Иногда они даже могут встречаться вместе как термин *текстовая строка*.

Количество символов в ячейке

В более ранних версиях Excel 5.0 и Excel 95 в одну ячейку можно было ввести до 255 символов. Но начиная с Excel 97, разработчики программных продуктов компании Microsoft значительно увеличили этот показатель. В ячейку приложения Excel версии 97 можно ввести уже 32 000 символов. В дальнейшем, в этой главе, мы будем вводить в ячейку до 30 000 символов. Безусловно, мы ни в коем случае не рекомендуем использовать процессор электронных таблиц вместо текстового. Но если у пользователя установлено приложение не старше Excel 97, то не стоит излишне беспокоиться о величине текста, вводимого в ячейку.



Несмотря на то, что ограничение вводимых в ячейку символов устанавливается в пределах 30 000, в действительности, количество символов, отображаемых на экране, ограничено. Более того, как будет показано несколько позже, многие функции могут работать некорректно в случае, если текстовая строка длиннее 255 символов.

Числа как текст

Как мы уже говорили, Excel различает текстовый и числовой форматы. И, если пользователь хочет, чтобы числовая информация рассматривалась как текст, необходимо выполнить следующее:

- ◆ Примените к ячейке формат **Текстовый**. Для этого выберите команду меню **Формат**⇒**Ячейки**, далее щелкните на вкладке **Число**, а затем в поле со списком выберите категорию **Текстовый**. Если к ячейке не применялось никакое специальное горизонтальное выравнивание, значения будут выровнены по левому краю ячейки (так же как и в случае обычного текста).
- ◆ Введите перед числом знак апострофа. Знак апострофа, в данном случае, не будет отображаться, но ячейка будет выглядеть так, как если бы в нее был введен текст.

Установив для ячейки текстовый формат (или используя апостроф), с нею все еще можно осуществлять *некоторые* математические операции. Для этого *выражения* в ячейке должны вводиться в виде числа. К примеру, предположим, что ячейка A1 содержит значение, которому предшествует знак апострофа. Ниже приведена формула, которая отображает значение ячейки A1, увеличенное на 1:

=A1+1

Следующая формула будет обрабатывать содержимое ячейки A1 как 0:

=СУММ(A1:A10)

Если пользователь знаком с Lotus 1-2-3, он вероятно заметит это существенное отличие. Lotus 1-2-3 никогда не воспринимает текст как значение. В некоторых случаях обработка текста как числа может быть весьма полезной. В других, это вызывает определенные проблемы. Каков же итог? Помните о несостоятельности Excel в части того, как он обрабатывает число, форматированное как текст.



Имейте в виду, что Excel 2002 помечает числа, которым предшествует апостроф, смарт-тэгом. Смарт-тэг можно использовать для того, чтобы конвертировать "текст" в реальное числовое значение.

Когда число совсем не число?

Импортируя данные в приложение Excel, пользователь может столкнуться с одной общей проблемой: иногда, импортируемые значения быстро преобразуются в фактические значения. Активизируйте любую пустую ячейку и введите в нее значение 1. Выберите команду **Правка**⇒**Копировать**, чтобы скопировать это значение в буфер обмена. Затем выберите диапазон, содержащий значения, которые необходимо преобразовать. Выберите команду **Правка**⇒**Специальная вставка**. В диалоговом окне **Специальная вставка** выберите переключатель **умножить**, а затем щелкните на кнопке **ОК**. Прделанная процедура заставит Excel обрабатывать нечисловую информацию как фактические значения.

Текстовые функции

Приложение Excel имеет довольно широкий ассортимент функций, с помощью которых обрабатывается текст. Большинство из этих функций пользователь может найти в диалоговом окне Мастер функций, выбрав опцию Текстовые в раскрывающемся списке Категория. Такое расположение функций весьма удобно для пользователя. Некоторые другие функции, которые тоже могут использоваться при работе с текстом, встречаются в других категориях. Например, функция ЕТЕКСТ находится в категории Проверка свойств и значений в том же диалоговом окне.



Обратите внимание, в приложении Б приведен перечень функций, относящихся к категории текстовых. Просмотреть эти функции можно также в диалоговом окне Мастер функций, вызвав его с помощью команды Вставка⇒Функция.

Область применения большинства текстовых функций не ограничивается только текстом. Иначе говоря, эти функции могут использоваться и в ячейках, содержащих числовые значения. В отличие от других электронных таблиц (как, например, Lotus 1-2-3), Excel предоставляет прекрасную возможность обрабатывать числа как текст и, наоборот, текст как числа.

В этой главе приводятся примеры некоторых общих, широко используемых операций, которые пользователь может выполнять с текстом. Возможно, читатель захочет взять себе на вооружение некоторые из приведенных ниже примеров.

Определение наличия в ячейке текста

В некоторых случаях у пользователя возникает необходимость определить тип данных, содержащихся в отдельной ячейке, для чего требуется соответствующая формула, позволяющая сделать это. К примеру, можно использовать функцию ЕСЛИ, чтобы вставить в ячейку результат только в том случае, если он является текстом. Для того чтобы определить, содержит ли ячейка определенный текст в приложении Excel имеется три функции:

1. ЕТЕКСТ
2. ЯЧЕЙКА
3. ТИП

Однако, как мы увидим далее, эти функции не всегда достаточно надежны:

Функция ЕТЕКСТ

Функция ЕТЕКСТ использует один аргумент и возвращает значение ИСТИНА в том случае, если ячейка содержит текст, в противном случае, функция возвращает значение ЛОЖЬ. Обратите внимание, если ячейка A1 содержит текст, то следующая формула возвращает значение ИСТИНА:

=ЕТЕКСТ (A1)



Безусловно, даже несмотря на всю свою полезность, функция ЕТЕКСТ несовершенна. На самом деле, в отдельных случаях, результат ее действия может быть совсем корректным. Несмотря на то, что Excel 97 и более поздние версии приложения могут сохранять в ячейке огромное количество символов (до 32 000), функция ЕТЕКСТ не учитывает этот факт и возвращает значение ЛОЖЬ в том случае, если ее аргумент обращается к ячейке, которая содержит более, чем 255 символов. В Excel 2000 эта проблема исправлена и функция ЕТЕКСТ работает так, как и полагается, независимо от количества введенных символов в ячейке.

Функция ТИП

Функция ТИП также использует один аргумент и возвращает значение, которое указывает тип данных, содержащихся в ячейке. Например, если ячейка А1 содержит текстовую информацию, формула, приведенная ниже, возвратит значение 2 (кодový номер текстового формата):

=ТИП (А1)



Имейте в виду, что функция ТИП перестает работать, когда в ячейке содержится более, чем 255 символов. В этом случае, функция возвращает значение 16 — кодový номер значения Ошибка.

Функция ЯЧЕЙКА

Теоретически, функция ЯЧЕЙКА помогает пользователю определить, применяется ли к данной ячейке текстовый формат или используется ли в ней знак апострофа. Первый аргумент функции ЯЧЕЙКА может быть любым из 12 ключевых слов, включая *формат*, *префикс* или *тип*.

Имейте в виду, что в том случае, если к ячейке применен текстовый формат, то при выборе любого из этих значений функция не работает. Например, если в ячейку А1 ввести число, а затем применить к ней формат Text, то приведенная ниже формула возвратит значение 6. В данном случае приложение Excel будет полагать, что для форматирования ячейки был использован формат Общий:

=ЯЧЕЙКА ("формат"; А1)

При использовании ключевого слова *префикс* в качестве первого аргумента ЯЧЕЙКА, функция возвращает знак апострофа, если значению, содержащемуся в ячейке, предшествует апостроф. Однако, если к ячейке, содержащей числовую информацию, применен формат Текстовый, функция не возвратит ничего. То же самое происходит и при использовании в качестве первого аргумента функции ЯЧЕЙКА ключевого слова *тип*. В этом случае функция также будет работать некорректно. Например, если ячейка содержит более, чем 255 символов, функция возвратит значение v.

Работа с кодами символов

Каждому символу, который пользователь видит на экране своего компьютера, соответствует определенное кодовое число. Для работы в системе Windows приложение Excel использует стандартный набор символов ANSI, который содержит 255 символов, пронумерованных числами в диапазоне от 1 до 255.

Обратите внимание, на рис. 5.1 отображена часть рабочего листа приложения Excel, на котором показаны все 255 символов. В этом примере использовался шрифт Arial (другие шрифты могут иметь символы, отличные от приведенных на рисунке).

При работе с кодами символов Excel задействует две специальные функции: КОДСИМВ и СИМВОЛ. Несмотря на то, что эти функции не столь популярны, как остальные, они весьма уместны при совместном использовании с другими функциями, которые мы обсудим несколько позже, в следующих разделах.



Имейте в виду, что функции КОДСИМВ и СИМВОЛ работают только со строками ANSI. Эти функции не будут выполняться для двухбайтовых строк Unicode.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
3	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
4	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
5	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
6	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
7	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
8	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
9	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
10	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
11	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176
12	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
13	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208
14	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
15	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
16	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256
17	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272
18	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288
19	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304
20	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320
21	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336
22	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352
23	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368
24	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384
25	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
26	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416
27	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432
28	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448
29	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464
30	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480

Рис. 5.1. Набор символов ANSI для шрифта Arial

Функция КОДСИМВ

Функция КОДСИМВ, используемая в приложении Excel, возвращает код символа, введенного в качестве аргумента функции. Например, формула, приведенная ниже, возвращает значение 65 — код для символа А, введенного в верхнем регистре:

=КОДСИМВ ("А")

В том случае, если аргумент функции КОДСИМВ содержит более одного символа, функция использует только первый символ. Поэтому следующая формула возвращает то же самое значение — 65:

=КОДСИМВ ("Abbey Road")

Функция СИМВОЛ

По своей сути функция СИМВОЛ полностью противоположна функции КОДСИМВ. Аргументом функции является числовое значение в интервале от 1 до 255, а используемая функция возвращает символ, соответствующий этому значению. Например, приведенная ниже формула возвращает символ А:

=СИМВОЛ (65)

Чтобы продемонстрировать характер противоположности функций КОДСИМВ и СИМВОЛ, давайте введем в ячейку следующую формулу:

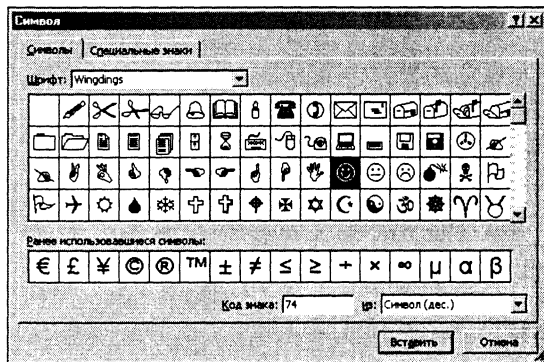
=СИМВОЛ (КОДСИМВ ("А"))

Эта формула возвращает символ А. Этот пример призван скорее проиллюстрировать действие функций, вряд ли он будет полезен на практике. Сначала введенный символ конвертируется в соответствующее значение кода (65), после чего функция СИМВОЛ возвращает символ А, соответствующий данному значению.

Специальные символы?

В Excel 2002 пользователю предоставляется прекрасная возможность использовать диалоговое окно Символ, которое можно вызвать, выбрав команду Вставка⇒Символ. Это диалоговое окно позволяет вставить в ячейку любые специальные символы (включая символы Unicode). Например, пользователь может вставить в электронную таблицу улыбающуюся рожицу (если вдруг у него возникает такая необходимость). Для этого в Excel достаточно отобразить диалоговое окно Символ и выбрать шрифт Wingdings (см. рисунок ниже). Затем просмотрите символы, найдите изображение рожицы и щелкните на кнопке Вставить. Обратите внимание, что этому символу соответствует код 74.

Если в компьютере пользователя установлена более ранняя версия приложения Excel, подобные функциональные возможности можно получить, используя программу Windows Character Map (charmap.exe).



Предположим, что ячейка A1 содержит символ А (верхний регистр). Приведенная ниже формула возвращает символ а (нижний регистр):

=СИМВОЛ (КОДСИМВ (A1) +32)

Эта формула использует определенные преимущества, заключающиеся в фактическом упорядочении символов. Дело в том, что все символы латинского алфавита в пределах набора расположены в алфавитном порядке. Символы нижнего регистра следуют за прописными буквами (через несколько других символов, добавленных между ними). Каждому символу нижнего регистра соответствует код точно на 32 больше, чем код соответствующей прописной буквы.

Равенство строк?

Чтобы определить, содержат ли две ячейки похожие записи, можно использовать простую логическую формулу. Например, ниже приведена формула, с помощью которой можно определить, содержит ли ячейка A1 то же самое значение, что и ячейка A2:

=A1=A2

Однако имейте в виду, что Excel может допускать некоторую неточность при сравнении содержимого ячеек, если в ячейки включен текст. Рассмотрим, например, ситуацию, когда ячейка A1 содержит слово *Январь* (первая буква прописная), а ячейка A2 — слово *ЯНВАРЬ* (все буквы верхнего регистра). Приведенная выше формула возвращает значение ИСТИНА, несмотря на то, что эти две ячейки содержат не одинаковое текстовое значение. Говоря иными словами, сравнение, в данном случае, не учитывает регистр символов.

Чаще всего пользователь не испытывает особого волнения по поводу регистра текста. Если же необходимо сделать точное сравнение, учитывая регистр символа, можно использовать

функцию СОВПАД. Приведенная ниже формула возвращает значение ИСТИНА только в том случае, если ячейки A1 и A2 содержат *точно* идентичные записи:

=СОВПАД (A1 ; A2)

Следующая формула возвращает значение ЛОЖЬ, поскольку первая строка содержит в конце пробел:

=СОВПАД ("ноль " ; "ноль ")

Объединение двух и более ячеек

Excel использует знак & как оператор конкатенации (объединения). *Конкатенация* — это просто модный термин, который описывает, что происходит после того, как пользователь объединяет содержимое двух или более ячеек. Например, если ячейка A1 содержит текст *San Diego*, а ячейка A2 — *California*, приведенная ниже формула возвратит текст *San DiegoCalifornia*:

=A1&A2

Обратите внимание, что эти две строки объединены без промежуточного пробела. Для того чтобы добавить пробел между этими двумя строками и получить текст *San Diego California*, необходимо использовать следующую формулу:

=A1&" "&A2

Другой способ, который может быть даже лучше предыдущего, — использовать запятую и пробел для того, чтобы получить текст *San Diego, California*:

=A1&" , "&A2

Еще одна операция, предназначенная для устранения символов кавычек, заключается в использовании функции СИМВОЛ с соответствующим аргументом. Обратите внимание, в приведенном ниже примере функции СИМВОЛ в объединяемый текст вставляются запятая (44) и пробел (32):

=A1&СИМВОЛ (44) &СИМВОЛ (32) &A2

Если пользователь хочет выполнить “перенос” текста, то при объединении текста в строке необходимо использовать функцию СИМВОЛ (10), после чего проверить, применен ли к ячейке текстовый формат с переносом слов. Пример объединения текста в ячейке A1 и текста в ячейке B2, с символом переноса между ними, продемонстрирован в следующей формуле:

=A1&СИМВОЛ (10) &B1

Ниже приведен еще один пример использования функции СИМВОЛ. Следующая формула возвращает строку *Stop*, объединяя четыре символа, полученные с помощью функции СИМВОЛ:

=СИМВОЛ (83) &СИМВОЛ (116) &СИМВОЛ (111) &СИМВОЛ (112)

Рассмотрим еще один последний пример использования оператора &. В данном случае формула объединяет текст с результатом выражения, которое возвращает максимальное значение столбца C:

= "Максимальное значение в столбце C " &МАКС (C:C)

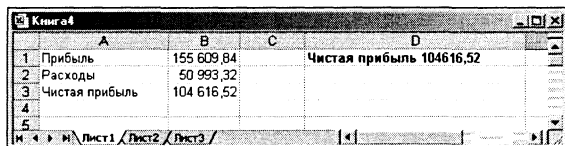


Имейте в виду, что приложение Excel использует также функцию СЦЕПИТЬ, которая поддерживает до 30 аргументов. Эта функция просто объединяет аргументы в единую строку. Безусловно, многим пользователям нравится эта формула, однако использование знака & приводит к более короткой записи.

Отображение значений, отформатированных в текстовом формате

Функция ТЕКСТ в приложении Excel позволяет отобразить значение ячейки в определенном числовом формате. На первый взгляд может показаться, что эта функция имеет весьма сомнительную пригодность. Но она помогает осуществить некоторые полезные действия, предназначенные для определенных целей, например таких, что продемонстрированы в этом разделе. Обратите внимание, на рис. 5.2 отображен обычный рабочий лист Excel. Ниже приведена формула, используемая в этом рабочем листе в ячейке D1:

= "Чистая прибыль " & V3



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D
1	Прибыль	155 609,84		Чистая прибыль 104616,52
2	Расходы	50 993,32		
3	Чистая прибыль	104 616,52		
4				
5				

Рис. 5.2. Формула, содержащаяся в ячейке D1, отображает уже не просто форматированное число

По существу, эта формула объединяет текстовую строку с содержимым ячейки V3 и отображает полученный результат. Обратите внимание, что к ячейке V3 не был применен ни один специальный формат. Тем не менее, при желании пользователь может установить для этой ячейки денежный формат с использованием символа валюты.



Имейте в виду, что, вопреки всем ожидаемым результатам, при применении к ячейке, содержащей формулу числового формата, это не даст никакого эффекта. Все дело в том, что используемая формула возвращает строку, а не числовое значение.

Обратите внимание, как была изменена формула, использующая функцию ТЕКСТ для того, чтобы применить форматирование к значению, содержащемуся в ячейке V3:

= "Чистая прибыль " & ТЕКСТ (V3 ; "\$#, ##0,00 ")

Эта формула будет отображать и текст, и само отформатированное числовое значение: *Чистая прибыль \$ 104 616,52.*

Второй аргумент функции ТЕКСТ содержит стандартную строку числового формата, используемого в приложении Excel. В качестве этого аргумента пользователь может ввести любую допустимую строку числового формата.

В предыдущем примере мы использовали простую ссылку на рассчитываемую ячейку V3. Но это не единственная возможность. Вместо ячейки можно использовать определенное выражение. Ниже приведен пример, в котором текст объединяется с числом, полученным путем следующего вычисления:

= "Средний расход: " & ТЕКСТ (СРЗНАЧ (A:A) ; "\$#, ##0,00 ")

Эта формула возвращает строку: *Средний расход: \$ 7 794,57.* Давайте рассмотрим другой пример, в котором используется функция СЕГОДНЯ, возвращающая текущую дату и время. Использование этой функции совместно с функцией ТЕКСТ позволяет отобразить на экране текущую дату и время, представленную в удобном для восприятия формате:

= "Отчет напечатан " & ТЕКСТ (СЕГОДНЯ () ; "д ММММ, гггг в ч:мм AM/PM ")



Для получения более подробной информации о числовых форматах приложения Excel, обратитесь к приложению В.

Отображение денежных значений, отформатированных как текст

В приложении Excel существует функция РУБЛЬ, с помощью которой содержащееся в ячейке число преобразуется в текст, представленный в денежном формате. Эта функция использует два аргумента: число, которое необходимо преобразовать, и количество знаков, отображаемых на экране после запятой. Как правило, функция РУБЛЬ использует обозначения денежной единицы, принятых в данном регионе (например, \$ в США).

В отдельных случаях функция РУБЛЬ может использоваться вместо функции ТЕКСТ. Однако намного эффективней использовать все-таки функцию ТЕКСТ, которая является более гибкой, поскольку не ограничивает пользователя определенным числовым форматом.

Приведенная ниже формула возвращает текст: *Итого:1 287,37р.* Второй аргумент функции РУБЛЬ определяет число десятичных знаков после запятой.

```
=Итого: "&РУБЛЬ(1287,367;2)
```

Повторяющиеся символы или строки

Функция ПОВТОР предназначена для того, чтобы повторить любую строку текста или символ (первый аргумент) заданное пользователем количество раз (второй аргумент). Например, следующая формула возвращает текст *НоНоНо*:

```
=ПОВТОР("Но";3)
```

Эту функцию очень удобно использовать для создания горизонтального разделителя между ячейками. Например, приведенная ниже формула создает строку, состоящую из 20 волнистых линий (тильда), расположенных по длине строки:

```
=ПОВТОР("~";20)
```

Создание текстовой гистограммы

Одно из наиболее полезных направлений использования функции ПОВТОР — это создание простой гистограммы непосредственно на рабочем листе. В этом случае необходимость использования диаграммы полностью отпадает. Обратите внимание, на рис. 5.3 приведен пример такой гистограммы. Такой вид графического отображения данных может быть особенно полезным, когда пользователю необходимо визуально отобразить сумму множества значений, поскольку стандартная диаграмма Excel, для подобных случаев, может быть очень громоздкой.

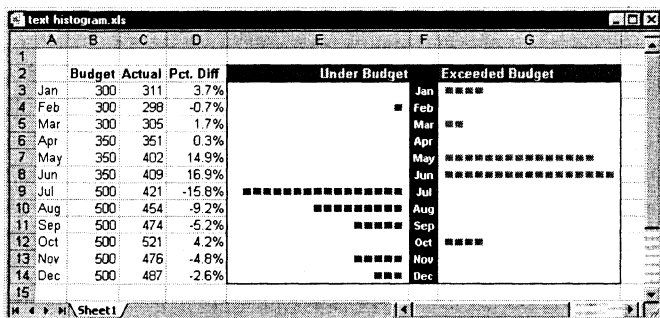


Рис. 5.3. Чтобы быстро создать гистограмму в пределах пространства рабочего листа, воспользуйтесь функцией ПОВТОР

Формулы, содержащиеся в ячейках столбцов E и G, графически отображают ежемесячное изменение бюджета, выводя на экран целый ряд символов шрифта Wingdings. В этом примере мы использовали символ n, который в этом шрифте отображается как маленький квадрат. Формула, использующая функцию ПОВТОР, определяет количество символов, которое должно быть отображено на экране. Формулы для ячеек, формирующих диаграмму, будут следующими:

E3: =ЕСЛИ(D3<0;ПОВТОР("n";-ОКРУГЛ(D3*100;0));" ")

F3: =A3

G3: =ЕСЛИ(D3>0;ПОВТОР("n";ОКРУГЛ(D3*100;0));" ")

Установите для ячеек E3 и G3 шрифт Wingdings, а затем скопируйте формулы, содержащиеся в этих ячейках, вниз в ячейки столбцов, чтобы распределить данные. Выровняйте текст в столбце E по правому краю, после чего примените любое другое форматирование ячеек. Имейте в виду, что в зависимости от величины числового диапазона данных может возникнуть необходимость изменить масштаб значений. Поэкспериментируйте с перемещением 100 значений в формулах. Выбранный ранее для диаграммы символ n можно заменить на любой другой.

Добавление к числу определенных символов

Вероятно, многие пользователи уже не раз сталкивались с таким распространенным (особенно при печатании чеков) методом защиты, как дополнение числовых значений справа звездочками. Приведенная ниже формула, наряду со значением, содержащимся в ячейке A1, отображает знаки звездочек, дополняя общее количество символов до 24:

=(A1&ПОВТОР(" * ";24-ДЛСТР(A1)))

Используя следующую формулу, пользователь может добавить звездочки к числу слева:

=ПОВТОР(" * ";24-ДЛСТР(A1)) &A1

Следующая формула отображает звездочки, дополняющие число сразу с двух сторон. Эта формула возвращает 24 символа в том случае, если число в ячейке A1 содержит четное количество символов, и 23 символа, если число содержит нечетное количество символов.

=ПОВТОР(" * ";12-ДЛСТР(A1)/2)&A1&ПОВТОР(" * ";12-ДЛСТР(A1)/2)

Тем не менее, приведенные выше формулы не столь совершенны, поскольку не могут отображать любое форматированное число. Обратите внимание, ниже приведена усовершенствованная версия формулы, которая отображает значение, содержащееся в ячейке A1 (форматированной ячейке), также, как и знак звездочки справа от этого значения:

=(ТЕКСТ(A1;"\$#,##0,00")&ПОВТОР(" * ";24-ДЛСТР(ТЕКСТ(A1;"\$#,##0,00"))))

На рис. 5.4 показано действие приведенной выше формулы.

	A	B	C
1	143,55	\$143,550,00	
2	9,9754	\$9,975,40	
3	1933,43	\$1933,430,00	
4	-908,32	-\$908,320,00	
5			
6			

Рис. 5.4. Так будет выглядеть число, дополненное звездочками при использовании приведенной выше формулы

В отдельных случаях, если возникает такая необходимость, пользователь может использовать свой собственный числовой формат. Для того чтобы дополнить ширину столбца, достаточно просто включить в собственный формат звездочку (*). Например, используя следующий вид числового формата, можно дополнить число знаками тире:

\$#, ##0.00*-

Чтобы просто дополнить число звездочками, используйте в формате две звездочки, так как это показано ниже:

\$#, ##0.00**



Для получения дополнительной информации о создании форматов пользователя обратитесь к приложению В. Здесь вы также найдете дополнительные примеры использования в отдельных форматах символа звездочки.

Удаление лишних пробелов и непечатаемых символов

Весьма часто данные, импортируемые в рабочую книгу приложения Excel, содержат излишнее количество пробелов, а также незнакомые символы, которые к тому же могут не выводиться на печать. На этот случай в арсенале Excel имеются две функции, которые помогают пользователю подогнать данные под определенную форму. Это функции СЖПРОБЕЛЫ и ПЕЧСИМВ.

1. Функция СЖПРОБЕЛЫ удаляет все начальные и конечные пробелы, а также заменяет несколько пробелов, расположенных рядом внутри текста, на один.
2. Функция ПЕЧСИМВ удаляет из строки все непечатаемые символы. Такие символы — это “ненужные данные”, они часто появляются при импортировании в Excel отдельных типов информации. Интересно, что из 255 кодовых символов ANSI, 39 — это непечатаемые символы. К ним относятся символы с кодами 1-31, 128-129, 141-144 и 157-158.

В приведенном ниже примере используется функция СЖПРОБЕЛЫ. Формула возвращает значение *Прибыль четвертого квартала* без излишних пробелов:

=СЖПРОБЕЛЫ(" Прибыль четвертого квартала ")

Подсчет символов в строке

Следующая функция Excel, подлежащая рассмотрению, — ДЛСТР, использует только один аргумент и возвращает количество символов, содержащихся в ячейке. Например, возьмем ячейку A1, содержащую строку *Продажи в сентябре*. Формула, приведенная ниже, возвратит значение 18:

=ДЛСТР (A1)

Обратите внимание, что пробелы также считаются символами и включаются в результирующее значение. Следующая формула возвращает общее количество символов в диапазоне ячеек A1 : A3:

=СУММ (ДЛСТР (A1) ; ДЛСТР (A2) ; ДЛСТР (A3))



Позже в этой главе мы еще рассмотрим несколько примеров, в которых показано, как подсчитывается число определенных символов в пределах одной строки. Кроме того, информацию по этому вопросу можно найти в главе 7, в которой обсуждаются методы вычисления значений в Excel и в главе 15, где речь пойдет о необыкновенных возможностях формул массивов.

Изменение регистра текста

В приложении Excel имеются три весьма удобные функции, которые позволяют изменить регистр текста:

1. Функция ПРОПИСН. Эта функция преобразует все символы текста в верхний регистр.
2. Функция СТРОЧН преобразует весь текст в нижний регистр.
3. Функция ПРОПНАЧ устанавливает для каждого слова текста “соответствующий” регистр (Первый Символ В Каждом Слове Печатается Прописными Буквами).

Эти функции действуют очень просто. Например, следующая формула все символы текста, содержащегося в ячейке A1, приводит к соответствующему регистру. Если ячейка A1 содержит текст *MR. JOHN Q. PUBLIC*, формула возвратит строку *Mr. John Q. Public*.

=ПРОПНАЧ (A1)

Надо сказать, что эти функции применяются только для символов, вошедших в алфавит. Все остальные символы они просто игнорируют и возвращают неизменными.

Извлечение из строки определенных символов

Многие пользователи Excel достаточно часто сталкиваются с необходимостью извлечь из строки некоторые символы. Например, пользователь может иметь список имен служащих организации (имена и фамилии), из которого он хочет выбрать фамилию каждого из них, то есть выбрать отдельные символы из каждой ячейки. Excel предоставляет несколько превосходных функций, позволяющих осуществить эту процедуру:

1. Функция ЛЕВСИМВ — возвращает заданное количество символов от начала строки.
2. Функция ПРАВСИМВ — возвращает заданное количество символов от конца строки.
3. Функция ПСТР — возвращает заданное количество символов, начиная с любой позиции в пределах строки.

Формула, приведенная ниже, возвращает последние 10 символов, содержимого ячейки A1. Если же ячейка A1 содержит менее, чем 10 символов, формула возвратит весь текст ячейки полностью.

=ПРАВСИМВ (A1 ; 10)

В следующей формуле используется функция ПСТР. Эта формула возвращает из ячейки A1 пять символов, начиная со второго. Другими словами, формула возвращает символы со второго по шестой.

=ПСТР (A1 ; 2 ; 5)

В следующем примере формула возвращает из текста, содержащегося в ячейке A1, только первый символ в верхнем регистре. Здесь используется функция ЛЕВСИМВ, которая позволяет извлечь первый символ и преобразовать его в верхний регистр. После чего набор символов присоединяется к другой строке, которая создается с помощью функции ПРАВСИМВ, извлекая из заданного текста все, кроме первого символа и преобразуя извлеченные символы в нижний регистр.

=ПРОПИСН (ЛЕВСИМВ (A1)) &ПРАВСИМВ (СТРОЧН (A1)) ;ДЛСТР (A1) -1)

Таким образом, если бы ячейка A1 содержала текст *ПЕРВЫЙ КВАРТАЛ*, данная формула возвратила бы текст *Первый квартал*.

Замена одного текста другим

В некоторых ситуациях у пользователя может возникнуть необходимость заменить определенную часть текста строки другим текстом. Например, некоторые из импортируемых в приложение Excel данных содержат знаки звездочки, которые необходимо преобразовать в другой символ. Безусловно, чтобы выполнить замену, можно воспользоваться одной из команд Excel, выбрав в меню Правка команду Заменить. Но, если вы предпочитаете решить эту проблему с помощью формул, воспользуйтесь преимуществами любой из этих двух функций:

1. Функция ПОДСТАВИТЬ заменяет в строке определенный текст или символ. Эту функцию лучше всего использовать, когда известен символ, который необходимо заменить, а не его расположение.
2. Функция ЗАМЕНИТЬ заменяет символ, расположенный в определенном месте в пределах строки. Эта функция используется при замене любого текста при известном его местоположении. Точность самого текста или символа здесь не учитывается.

Ниже приведена формула, в которой используется функция ПОДСТАВИТЬ для замены значения года 2001 на 2002 в строке *Бюджет 2001*. Эта формула возвращает значение *Бюджет 2002*.

=ПОДСТАВИТЬ ("Бюджет 2001"; "2001"; "2002")

Следующая формула использует функцию ПОДСТАВИТЬ для удаления из строки всех пробелов. Другими словами, она заменяет все пробелы пустой строкой и возвращает название компакт-диска Лиз Фаир: *Whitechocolatespaceegg*.

= ПОДСТАВИТЬ ("White chocolate space egg"; " "; "")

Другая, приведенная ниже формула, использует функцию ЗАМЕНИТЬ, чтобы заменить всего лишь один символ, расположенный в пятой позиции, ничего не вставляя вместо него. Говоря иными словами, она просто удаляет пятый символ (дефис) и возвращает текст *Часть 544*.

=ЗАМЕНИТЬ ("Часть-544"; 6; 1; " ")

Безусловно, можно много раз вставлять эти функции, чтобы столько же раз выполнить замену с помощью этой формулы. Обратите внимание, насколько может быть эффективной формула, использующая несколько вложенных функций ПОДСТАВИТЬ. По большому счету, эта формула заменяет в ячейке A1 каждый из семи символов, следующих друг за другом: пробел, дефис, двоеточие, звездочка, символ подчеркивания, открывающая круглая скобка и закрывающая круглая скобка:

=ПОДСТАВИТЬ (ПОДСТАВИТЬ (ПОДСТАВИТЬ (ПОДСТАВИТЬ (ПОДСТАВИТЬ (ПОДСТАВИТЬ (ПОДСТАВИТЬ (A1; " "; ""); "-"; ""); ":"; ""); "*" ; ""); "_" ; ""); " ("; ""); ")"; "")

Поэтому, если ячейка A1 содержит строку *Part-2A - Z(4M1)_A**, то данная формула возвратит *Part2AZ4M1A*.

Поиск и определение расположения символа в строке

Поиск места расположения начальной позиции определенного текста или символа в пределах одной строки в приложении Excel можно осуществить с помощью функций НАЙТИ и ПОИСК:

1. Функция НАЙТИ обеспечивает поиск текста (подстроки) в пределах текстовой строки и возвращает исходную позицию этой подстроки. Пользователь может самостоятельно определить позицию, с которой следует начать поиск текста. Эту функцию удобно использовать для сравнения текста с учетом регистра, но она не поддерживает сравнение групповых символов.
2. Функция ПОИСК обеспечивает поиск определенной подстроки в пределах основной текстовой строки и возвращает значение исходной позиции подстроки. Для этой функции пользователь также может определить место начала поиска. Эта функция используется для поиска нечувствительного к регистру текста или тогда, когда используются групповые символы.

В формуле, приведенной ниже применена функция НАЙТИ, которая возвращает значение 7, то есть номер позиции первого символа *m*, встречающегося в строке. Обратите внимание, что эта формула чувствительна к регистру:

```
=НАЙТИ("m";"Big Mamma Thornton";1)
```

Следующая формула использует функцию ПОИСК и возвращает значение 5 — номер позиции первого, встречающегося символа *m*. В этом случае, регистр не учитывается:

```
=ПОИСК("m";"Big Mamma Thornton";1)
```

Кроме того, в качестве первого аргумента для функции ПОИСК можно использовать следующие групповые символы:

1. Вопросительный знак (?) соответствует любому одиночному символу.
2. Звездочка (*) соответствует любой последовательности символов.

Следующая формула исследует текст в ячейке A1 и возвращает позицию первых трех стоящих последовательно символов, которые содержат между собой знак дефиса. Иначе говоря, формула отыскивает любой символ, следующий за знаком дефиса и любым другим символом. Таким образом, если ячейка A1 содержит текст *Part-A90*, формула возвращает значение 4.

```
=ПОИСК("-?-" ; A1 ; 1)
```

Поиск и замена символа в пределах строки

Функция ПОИСК может использоваться совместно с функцией ЗАМЕНИТЬ, что позволяет заменить часть найденной текстовой строки другой строкой. В действительности функция ПОИСК используется для того, чтобы найти начало расположения символа, с которым затем будет работать функция ЗАМЕНИТЬ.

Например, предположим, что ячейка A1 содержит текст *“Annual Profit Figures”*. Формула, приведенная ниже отыскивает в этой строке слово *“Profit”* и заменяет его на слово *“Loss”*:

```
=ЗАМЕНИТЬ(A1;ПОИСК("Profit";A1);6;"Loss")
```

Следующая формула использует функцию ПОДСТАВИТЬ для достижения того же эффекта, но более квалифицированным образом:

```
=ПОДСТАВИТЬ(A1;"Profit";"Loss")
```

Дополнительные текстовые формулы

Многим пользователям примеры, приведенные в этом разделе, покажутся более сложными, чем те, что уже рассмотрены. Но убедитесь сами, сколько полезных действий с текстом можно осуществить с их помощью.

Подсчет символов в ячейке

Приведенная ниже формула подсчитывает в строке, содержащейся в ячейке A1 количество символов В, учитывая только символы верхнего регистра:

=ДЛСТР (A1) - ДЛСТР (ПОДСТАВИТЬ (A1 ; "В" ; " "))

В формуле используется функция ПОДСТАВИТЬ, создающая в памяти новую строку, из которой удалены все имеющиеся символы В. После этого длина созданной строки вычитается из длины первоначальной строки. В результате формула возвращает количество символов В, содержащихся в первоначальной строке.

Следующая формула более универсальна. Она подсчитывает количество символов В одновременно верхнего и нижнего регистра, содержащихся в ячейке A1.

=ДЛСТР (A1) - ДЛСТР (ПОДСТАВИТЬ (ПОДСТАВИТЬ (A1 ; "В" ; " ") ; "в" ; " "))

Подсчет строк, содержащихся в ячейке

Формулы, приведенные в предыдущем разделе, подсчитывают сколько раз определенный символ встречается в заданной ячейке. Следующая формула работает более, чем с одним символом, и возвращает количество экземпляров определенной подстроки (заданной в ячейке B1) в пределах строки, содержащейся в ячейке A1. Заданная подстрока может состоять из любого количества символов.

= (ДЛСТР (A1) - ДЛСТР (ПОДСТАВИТЬ (A1 ; B1 ; " "))) / ДЛСТР (B1)

Например, если ячейка A1 содержит текст *Blonde on Blonde*, а ячейка B1 текст *Blonde*, то данная формула возвратит значение 2.

Имейте в виду, что, в данном случае, при сравнении текста, учитывается регистр символов. Поэтому, если ячейка B1 будет содержать текст *blonde*, то формула возвратит значение 0. Следующая формула — просто модифицированная версия предыдущей, которая осуществляет поиск без учета регистра (без различия строчных и заглавных букв):

= (ДЛСТР (A1) - ДЛСТР (ПОДСТАВИТЬ (ПРОПИСН (A1) ; ПРОПИСН (B1) , " "))) / ДЛСТР (B1)

Определение заданного числа как порядкового номера

В некоторых случаях пользователю может потребоваться записать определенное число как порядковый номер. Например, в английском тексте *Today is the 21st day of the month*, число 21 преобразуется в порядковый номер, в конец которого добавляются символы *st*.

Символы, которые будут добавляться в конец порядкового номера, полностью зависят от самого номера. Поэтому невозможно привести стандартный пример этой формулы. Каждый пользователь всякий раз может создавать свою собственную сложную конструкцию. Большинство порядковых номеров в английском языке используют суффикс *th*. Исключения составляют номера, закан-

чивающиеся числами 1, 2 или 3. То же самое касается номеров, в которых используется первый разряд — число 1, например, 11, 12 или 13. Эти числа требуют применения определенных правил, которые при желании могут быть трансформированы в формулу приложения Excel.

Формула, приведенная ниже, преобразует число, содержащееся в ячейке A1 (предполагается, что это целое число), в порядковый номер.

`=A1&ЕСЛИ (ИЛИ (ЗНАЧЕН (ПРАВСИМВ (A1; 2)) = {11; 12; 13}); "th"; ЕСЛИ (ИЛИ (ЗНАЧЕН (ПРАВСИМВ (A1)) = {1; 2; 3}); ВЫБОР (ПРАВСИМВ (A1); "st"; "nd"; "rd"); "th"))`

Это довольно сложная формула, поэтому давайте более подробно рассмотрим ее компоненты. Обычно формула работает следующим образом:

1. Суффикс *th* используется в том случае, если последние две цифры порядкового номера представляют собой числа 11, 12 или 13.
2. Если первое правило не подходит к имеющемуся числу, то проверяется последняя цифра. Если последняя цифра 1, то используется суффикс *st*, если 2 — суффикс *nd*, если 3 — суффикс *rd*.
3. Если ни первое, ни второе правило не подходят, используется суффикс *th*.



Обратите внимание, что данная формула использует два массива, указанные в фигурных скобках. Дополнительную информацию об использовании в формулах массивов можно найти в главе 14.

На рис. 5.5, показано как работает эта формула.

Определение литеры столбца по заданному номеру

Следующая формула возвращает литеру столбца рабочего листа (в диапазоне от A до IV) в соответствии со значением, введенным в ячейку A1. Например, если ячейка A1 содержит число 29, то формула возвратит значение AC:

`=ЕСЛИ (A1>26; СИМВОЛ (64+ЦЕЛОЕ ((A1-1) / 26)); "") &СИМВОЛ (65+ОСТАТ (A1-1; 26))`

Имейте в виду, что данная формула не проверяет правильность введенного числового значения номера столбца. Другими словами, если ячейка A1 содержит значение меньше 1 или больше 256, то формула все-таки будет давать результат, хотя уже не имеющий никакого смысла. Ниже приведена модифицированная версия этой формулы, в которую введена функция ЕСЛИ для того, чтобы гарантировать вывод формулой допустимого столбца:

`=ЕСЛИ (И (A1>0; A1<257); ЕСЛИ (A1>26; СИМВОЛ (64+ЦЕЛОЕ ((A1-1) / 26)); "") &СИМВОЛ (65+ОСТАТ (A1-1; 26)); "")`

Извлечение имени файла из заданного пути

Ниже приведена формула, которая возвращает пользователю имя файла, извлеченное из определенного пути. Например, если ячейка A1 содержит путь: `c:\windows\desktop\myfile.xls`, то данная формула возвращает текст `myfile.xls`.

`=ПСТР (A1; НАЙТИ (" "; ПОДСТАВИТЬ (A1; "\ "; "*"); ДЛСТР (A1) - ДЛСТР (ПОДСТАВИТЬ (A1; "\ "; ""))) + 1; ДЛСТР (A1))`

Данная формула предполагает, что разделитель пути представляет собой обратную косую черту (\). По существу, эта формула возвращает весь текст, следующий за последним знаком обратной косой черты. В том случае, если ячейка A1 не содержит этот символ, формула возвращает значение ошибки.

Извлечение первого слова строки

Для того чтобы извлечь первое слово строки, в формуле необходимо задать поиск позиции первого пробела, после чего эта информация используется как аргумент функции ЛЕВСИМВ. Приведенная ниже формула выполняет именно эти действия:

=ЛЕВСИМВ (A1 ; НАЙТИ (" " ; A1) - 1)

Данная формула возвращает весь, находящийся до первого пробела, текст в ячейке A1. Однако и в этой формуле имеется один недочет: она возвращает значение ошибки в том случае, если ячейка A1 состоит из единственного слова. Чтобы решить эту проблему, достаточно ввести в формулу функцию ЕСЛИ, которая будет проверять наличие ошибки. Но, в этом случае, формула становится несколько сложней:

=ЕСЛИ (ЕОШ (НАЙТИ (" " ; A1)) ; A1 ; ЛЕВСИМВ (A1 ; НАЙТИ (" " ; A1) - 1))

Извлечение последнего слова строки

Извлечение последнего слова строки — более сложная задача, поскольку функция НАЙТИ работает только слева направо. Поэтому, вся суть проблемы состоит в определении места нахождения последнего пробела. Однако приведенная ниже формула позволяет решить эту задачу. Данная формула возвращает последнее слово строки, другими словами, весь текст, расположенный после последнего пробела:

=ПРАВСИМВ (A1 ; ДЛСТР (A1) - НАЙТИ ("*" ; ПОДСТАВИТЬ (A1 ; " " ; "*" ; ДЛСТР (A1) - ДЛСТР (ПОДСТАВИТЬ (A1 ; " " ; " ")))))

Тем не менее, эта формула имеет ту же проблему, что и первая, приведенная в предыдущем разделе: она возвращает значение ошибки в том случае, если строка не содержит хотя бы один пробел. Усовершенствовать ее можно тем же способом, что и в приведенном выше примере, а именно, использовать функцию ЕСЛИ для подсчета количества пробелов в ячейке A1. Если в содержимом ячейке пробелы отсутствуют, формула возвращает содержимое ячейки A1 полностью. В противном случае, в работу вступает предыдущая формула.

=ЕСЛИ (ДЛСТР (A1) - ДЛСТР (ПОДСТАВИТЬ (A1 ; " " ; " ")) = 0 ; A1 ; ПРАВСИМВ (A1 ; ДЛСТР (A1) - НАЙТИ ("*" ; ПОДСТАВИТЬ (A1 ; " " ; "*" ; ДЛСТР (A1) - ДЛСТР (ПОДСТАВИТЬ (A1 ; " " ; " "))))))

	A	B	C
1	Number	Ordinal	
2	1	1st	
3	4	4th	
4	7	7th	
5	10	10th	
6	13	13th	
7	16	16th	
8	19	19th	
9	22	22nd	
10	25	25th	
11	28	28th	
12	31	31st	
13	34	34th	
14	37	37th	
15	40	40th	
16	43	43rd	
17	46	46th	
18	49	49th	
19	52	52nd	
20			

Рис. 5.5. Данная формула используется для преобразования заданного числа в порядковое числительное

Извлечение всех слов строки кроме первого

Следующая формула полностью возвращает содержимое ячейки A1, за исключением первого слова:

=ПРАВСИМВ (A1 ; ДЛСТР (A1) -НАЙТИ (" " ; A1 ; 1))

Например, если ячейка A1 содержит текст *2002 Текущий Бюджет*, формула возвращает текст *Текущий Бюджет*.

Извлечение имени, отчества и фамилии

Предположим, что в одном из столбцов рабочего листа содержится именованный список сотрудников фирмы, состоящий из имен, отчеств и фамилий. Что сделать для того, чтобы разделить весь этот перечень на три отдельных столбца: один с именами, другой с отчествами и третий с фамилиями людей? Эта задача более сложна, чем может показаться на первый взгляд, поскольку отчество имеется не во всех строках. Тем не менее, она все-таки решаема.



Задача становится чуть более сложной, если имена, присутствующие в списке, также содержат титулы и звания (такие как, например, Mr. или Dr.) или сопровождаются некоторыми дополнительными подробностями (например, мл. или III). Фактически, приведенные ниже формулы не способны обработать такие сложные случаи. Тем не менее, они позволяют существенно продвинуться в решении этой задачи. Пользователю остается внести лишь несколько корректировок, чтобы вручную обработать частные случаи.

В примере, взятом для следующих ниже формул, предполагается, что список имен содержится в ячейке A1. Формула, возвращающая имя человека, выглядит весьма просто:

=ЛЕВСИМВ (A1 ; НАЙТИ (" " ; A1) - 1)

Full Name	First	Middle	Last
John Q. Public	John	Q.	Public
Lisa Smith	Lisa		Smith
J. R. Robins	J.	R.	Robins
A.P. Adams	A.P.		Adams
Roger Smith	Roger		Smith
Mr. James Olson	Mr.	James	Olson

Рис. 5.6. На этом рабочем листе используются формулы извлечения имен, отчеств и фамилий людей, содержащиеся в списке в столбце A

Сложнее ситуация обстоит со списками отчеств людей, так как эта информация может быть введена не во всех записях. Данная формула возвращает отчество человека (если такое имеется), в противном случае, формула не возвратит ничего:

=ЕСЛИ (ЕОШ (ПСТР (A1 ; НАЙТИ (" " ; A1) + 1 ; ЕСЛИ (ЕОШ (НАЙТИ (" " ; A1 ; НАЙТИ (" " ; A1) + 1) ; НАЙТИ (" " ; A1) ; НАЙТИ (" " ; A1 ; НАЙТИ (" " ; A1) + 1)) - НАЙТИ (" " ; A1) - 1) ; "" ; ПСТР (A1 ; НАЙТИ (" " ; A1) + 1 ; ЕСЛИ (ЕОШ (НАЙТИ (" " ; A1 ; НАЙТИ (" " ; A1) + 1) ; "" ; НАЙТИ (" " ; A1) - 1))

НАЙТИ (" " ; A1) ; НАЙТИ (" " ; A1 ; НАЙТИ (" " ; A1) + 1)) - НАЙТИ (" " ; A1) - 1))

И, наконец, следующая формула возвращает фамилию:

=ПРАВСИМВ (A1 ; ДЛСТР (A1) - НАЙТИ ("*" ; ПОДСТАВИТЬ (A1 ; " " ; "*" ; ДЛСТР (A1) - ДЛСТР (ПОДСТАВИТЬ (A1 ; " " ; " "))))

Однако существует формула, которая позволяет более простым способом извлечь из списка отчества людей. Следующая формула может быть весьма полезна, если изначально использовать две другие формулы для получения имени и фамилии. Для этого необходимо поместить имя человека в ячейку B1, а фамилию в ячейку D1.

=ЕСЛИ (ДЛСТР (B1&D1) + 2 >= ДЛСТР (A1) ; " " ; ПСТР (A1 ; ДЛСТР (B1) + 2 ; ДЛСТР (A1) - ДЛСТР (B1&D1) - 2)

Как видно из рис. 5.6, эти формулы работают достаточно хорошо. Тем не менее, несколько проблем все же существует, особенно с именами, которые содержат четыре "слова". Но, как было сказано ранее, это может быть исправлено вручную.



Для получения более подробной информации о создании приведенных в этом разделе формул, обратитесь к главе 20, в которой рассматриваются мегаформулы.

Удаление из имен титулов и званий

Приведенная здесь формула позволяет отбросить от имени человека три часто употребляемых обращения: *Mr.*, *Ms.* и *Mrs.* Например, если ячейка A1 содержит текст *Mr. Fred Munster*, следующая формула возвращает *Fred Munster*.

`=ЕСЛИ(ИЛИ(ЛЕВСИМВ(A1;2)="Mr";ЛЕВСИМВ(A1;3)="Mrs";ЛЕВСИМВ(A1;2)="Ms");ПРАВСИМВ(A1;ДЛСТР(A1)-НАЙТИ(" ";A1));A1)`

Подсчет количества слов в ячейке

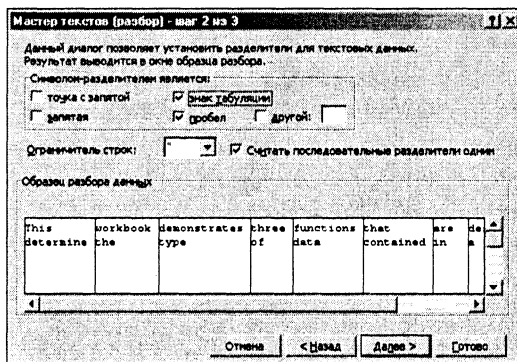
Следующая формула возвращает количество слов в ячейке A1:

`=ДЛСТР(СЖПРОБЕЛЫ(A1))-ДЛСТР(ПОДСТАВИТЬ(СЖПРОБЕЛЫ(A1);" ";""))+1`

В этой формуле используется функция СЖПРОБЕЛЫ, которая удаляет лишние пробелы. После этого с помощью функции ПОДСТАВИТЬ в памяти создается новая строка без пробелов. Затем длина созданной строки вычитается из длины исходной строки, в результате чего определяется количество пробелов. Это значение увеличивается на 1, и, таким образом, подсчитывается количество слов.

Разделение текстовых строк без использования формул

Во многих случаях пользователь может отказаться от применения формул для того, чтобы разобрать строки на составляющие их элементы, и использовать команду Данные⇒Текст по столбцам. При выборе этой команды на экране отображается мастер текстов приложения Excel, который содержит ряд диалоговых окон, позволяющих шаг за шагом преобразовать один столбец, содержащий данные, в несколько столбцов. Обратите внимание, что эффективней всего выбрать разделение данных по символам пробелов, установив переключатель С разделителями на первом шаге мастера и флажок Пробел на втором шаге.



Имейте в виду, что приведенная выше формула возвращает значение 1, если заданная ячейка не содержит данных. Решить эту проблему можно, усовершенствовав формулу следующим образом:

```
=ЕСЛИ(ДЛСТР(A1)=0;0;ДЛСТР(СЖПРОБЕЛЫ(A1)) -  
ДЛСТР(ПОДСТАВИТЬ(СЖПРОБЕЛЫ(A1);" ";" "))+1)
```

Создание собственных текстовых функций с помощью языка VBA

В приложении Excel имеется достаточное количество функций, предназначенных для работы с текстом, но, иногда, пользователь может столкнуться с ситуацией, которой не соответствует ни одна существующая функция. В таком случае пользователю предоставляется прекрасная возможность создать свою собственную функцию для заданного рабочего листа с использованием встроенного языка программирования VBA.



Глава 25 этой книги содержит несколько дополнительных текстовых функций, написанных на VBA. Ниже мы просто кратко рассмотрим их описание.

1. Функция **REVERSETEXT** возвращает в заданную ячейку текст, располагая в нем символы в обратном порядке. Например, если в функцию как аргумент введен текст *Evian*, то возвращается текст *naivE*.
2. Функция **ACRONYM** возвращает первый символ каждого слова, используемого в качестве аргумента функции. Например, при использовании текста *Power Utility Pak*, функция возвращает *PUP*.
3. Функция **SPELLDOLLARS** возвращает число, введенное в качестве аргумента, прописью (как это часто встречается в чеках). Например, используя при вводе аргумент *123,45*, функция возвращает текст *Сто двадцать три и сорок пять сотых*.
4. Функция **SCRAMBLE** возвращает случайный набор символов введенного аргумента. К примеру, введенный в качестве аргумента текст *Microsoft*, возвращается как *oficMorts*, либо в виде другой случайно полученной перестановки символов.
5. Функция **ISLIKE** возвращает значение **ИСТИНА** в том случае, если строка соответствует образцу, составленному из текста и групповых символов.
6. Функция **CELLHASTEXT** возвращает значение **ИСТИНА**, если ячейка, заданная как аргумент, содержит текст или значение в формате **Текстовый**. Эта функция позволяет преодолеть проблемы, которые мы рассматривали в начале этой главы.
7. Функция **EXTRACTELEMENT** извлекает элемент из строки, основываясь на определенном символе-разделителе (например, дефисе).

Резюме

В этой главе рассмотрены некоторые основные принципы управления в приложении Excel текстом, содержащимся в ячейках. Кроме того, здесь представлено много полезных примеров, в которых использованы текстовые функции.

В следующей главе мы рассмотрим формулы, позволяющие вычислять даты, время и другие значения, связанные с временными периодами.

Глава 6

Работа с данными даты и времени

В этой главе...

- ◆ Принципы управления данными даты и времени в Excel
- ◆ Функции управления датами
- ◆ Функции управления временем
- ◆ Резюме

Пользователи, только начинающие работать с приложением Excel, при использовании данных даты и времени часто сталкиваются с различными неприятными проблемами. Скорее всего, это основано на непонимании того, каким образом Excel обрабатывает этот тип информации. В этой главе мы постараемся рассказать пользователю о принципах обращения с данными, представляющими собой дату и время, и о правилах создания и использования эффективных формул, с помощью которых происходит управление этими данными.



Примеры, которые будут использоваться в этой главе, соответствуют формату даты, принятому в русифицированном приложении Excel: месяц.день.год (или день/месяц/год). К примеру, формат даты 1.03.1952 означает 1 марта 1952 года. Имейте в виду, что это не 3 января 1952 года, как принято в формате англоязычного приложения. Формат даты англоязычного программного обеспечения может показаться пользователю не совсем логичным. Тем не менее, в американских параметрах настройки системы используется именно этот формат.

Принципы управления данными даты и времени в Excel

В этом разделе мы кратко рассмотрим, как приложение Excel управляет информацией, представляющей собой дату и время. Раздел включает обзор основных принципов работы с порядковыми значениями даты и времени и предлагает краткие советы по вводу и приведению в соответствующий формат этого типа информации.



Информация, связанная с датой, приведена и в других главах этой книги. К примеру, в главе 7 приведены примеры расчета значений ячеек, использующих формат даты, а в главе 25 — некоторые полезные функции управления датами, написанные на VBA.

Понятие даты как порядкового номера

Для Excel дата — это обычное число. А точнее, дата — это “порядковое числительное”, которое представляет собой порядковый номер дня, начиная с 0 января 1900 года. Это означает, что порядковый номер дня 1 соответствует 1 января 1900 года; порядковый номер 2 — 2 января 1900 года и так далее. Существование такой системы позволяет использовать формулы при работе с датами. Например, пользователь может создать формулу, которая будет вычислять количество дней между двумя датами.

Безусловно, понятие 0 января 1900 года может вызвать недоумение. Конечно, оно не является датой, которая соответствует нулевому дню, а, фактически, означает определенное время дня. Чуть позже в этой главе мы рассмотрим это понятие более подробно.

Для того чтобы представить числовое значение в виде даты, к ячейке, содержащей этот номер, необходимо применить формат Дата. Для этого вызовите диалоговое окно Формат ячеек и щелкните на вкладке Число, а затем выберите необходимый формат.



Excel 97 и более поздние версии этого приложения поддерживают даты в диапазоне от 1 января 1900 года до 31 декабря 9999 года (порядковый номер этой даты 2 958 465). Диапазон дат, доступный в более ранних версиях приложения, значительно меньше. Здесь используются даты, начиная с 1 января 1900 года и до 31 декабря 2078 года (порядковый номер 65 380).

Выберите систему дат: 1900 или 1904

Фактически, Excel поддерживает две системы дат: система дат, начиная с 1900 года, и система с 1904 года. Какую из этих систем лучше всего использовать в рабочей книге, пользователь определяет самостоятельно, в зависимости от того, какая дата служит основанием для других дат. В системе дат, начинающейся с 1900 года день 1 января 1900 года имеет порядковый номер 1. В системе 1904 года дата отсчета — 1 января 1904 года. По умолчанию для систем Windows приложение Excel использует систему дат, начинающую отсчет с 1 января 1900 года, а для систем Macintosh — дату, начиная с 1904 года. Тем не менее, и для систем Windows, и для обеспечения совместимости с файлами Macintosh, Excel также поддерживает систему дат 1904. Систему дат можно изменить, используя диалоговое окно Параметры. Чтобы вызвать его, выберите команду Сервис⇒Параметры, а затем щелкните на вкладке Вычисления. Имейте в виду, что система дат не может изменяться, если на компьютере используется приложение Excel для Macintosh.

На самом деле, лучше всего использовать систему дат, установленную по умолчанию — начиная с 1900 года. Будьте особенно осторожны, если используете две различные системы дат в связанных рабочих книгах. К примеру, предположим, что Книга1 использует систему дат, начиная с 1904 года, и содержит в ячейке A1 дату 15/1/1999. Книга2, наоборот, использует систему дат с 1900 года, и содержит некоторую ссылку на ячейку A1 книги Книга1. В книге Книга2 эта дата будет отображаться как 14/1/1995. Таким образом, обе рабочие книги используют один и тот же порядковый номер дня (34 713), но в той и в другой книгах он будет интерпретироваться по-разному.

Существует одно преимущество использования системы дат 1904. Эта система позволяет отображать отрицательные значения времени. Другими словами, при использовании системы дат 1900 решение такой задачи как 4:00 PM - 5:30 PM не может быть отображено. При использовании системы дат 1904, результат отображается как отрицательное время -1:30 (то есть разница составляет один час и 30 минут).

Введение даты

Конечно, пользователь может вводить дату как числовое значение, но, как правило, дата вводится с использованием любого из нескольких поддерживаемых в Excel форматов даты. После введения даты приложение Excel автоматически преобразует ее в соответствующий

этой дате порядковый номер, который в дальнейшем используется для вычислений. Затем в ячейке применяется заданный по умолчанию формат даты и на экране отображается привычный всем вид информации о дате, нежели загадочный порядковый номер.

Например, если пользователю необходимо ввести дату 1 июня 2002 года, то это можно сделать весьма просто — *1 Июнь, 2002*, используя этот формат или любой другой из имеющихся форматов даты. Excel преобразует введенную информацию и сохраняет ее как числовое значение 37 408 — порядковый номер, соответствующий этой дате. После этого Excel применяет формат даты, заданный по умолчанию, поэтому содержимое ячейки может отображаться в представлении, несколько отличающемся от введенного пользователем.



В зависимости от региональных стандартов системы пользователя, дата, введенная в формате 1 Июнь, 2002, может быть интерпретирована как текстовая строка. В общем случае можно использовать указанный формат 1 Июнь, 2002.

Обратите внимание, что при активизации ячейки, содержащей дату, в строке формул отображается ее содержимое, представленное в заданном по умолчанию формате, который соответствует сокращенному виду даты, принятому в системе. Имейте в виду, что строка формул не отображает порядковый номер даты. Чтобы определить порядковый номер даты, просто перейдите от формата даты к числовому формату ячейки.



Чтобы изменить заданный по умолчанию формат даты, необходимо изменить региональные настройки самой системы. Для этого, находясь в Windows, откройте папку Панель управления, запустите апплет Язык и региональные стандарты, щелкните на кнопке Настройка, а затем перейдите на вкладку Дата. Краткий формат даты, заданный по умолчанию в Excel, определяет опция, выбранная из раскрывающегося списка Краткий формат. Просто измените текущее представление на любое другое.

Обратите внимание, в табл. 6.1 приведены образцы форматов даты, которые поддерживает Excel (использующиеся в параметрах настройки для России и Украины). Однако они могут отличаться, если на компьютере используются настройки другого региона.

Таблица 6.1. Преобразование введенных дат в форматы, поддерживаемые Excel

Введенные данные	Воспринимается в Excel (настройка для России/Украины)
1-6-01	1 июня 2001 года
1-6-2001	1 июня 2001 года
1/6/01	1 июня 2001 года
1/6/2001	1 июня 2001 года
1-6/01	1 июня 2001 года
1 Июнь, 2001	1 июня 2001 года
1 Июн	1 июня текущего года
1 Июнь	1 июня текущего года
1/6	1 июня текущего года
1-6	1 июня текущего года
1-июн-2001	1 июня 2001 года
2001/6/1	1 июня 2001 года

Рассматривая примеры, приведенные в этой таблице, можно сделать вывод, что Excel обладает достаточно широкими интеллектуальными возможностями преобразования введенных дат. Однако и она не вполне совершенна. Например, она не признает в качестве даты следующую введенную информацию:

1. 1 июня 2001
2. Июн-1 2001
3. Июн-1/2001

Эту информацию Excel понимает как текст. Поэтому, прежде чем использовать даты в формулах, убедитесь, что Excel поддерживает тот формат даты, который вы вводите. В противном случае, формулы, которые обращаются к этим датам, будут работать некорректно.

Существует еще один недостаток. Если введенная дата находится вне поддерживаемого диапазона дат, Excel также интерпретирует ее как текст. И, если порядковый номер, находящийся за пределами доступных для даты числовых значений, попытаться привести к формату даты, вместо значения будет отображаться ряд знаков “решетки” (#####).

Понятие времени как порядкового номера

Работа с числовыми значениями, представляющими собой характеристики времени — это не что иное, как простая детализация (то есть деление на десятичные доли) порядковых номеров, используемых в выражениях даты. Другими словами, при работе со временем приложение Excel использует разделение на дни. К примеру, если порядковый номер даты 1 июня 2001 года — это 37 043; полдень того же дня будет иметь значение 37 043,5.

Поиск даты

Иногда, если рабочий лист использует много дат, может возникнуть необходимость найти определенную дату. Для этого воспользуйтесь диалоговым окном Найти и заменить Excel, которое можно открыть, выбрав команду меню Правка⇒Найти или нажав комбинацию клавиш <Ctrl+F>. Имейте в виду, что Excel достаточно требователен к точности ввода искомой информации. Вам придется ввести полную четырехразрядную дату в поле Найти, находящемся в этом диалоговом окне. Формат даты должен соответствовать краткому представлению даты, принятому в системе. Это представление можно увидеть в строке формул.

Порядковый номер одной минуты эквивалентен приблизительно 0,00069444. Формула, приведенная ниже, вычисляет этот номер, умножая 24 часа на 60 минут, а затем разделив 1 на полученный результат. Таким образом, в знаменателе получается количество минут в одном дне (1 440).

$$=1 / (24 * 60)$$

Тем же способом в Excel определяется порядковый номер одной секунды — приблизительно 0,00001157. Для этого 1 делится на 24 часа, умноженные на 60 минут и 60 секунд. В этом случае, знаменатель содержит количество секунд в одном дне (86 400).

$$=1 / (24 * 60 * 60)$$

Самая маленькая единица измерения времени в приложении Excel — одна тысячная доля секунды (миллисекунда). Порядковый номер приведенного ниже значения представляет время 23:59:59,999 или без одной тысячной секунды полночь:

$$0,99999999$$

В табл. 6.2 приведены различные время дня и соответствующие ему порядковые номера, по которым Excel распознает это время.

Таблица 6.2. Время дня и преобразование его в порядковый номер

Время дня	Порядковый номер времени
12:00:00 AM (полночь)	0,00000000
1:30:00 AM	0,06250000
3:00:00 AM	0,12500000
4:30:00 AM	0,18750000
6:00:00 AM	0,25000000
7:30:00 AM	0,31250000
9:00:00 AM	0,37500000
10:30:00 AM	0,43750000
12:00:00 PM (полдень)	0,50000000
1:30:00 PM	0,56250000
3:00:00 PM	0,62500000
4:30:00 PM	0,68750000
6:00:00 PM	0,75000000
7:30:00 PM	0,81250000
9:00:00 PM	0,87500000
10:30:00 PM	0,93750000

Введение времени

Как и в случае с датами, пользователю нет необходимости вводить порядковый номер времени. Достаточно ввести в ячейку время в любом из форматов, поддерживаемых в Excel. Обратите внимание, в табл. 6.3 приведены некоторые примеры форматов времени, поддерживаемые в Excel.

Таблица 6.3. Преобразование введенного времени в формат Excel

Введенные данные	Преобразование Excel
11:30:00 am	11:30 AM
11:00:00 AM	11:30 AM
11:30 pm	11:30 PM
11:30	11:30 AM
13:30	1:30 PM

Поскольку приведенные выше примеры времени не связаны ни с каким определенным днем, Excel по умолчанию использует порядковый номер 0, соответствующий 0 января 1900 года, который, по своей сути, не является датой. Но в большинстве случаев у пользователя возникает необходимость объединить дату и время. Чтобы осуществить эту задачу, используйте совместный формат даты и времени, который состоит из формата даты, пробела и формата времени. Например, если в ячейку ввести текст 11:30 1

июнь, 2001, приложение Excel вычислит порядковый номер 37 043,4791666667 и интерпретирует его в доступном формате даты и времени:

01.06.2001 11:30:00

В некоторых случаях, когда вводится время, превышающее 24 часа, связанная с ним дата соответственно увеличивается. Например, если ввести в ячейку время 25:00:00, Excel увеличит ту часть, которая относится к дню, поскольку она превышает 24 часа и отобразит введенное значение как:

01.01.2000 1:00:00

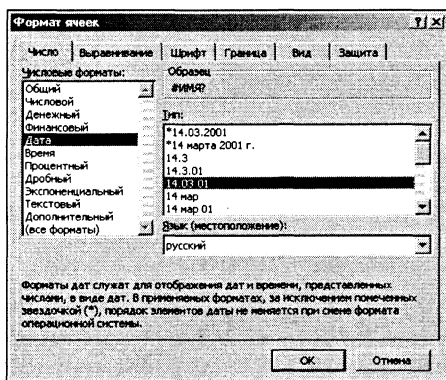
То же самое происходит, если ввести одновременно и дату, и время, при этом время, превышающее 24 часа. Введенная дата при отображении будет откорректирована. Например, при вводе в ячейку 01.09.1999 25:00:00 на экране отобразится следующая информация:

02.09.1999 1:00:00

Обратите внимание, что, если вводится только время, без соответствующей даты, то максимальное время, которое можно ввести в ячейку, — 9999:59:59, то есть менее 10 000 часов. В этом случае Excel добавляет соответствующее количество дней и на экране отображается пересчитанная дата. Если же ввести время, превышающее предел 10 000 часов, то оно будет появляться как простая текстовая строка.

Форматирование даты и времени

Работая с ячейками, содержащими любые даты и время, пользователь имеет прекрасную



возможность форматировать их так, как это ему необходимо. Например, можно установить такой формат ячейки, чтобы в ней отображалась только определенная часть даты или времени, или только заданные части времени и даты одновременно.

Чтобы форматировать дату или время, выделите необходимую ячейку, а затем вызовите диалоговое окно **Формат ячеек** и перейдите на вкладку **Число** (рис. 6.1). После этого в поле со списком **Числовые форматы** выберите необходимый тип формата и щелкните на кнопке **OK**. В категории **Дата** отображаются форматы даты, поддерживаемые в Excel, а в категории **Время** — встроенные форматы времени. Существует также несколько форматов, которые могут отображать дату и время одновременно.

Рис. 6.1. Чтобы изменить текущий формат ячейки для отображения даты и времени, вызовите диалоговое окно **Формат ячеек** и перейдите на вкладку **Число**



При составлении формул, содержащих ссылку на ячейку, содержимое которой представляет собой дату или время, Excel автоматически преобразует эти формулы в формат даты или времени. В некоторых случаях, это может быть весьма кстати, в других — создает определенные неудобства и даже раздражает пользователя. К сожалению, Excel не дает никакой возможности отключить автоматическое форматирование ячейки. Тем не менее, в подобных случаях, можно воспользоваться определенной комбинацией клавиш, чтобы отменить форматирование всех чисел ячейки и вернуться к формату **Общий**, заданному по умолчанию. Для этого достаточно выделить ячейку и нажать комбинацию клавиш **<Ctrl+Shift+~>**.

Иногда пользователь может столкнуться с тем, что ни один из встроенных форматов не подходит для заданного представления даты и времени. Не отчаивайтесь. Excel предоставляет пользователю прекрасную возможность создавать свои собственные числовые форматы. Просто выберите из списка категорию (**все форматы**), а затем введите собственный тип формата в поле Тип. (Для получения дополнительной информации о создании собственных числовых форматов см. Приложение В).



Следующий собственный пользовательский формат особенно полезен для отображения времени:

[ч] : мм : сс

Заключение часов в квадратные скобки позволяет отображать часы за пределами 24. Использование этого формата может быть очень эффективно при добавлении более 24 часов. Пример использования этого формата мы будем рассматривать в этой главе несколько позже в разделе “Суммирование времени, превышающего 24 часа”.

Трудности, возникающие при использовании даты

Несмотря на всю свою эффективность, Excel все же имеет несколько проблем, возникающих при работе с датой. Большинство из них уходит далеко в прошлое. Excel была разработана много лет назад, еще раньше, чем проблема Y2K впервые потрясла мир своим существованием. Но, как мы уже говорили, во многом разработчики Excel отталкивались от Lotus 1-2-3, с ее ограниченными возможностями работы с датой и временем. Поэтому все самые ужасные технические дефекты были непреднамеренно перенесены в Excel. Кроме того, версии самой Excel имеют между собой некоторую несовместимость в отношении того, как в них интерпретируется запись в ячейке, которая содержит год, состоящий из двух цифр. И, наконец, то, как Excel понимает запись даты в ячейке, напрямую зависит от региональных настроек, установленных на компьютере пользователя.

Безусловно, если бы Excel разрабатывалась в настоящее время, начиная с нуля, она была бы гораздо совершенней в отношении работы с датами. Но, к сожалению, сейчас мы вынуждены работать с программным продуктом, который все же оставляет желать лучшего.

Ошибка високосного года

Високосный год, который наступает через каждые четыре года, имеет один дополнительный день — 29 февраля. Несмотря на то, что 1900 год не был таковым, Excel воспринимает его как високосный. Другими словами, если ввести в ячейку 29.02.1900, Excel не выдаст сообщение об ошибке, а интерпретирует это как значение даты, соответствующее 60 дню с начала отсчета.

Однако, если ввести другую недопустимую дату 29.02.1901, Excel выдаст сообщение об ошибке. Приложение не преобразует эту запись в дату, а, скорее, воспримет ее как текстовую строку.

Возникает вопрос: как программный продукт, который ежедневно используют миллионы людей, может содержать такую очевидную ошибку? Ответ кроется в истории его создания. Исходная версия Lotus 1-2-3, на которой базировалось приложение Excel, содержала дефект, по причине которого 1900 год рассматривался как високосный. Разработчики приложения Excel знали об этой ошибке и, позднее, при выпуске приложения, специально перенесли этот дефект, чтобы обеспечить его совместимость с рабочими листами файлов Lotus.

Но почему же эта ошибка все еще существует в более поздних версиях Excel? Компания Microsoft утверждает, что недостатки, которые могут возникнуть при исправлении этой ошибки, значительно перевешивают преимущества. Если бы существующая ошибка была устранена, это привело бы в беспорядок сотни тысяч существующих рабочих книг. Кроме то-

го, исправление этой проблемы сказалось бы на совместимости самого приложения Excel с другими программами, использующими дату. Именно поэтому, в действительности, эта ошибка влечет за собой не так уж и много проблем, как ее исправление. К тому же, большинство пользователей не использует даты, ранее 1 марта 1900 года.

Использование дат до 1900 года

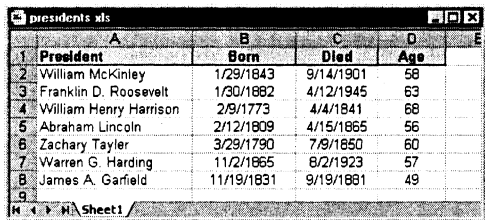
Безусловно, мир не начал свое существование с 1 января 1900 года. Многие пользователи Excel, которые работают с исторической информацией, часто используют даты до 1 января 1900 года. Но, к сожалению, единственный способ работы с такими датами — это ввести их в ячейку как текстовую информацию. Например, если ввести в ячейку следующую строку, Excel воспримет ее без сообщения об ошибке:

4 июль, 1776

Однако, датами, принимаемыми в виде текста, уже нельзя манипулировать как номинальными, измеряемыми в единицах времени. Эти даты нельзя форматировать, изменяя числовой формат, как это принято в Excel. Нельзя определить день недели, на который приходится эта дата, а также вычислить дату, которая следует семью днями позже.

Ввод несовместимых дат

Будьте осторожны при вводе дат, использующих два разряда для отображения года. В этом случае Excel применяет некоторые правила, которые определяют, какое столетие необходимо использовать. Эти правила изменяются в зависимости от версии Excel, которая установлена в компьютере.



	A	B	C	D
1	President	Born	Died	Age
2	William McKinley	1/29/1843	9/14/1901	58
3	Franklin D. Roosevelt	1/30/1882	4/12/1945	63
4	William Henry Harrison	2/9/1773	4/4/1841	68
5	Abraham Lincoln	2/12/1809	4/15/1865	56
6	Zachary Taylor	3/29/1790	7/9/1850	60
7	Warren G. Harding	11/2/1865	8/2/1923	57
8	James A. Garfield	11/19/1831	9/19/1881	49

Рис. 6.2. Настройка *Extended Date Function* позволяет работать с датами до 1900 года

управлением Windows 98 и выше, в качестве граничного года, заданного по умолчанию, используется 2029 год. Но пользователь может самостоятельно установить граничный год. Для этого воспользуйтесь папкой Панель управления. Откройте диалоговое окно Язык и региональные стандарты, щелкните на кнопке Настройка и перейдите на вкладку Дата.

В более ранних версиях приложения Excel, начиная с Excel 3 и до Excel 95, временные периоды сдвинуты еще дальше. Годы, отображенные с помощью двух чисел в пределах между 00 и 19, интерпретируются как даты 21-ого столетия, а годы между 20 и 99 — как даты 20-го столетия. Если в этих версиях Excel пользователь вводит дату 05.12.19, то Excel воспринимает ее как 5 декабря 2019 года, но, если ввести дату 05.12.20, то Excel воспримет ее уже как 5 декабря 1920 года.

Если же пользователь, по какой-либо причине до сих пор использует Excel 2, то ввод любой даты с двумя числами в качестве года всегда интерпретируется как дата 20-го столетия. В табл. 6.4 сведены все отличия, встречающиеся в различных версиях приложения Excel.

Таблица 6.4. Интерпретация года, представленного двумя числами, в различных версиях приложения Excel

Версия Excel	Годы 20-го столетия	Годы 21-го столетия
2	00-99	Н/Д
3, 4, 5, 7, (95)	20-99	00-19
8 (97), 9 (2000), 10 (2002)	30-99	00-29

Тем не менее, чтобы избежать всякого рода неожиданностей при работе с датами, просто введите год полностью, используя все четыре цифры.

Функции управления датами

Excel имеет не так много функций, предназначенных для работы с датой. Эти функции расположены в категории Дата и время диалогового окна Мастер функций.

Обратите внимание, в табл. 6.5 приведены все функции, связанные с использованием даты, доступные в приложении Excel. Но некоторые из них требуют установки надстройки "Пакет анализа".

Таблица 6.5. Функции, связанные с датой

Функция	Описание функции
ДАТА	Возвращает дату в числовом формате Microsoft Excel для заданной даты
ДАТАЗНАЧ	Преобразует дату из текстового формата в числовой в коде Excel
ДЕНЬ	Преобразует дату в день месяца (число от 1 до 31)
ДНЕЙ360	Вычисляет количество дней между двумя датами на основе 360-дневного года (двенадцать месяцев по 30 дней)
ДАТАМЕС*	Возвращает дату в числовом формате, отстоящую на заданное число месяцев вперед или назад от начальной даты
КОНМЕСЯЦА*	Возвращает дату в числовом формате для последнего дня месяца, отстоящего вперед или назад на заданное число месяцев
МЕСЯЦ	Возвращает номер месяца (число в диапазоне от 1 до 12), соответствующий введенному порядковому номеру даты
ЧИСТРАБДНИ*	Возвращает количество рабочих дней, расположенных между двумя датами
ТДАТА	Возвращает текущую дату и время в числовом формате
СЕГОДНЯ	Возвращает текущую дату в виде даты в числовом формате
ДЕНЬНЕД	Преобразует дату в числовом формате в номер дня недели (число от 1 до 7)
НОМНЕДЕЛИ*	Возвращает номер недели в году
РАБДЕНЬ*	Возвращает дату в числовом формате, отстоящую вперед или назад на заданное количество рабочих дней
ГОД	Преобразует дату в числовом формате в год (целое число в диапазон от 1900 до 9999)
ДОЛЯГОДА*	Возвращает долю года, которая составляет количество дней между начальной и конечной датами

* Функции доступны только в том случае, если установлена надстройка "Пакет анализа"

Отображение текущей даты

Функция, приведенная ниже, отображает в ячейке текущую дату:

=СЕГОДНЯ()

Можно отображать не только одну лишь текущую дату, но и объединять ее с любым текстом. Например, следующая формула будет выводить на экран текст: *Сегодня понедельник, 9 Апрель, 2001.*

= "Сегодня " & ТЕКСТ(СЕГОДНЯ()); "дддд, д ММММ, гггг"

Важно помнить, что функция СЕГОДНЯ изменяется каждый раз при расчете рабочего листа. Например, если ввести в ячейку рабочего листа любую из рассмотренных формул, они будут отображать текущую дату. Но откройте этот же рабочий лист на следующий день и вы увидите, что формулы отображают уже совсем другую, но все же текущую дату.



Чтобы ввести в ячейку дату как временную отметку, которая не будет изменяться каждый последующий день, воспользуйтесь комбинацией клавиш <Ctrl+>. Текущая дата будет просто введена в ячейку без использования формулы, поэтому она не будет изменяться.

Отображение различных дат

Как мы уже говорили ранее в этой главе, дату можно добавить в ячейку, непосредственно введя ее в любом формате, поддерживаемом в приложении Excel. Дату также можно создать, воспользовавшись функцией ДАТА. Эта функция использует три аргумента: год, месяц и день. Например, приведенная ниже формула возвращает дату, состоящую из года, взятого из ячейки A1, месяца, взятого из ячейки B1, и дня — из ячейки C1.

=ДАТА(A1;B1;C1)



Обратите внимание, что функция ДАТА принимает также и недопустимые аргументы и приводит результат в соответствие. Например, следующая формула использует число 13 как аргумент месяца и возвращает 1 Январь, 2002. Аргумент месяца автоматически преобразуется в первый месяц следующего года.

=ДАТА(2001;13;1)

Достаточно часто функция ДАТА использует другие функции в качестве аргументов. Например, в приведенной ниже формуле функции ГОД и СЕГОДНЯ, используемые как аргументы, возвращают День Независимости (4 Июль, 2002) текущего года:

=ДАТА(ГОД(СЕГОДНЯ());7;4)

Следующая функция — ДАТАЗНАЧ преобразует текстовую строку, представляющую собой дату, в порядковый номер даты. Следующая формула возвращает значение порядкового номера даты 22 Август, 2002 — 37 490:

=ДАТАЗНАЧ("22.8.2002")

Для того чтобы отобразить результат этой формулы в виде даты, достаточно применить к этой ячейке формат даты.



Будьте очень внимательны при использовании функции ДАТАЗНАЧ. Текстовая строка, которая может быть использована как дата в одной стране, совершенно не подходит для ввода даты в другой стране. Предыдущий пример работает прекрасно в том случае, если система отрегулирована для форматов даты России/Украины. Однако при использовании других региональных форматов даты, формула может возвращать ошибку, поскольку приложение Excel будет искать восьмой день двадцать второго месяца!

Создание ряда последовательных дат

В отдельных случаях у пользователя может возникнуть необходимость вставить в рабочий лист целый ряд последовательных дат. Например, при еженедельном отслеживании продаж нужно ввести несколько последовательных дат с интервалом в семь дней. Эти даты будут служить основой для систематизации коммерческой информации.

Самый эффективный способ ввода ряда последовательных дат не требует никаких формул. Просто воспользуйтесь компонентом, встроенным в Excel, — Автозаполнение. Для этого введите в ячейку первую дату, а затем, удерживая нажатой правую кнопку мыши, перетащите в нужном направлении маркер выделения. Отпустите правую кнопку мыши и выберите из контекстного меню команду Заполнить по дням (рис. 6.3).

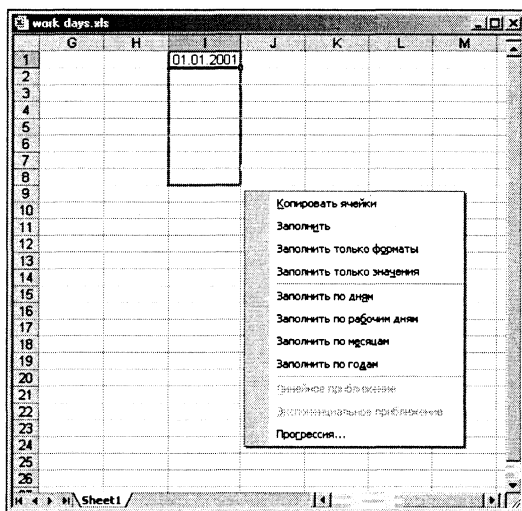


Рис. 6.3. Использование средства автозаполнения позволяет создавать в Excel ряд последовательных дат

Но использование формул все-таки имеет свои преимущества. В этом случае для создания ряда дат достаточно указать первую дату, после чего остальные автоматически приводятся в соответствие на ее основе. Для этого введите в ячейку начальную дату, а затем добавьте формулы, скопировав их вниз по столбцу, чтобы получить даты, следующие за исходной.

В следующем примере мы введем первую дату ряда в ячейку A1, а формулу в ячейку A2. Скопируйте формулу в ячейке вниз по столбцу столько раз, сколько это необходимо.

Эта формула позволяет создать ряд последовательных дат, следующих одна за другой с интервалом в семь дней:

=A1+7

Чтобы создать ряд последовательных дат,стоящих друг от друга на месяц, воспользуйтесь следующей формулой:

=ДАТА (ГОД (A1) ; МЕСЯЦ (A1) +1 ; ДЕНЬ (A1))

Формула, приведенная ниже, возвращает даты, следующие одна за другой через год:

=ДАТА (ГОД (A1) +1 ; МЕСЯЦ (A1) ; ДЕНЬ (A1))

Чтобы создать ряд, состоящий из дат, относящихся только к рабочим дням, исключая субботы и воскресенья, воспользуйтесь приведенной ниже формулой. Имейте в виду, что эта формула предполагает, что дата, введенная в ячейку A1, рабочий день.

=ЕСЛИ (ДЕНЬНЕД (A1) =6 ; A1+3 ; A1+1)

Преобразование в дату строки, не означающей дату

Даты, импортируемые как текстовые строки, также можно преобразовать в дату. К примеру, возьмем текстовую строку, которая в действительности представляет собой дату 21 августа 2001 года. Этот текст состоит из четырех знаков, относящихся к году, двух знаков месяца и двух знаков дня:

20010821

Чтобы преобразовать эту строку в фактическую дату, воспользуйтесь формулой, приведенной ниже. Эта формула предполагает, что исходные данные занесены в ячейку A1:

=ДАТА (ЛЕВСИМВ (A1 ; 4) ; ПСТР (A1 ; 5 ; 2) ; ПРАВСИМВ (A1 ; 2))

Эта формула использует текстовые функции ЛЕВСИМВ, ПСТР и ПРАВСИМВ для того, чтобы извлечь из заданной строки определенные числа, после чего она использует их как аргументы функции ДАТА.



Для получения дополнительной информации об использовании формул управления текстом, обратитесь к главе 5.

Вычисление количества дней между двумя датами

Самый обычный тип вычислений, который может применяться в отношении даты — это определение количества дней между двумя заданными датами. Предположим, что на одном из листов рабочей книги вычисляется процент, получаемый по срочному вкладу. Размер полученного процента зависит от количества дней, на которое открыт счет. И, если рабочий лист содержит дату открытия и дату закрытия счета, то можно легко определить срок в днях, на который был открыт счет.

Поскольку даты на рабочем листе, как правило, сохраняют свои последовательные значения, для того чтобы вычислить количество дней, можно использовать простое вычитание. Предположим, что ячейки A1 и B1 содержат некоторые даты. Следующая формула возвращает количество дней между этими датами:

=A1-B1

Скорее всего приложение Excel автоматически отформатирует результирующую ячейку как дату, а не как значение. Вам необходимо будет изменить формат этой ячейки на числовой. Если ячейка B1 содержит более позднюю дату, чем ячейка A1, то результат, возвращаемый формулой будет отрицательным.



Если приведенная выше формула отображает неправильное значение, убедитесь, что ячейки A1 и B1 на самом деле содержат даты, а не текстовую информацию, только напоминающую дату.

В некоторых случаях вычисление разницы между двумя датами может потребовать значительных усилий. Чтобы продемонстрировать это, проведем аналогию с “телеграфными столбами”. Например, если стоит задача определить количество телеграфных столбов, то она может иметь два решения. Во-первых, можно непосредственно сосчитать количество столбов, а во-вторых можно определить число пролетов между столбами. При этом имейте в виду, что число самих столбов всегда на один больше, чем число пролетов.

Чтобы рассмотреть эту ситуацию применительно к некоторому диапазону дат, возьмем для примера кампанию по продвижению на рынке определенного товара. Предположим, что начальная дата кампании по продвижению товара — 1 февраля, а завершающая дата — 9 февраля. Просто вычитая одну цифру из другой, мы получим ответ: восемь дней. Но в действительности срок продвижения товара — девять дней. В этом случае, чтобы получить правильный ответ, необходимо считать дни “включительно”, а не определять количество дней между датами. Ниже приведена формула, позволяющая вычислить длительность кампании по продвижению товара (предположим, что мы имеем ячейки соответствующие этим названиям):

=ПоследнийДень-НачальныйДень+1

Вычисление количества рабочих дней между двумя датами

Вычисление количества дней между двумя датами — это еще не все возможности, которыми обладает Excel. В зависимости от необходимости пользователь может исключать из числа этих дней выходные и праздничные дни. Например, определим, сколько рабочих дней выпадает на ноябрь месяц. Для этого необходимо исключить из расчета субботы, воскресенья и праздничные дни, если таковые имеются. Это можно сделать с помощью функции ЧИСТРАБДНИ. (Как мы уже говорили, эта функция доступна, если установлена надстройка “Пакет анализа”.)



Название функции — ЧИСТРАБДНИ часто вводит пользователя в заблуждение. Тем не менее, она не имеет никакого отношения к чистоте или рабству, а, скорее, определяет схему рабочих дней между двумя заданными датами.

Итак, функция ЧИСТРАБДНИ вычисляет разницу между двумя датами, исключая выходные дни (субботы и воскресенья). Кроме того, как дополнительное условие, пользователь может определить диапазон ячеек, содержащий даты, на которые приходится праздничные дни. Эти даты также могут исключаться при расчете. Естественно, Excel не имеет никакой возможности самостоятельно определять праздничные дни, поэтому пользователь должен обеспечить его этой информацией, указав заданный диапазон ячеек.

Обратите внимание, на рис. 6.4 отображен рабочий лист, на котором вычисляется количество рабочих дней между определенными датами. В диапазоне A2:A11 содержится пере-

чень дат, приходящихся на праздничные дни. А формулы, введенные в столбце С, вычисляют количество рабочих дней, расположенных между датами в столбце А и столбце В. К примеру, так будет выглядеть формула в ячейке С15:

=ЧИСТРАВДНИ (А15; В15; А2 : А11)

	A	B	C
1			
2	01.01.2002	New Year's Day	
3	21.01.2002	Martin Luther King Jr. Day	
4	18.02.2002	Presidents' Day	
5	27.06.2002	Memorial Day	
6	04.07.2002	Independence Day	
7	02.09.2002	Labor Day	
8	11.11.2002	Veterans Day	
9	14.10.2002	Columbus Day	
10	28.11.2002	Thanksgiving Day	
11	25.12.2002	Christmas Day	
12			
13			
14	First Day	Last Day	Working Days
15	вторник 1.1.2002	понедельник 1.7.2002	4
16	вторник 1.1.2002	вторник 12.31.2002	251
17			
18			
19			
20			
21			

Рис. 6.4. Чтобы определить количество рабочих дней в промежутке между двумя заданными датами, воспользуйтесь функцией ЧИСТРАВДНИ

Формула возвращает значение 4. Это означает, что семидневный период, начинающийся с 1 января, состоит из четырех рабочих дней. Другими словами, при расчете исключаются один праздничный день, один субботний и один воскресный день. Ячейка С16 содержит формулу, которая вычисляет общее количество рабочих дней в году.

Вычисление даты по графику рабочих дней

Следующая функция — РАВДЕНЬ — также доступна только в том случае, если установлена надстройка Excel “Пакет анализа”, и является логичным продолжением функции ЧИСТРАВДНИ. Например, при запуске в работу определенного проекта, требующего на выполнение десять дней, функция вычисляет дату окончания проекта.

В приведенной ниже формуле с помощью функции РАВДЕНЬ определяется дата, следующая через десять рабочих дней после 4 января 2001 года. Рабочими днями считаются будние дни недели (с понедельника по пятницу).

=РАВДЕНЬ ("04.01.2001"; 10)

Формула возвращает значение 18 января 2001 года (между 4 и 18 января выпадает четыре выходных дня).



Имейте в виду, что результат, возвращаемый данной формулой, полностью зависит от региональных настроек даты. В других регионах жестко установленная дата может интерпретироваться как 1 апреля 2001 года. Чтобы избежать этого, используйте следующую формулу:

=РАВДЕНЬ (ДАТА (2001; 1; 4); 10)

Интересно, что второй аргумент функции РАБДЕНЬ может быть отрицательным. И, как и в случае с функцией ЧИСТРАВДНИ, функция РАБДЕНЬ может использовать третий необязательный аргумент, содержащий диапазон ячеек с датами, относящимися к праздничным дням и дням отпуска.

Вычисление количества лет между двумя датами

Следующая формула вычисляет количество лет между двумя датами. Здесь предполагается, что обе ячейки: и A1, и B1 содержат определенные даты.

=ГОД (A1) -ГОД (B1)

Для извлечения года из каждой заданной даты в этой формуле используется функция ГОД, после чего один год вычитается из другого. Имейте в виду, что, если B1 содержит более близкий к текущей дате год, чем дата в ячейке A1, результат будет отрицательным.

Обратите внимание, данная функция не вычисляет *часть* года. Например, если ячейка A1 содержит дату 31.12.2001, а ячейка B1 дату 01.01.2002, то формула возвратит разницу в один год, несмотря на то, что эти даты отличаются всего лишь на один день.

Определение возраста человека

Возраст человека — это число полных лет, которые человек уже прожил. Формула, приведенная для вычисления количества лет между двумя датами в предыдущем разделе, не сможет корректно определить это значение. Тем не менее, мы можем использовать две другие формулы, позволяющие правильно вычислить возраст человека.

Следующая формула возвращает возраст человека, дата рождения которого содержится в ячейке A1. Эта формула использует функцию ДОЛЯГОДА, которая доступна только при подключении надстройки “Пакет анализа”.

=ЦЕЛОЕ (ДОЛЯГОДА (СЕГОДНЯ () ; A1 ; 1))

Ниже приведена формула, которая уже не использует функции надстройки “Пакет анализа”, а применяет для вычисления возраста функцию РАЗНДАТ (см. врезку “Где найти функцию РАЗНДАТ?”):

=РАЗНДАТ (A1 ; СЕГОДНЯ () ; "y")

Если же необходимо детализировать расчетные данные и вычислить возраст точно, учитывая годы, месяцы и дни, воспользуйтесь следующей формулой:

=РАЗНДАТ (A1 ; СЕГОДНЯ () ; "y") & " лет " &РАЗНДАТ (A1 ; СЕГОДНЯ () ; "ym") & " мес. , "
"&РАЗНДАТ (A1 ; СЕГОДНЯ () ; "md") & " дн. "

Данная формула возвратит текстовую строку: *49 лет, 3 мес., 29 дн.*

Вычисление дня года

Всем понятно, что 1 января — это первый день года, а 31 декабря — последний. Как определить, какой по счету день приходится на произвольно взятую дату в течение года? Следующая ниже формула возвращает порядковый номер дня года для даты, содержащейся в ячейке A1:

=A1-ДАТА (ГОД (A1) ; 1 ; 0)

Иногда порядковый номер дня года может упоминаться как *дата по Юлианскому календарю*.

Данная формула возвращает количество дней в году, оставшихся после наступления определенной даты, заданной в ячейке A1:

=ДАТА (ГОД (A1) ; 12 ; 31) -A1

Где найти функцию РАЗНДАТ?

Наверное, читатель заметил, что мы уже не раз в этой главе упоминали функцию РАЗНДАТ. Обратите внимание, что этой функции нет среди тех, которые доступны в диалоговом окне Мастер функций. Поэтому для того, чтобы использовать эту функцию, ее необходимо ввести вручную.

Функция РАЗНДАТ была известна еще с Lotus 1-2-3 и, очевидно с целью обеспечения совместимости, Excel поддерживает работу этой функции. По определенным причинам Microsoft старается сохранить эту функцию в секрете. Интересно, что в версиях приложения ниже Excel 2000 функция РАЗНДАТ даже не упоминалась. А из интерактивной справки Excel 2002 были удалены все ссылки на эту функцию, хотя сама функция в этой версии приложения все еще доступна.

Функция РАЗНДАТ очень удобна. Она позволяет вычислить количество дней, месяцев и лет, находящихся между двумя датами. Эта функция использует три аргумента: начальная_дата, конечная_дата и определенный код, с помощью которого задается единица времени, интересующая пользователя. Код в функции всегда заключается в кавычки. Ниже приведена таблица, в которой представлены допустимые коды для третьего аргумента.

Код единицы измерения	Возвращаемое значение
"y"	Количество полных лет в заданном периоде.
"m"	Количество полных месяцев в заданном периоде.
"d"	Количество дней в заданном периоде.
"md"	Разница в днях между начальной и конечной датой. Месяцы и годы, в данном случае, игнорируются.
"ym"	Разница в месяцах между начальной и конечной датой. Игнорируются дни и годы.
"yd"	Разница в днях между начальной и конечной датой. Игнорируются годы между заданными датами.

Имейте в виду, что аргумент начальная_дата должен быть более отдален от текущего времени, чем аргумент конечная_дата, в противном случае функция возвращает ошибку.

Обратите внимание, что при вводе любой из этих формул, Excel применяет к ячейке формат даты. Но для того чтобы увидеть полученный результат, необходимо применить к ячейке числовой формат.

Следующую формулу рекомендуется использовать для того, чтобы преобразовать заданный порядковый номер дня года, например, девяностый день, в определенную дату и год, соответствующие этому дню. Для этой формулы предполагается, что содержимое ячейки A1 — это год, а содержимое ячейки B2 — это день года.

=ДАТА (A1 ; 1 ; B1)

Определение дня недели

Для определения дня недели часто применяется функция ДЕНЬНЕД. Эта функция использует в качестве аргумента дату и возвращает целое число, соответствующее дню недели, в диапазоне между 1 и 7. Обратите внимание, что, в данном случае, отсчет дней начинается с

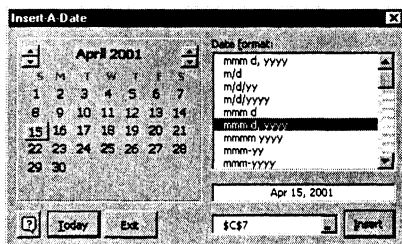
воскресенья, как это принято в англоязычных программных продуктах. Например, следующая формула возвращает значение 3, поскольку первый день 2002 года выпадает на вторник:

```
=ДЕНЬНЕД(ДАТА(2002;1;1))
```

Утилиты для работы с датами из надстройки Power Utility Pak

Специально для работы с датами автором этой книги разработана надстройка Power Utility Pak, которая содержит несколько утилит, предназначенных для работы с датами:

1. Perpetual Calendar (Вечный календарь) — позволяет отображать календарь на любой месяц, а также создавать графическое представление календаря и размещать его на рабочем листе.
2. Insert-A-Date (Вставка даты). Утилита упрощает ввод даты. Чтобы вставить дату в ячейку, достаточно просто щелкнуть на календаре и выбрать один из общих доступных форматов даты.
3. Reminder Alarm (Напоминание). Эта утилита в указанное время дня или после сигнализирует, включая звуковой сигнал, об истечении установленного временного периода.
4. Time Tracker (Таймер) — отслеживает количество потраченного времени и имеет возможность обслуживания до шести проектов.
5. Date Report (Отчет по датам). Утилита создает сообщения обо всех датах, имеющихся в рабочей книге. Эта утилита полезна для определения потенциальной проблемы Y2K.



Функция ДЕНЬНЕД может использовать и второй необязательный аргумент, который определяет систему нумерации дней недели для вывода результата. Если в качестве второго аргумента введено значение 2, то функция возвращает значение 1 для понедельника, 2 для вторника и так далее. Если же в качестве второго аргумента ввести значение 3, то функция возвратит 0 для понедельника, 1 для вторника и далее.



Пользователь может определить день недели даты в ячейке. Для этого достаточно использовать свой собственный числовой формат. Например, если в ячейке установлен приведенный ниже формат, то день недели отобразится одним словом: *вторник*:
DDDD

Определение даты самого ближайшего воскресенья

Хотите узнать, на какую дату приходилось прошедшее воскресенье? Воспользуйтесь формулой, приведенной ниже. Но имейте в виду, если текущая дата выпадает на воскресенье, то формула возвращает текущую дату:

```
=СЕГОДНЯ() - ОСТАТ(СЕГОДНЯ() - 1; 7)
```

Для того чтобы найти дату другого дня, а не воскресенья, измените аргумент 1 в формуле на любой другой в диапазоне между 2 (найти понедельник) и 7 (суббота).

Вычисление дня недели, следующего за определенной датой

А эта формула возвращает день недели, следующий за определенной датой. Например, используйте эту формулу, чтобы определить дату первого понедельника после 1 июня 2001 года. Для этого предположим, что ячейка A1 содержит дату, а ячейка A2 содержит число между 1 и 7 (1 соответствует воскресенью, 2 — вторнику и так далее).

$=A1+A2-ДЕНЬНЕД(A1) + (A2 < ДЕНЬНЕД(A1)) * 7$

Если ячейка A1 содержит дату 1 июня 2001 года, а ячейка A2 содержит значение 2 (понедельник), то формула возвратит дату 4 июня 2001 года. Это будет первый понедельник, следующий после пятницы 1 июня 2001 года.

Определение даты дня недели в заданном месяце

В некоторых случаях у пользователя может возникнуть возможность определить дату, приходящуюся на заданный день недели. Приведем пример. Каждую вторую пятницу каждого месяца в компании день выплаты заработной платы. Как определить дату выплат в каждом месяце? Воспользуйтесь для этого формулой, приведенной ниже:

$=ДАТА(A1; A2; 1) + A3 - ДЕНЬНЕД(ДАТА(A1; A2; 1)) + (A4 - (A3 > ДЕНЬНЕД(ДАТА(A1; A2; 1))) * 7$

Для этой формулы приняты следующие условия:

1. Ячейка A1 содержит значение года.
2. Ячейка A2 содержит значение месяца.
3. Ячейка A3 содержит значение дня (1 для воскресенья, 2 для понедельника и так далее).
4. Ячейка A4 содержит расположение буднего дня, заданного в ячейке A3, в пределах месяца.

Например, если использовать эту формулу для определения даты первой пятницы в июне 2002 года, формула возвратит дату 7 июня 2002 года.



Иногда значение, содержащееся в ячейке A4, может выпасть за пределы количества дней в указанном месяце. В таких случаях, формула возвращает дату, относящуюся к следующему месяцу. Например, если задать определение даты пятой пятницы июня 2002 года (а такой пятницы не существует), то формула возвращает дату первой пятницы июля.

Определение числа заданных дней недели в месяце

Формула, приведенная ниже, позволяет вычислить количество определенных дней недели, выпадающих в течение заданного месяца. Предположим, что ячейка A1 содержит дату, а ячейка A2 порядковый номер дня недели (значение 1 соответствует воскресенью, 2 — поне-

дельнику и так далее). Имейте в виду, что эта формула — формула массива, поэтому чтобы ввести ее, используйте клавиши <Ctrl+Shift+Enter>.

```
{=СУММ( ( ДЕНЬНЕД ( ДАТА ( ГОД ( А1 ) ; МЕСЯЦ ( А1 ) ; СТРОКА ( ДВССЫЛ ( " 1 : " & ДЕНЬ ( ДАТА ( ГОД ( А1 ) ; МЕСЯЦ ( А1 ) + 1 ; 0 ) ) ) ) ) ) ) = В1 ) * 1 ) }
```

Если ячейка А1 содержит дату 5 января 2002 года, а ячейка В1 — значение 3, то есть вторник, формула возвращает значение 5. Это значит, что январь 2002 года содержит пять вторников.

Приведенная выше формула вычисляет год и месяц, используя функции ГОД и МЕСЯЦ. Однако она может быть несколько упрощена. Для этого введите год и месяц заданной даты в отдельные ячейки. Для следующей ниже формулы, которая также является формулой массива, мы приняли условия, что ячейка А1 содержит значение года, ячейка А2 — месяца и ячейка В2 — значение дня недели:

```
{=СУММ( ( ДЕНЬНЕД ( ДАТА ( А1 ; А2 ; СТРОКА ( ДВССЫЛ ( " 1 : " & ДЕНЬ ( ДАТА ( А1 ; А2 + 1 ; 0 ) ) ) ) ) ) = В1 ) * 1 ) }
```

Обратите внимание, на рис. 6.5 отображено применение этой формулы на рабочем листе. В этом случае используются и абсолютные, и смешанные ссылки на ячейки, поэтому пользователь может скопировать эту формулу. К примеру, формула в ячейке С3 выглядит следующим образом:

```
{=СУММ( ( ДЕНЬНЕД ( ДАТА ( $B$2 ; $A3 ; СТРОКА ( ДВССЫЛ ( " 1 : " & ДЕНЬ ( ДАТА ( $B$2 ; $A3 + 1 ; 0 ) ) ) ) ) ) = C$1 ) * 1 ) }
```

Итоговые формулы на этом листе используют функцию СУММ. Они вычисляют общее количество дней в месяце (столбец J) и количество каждого дня недели в году (строка 15).

YEAR ->	2002	Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	January	4	4	5	5	5	4	4
2	February	4	4	4	4	4	4	4
3	March	5	4	4	4	4	5	5
4	April	4	5	5	4	4	4	4
5	May	4	4	4	5	5	5	4
6	June	5	4	4	4	4	4	5
7	July	4	5	5	5	4	4	4
8	August	4	4	4	4	5	5	5
9	September	5	5	4	4	4	4	4
10	October	4	4	5	5	5	4	4
11	November	4	4	4	4	4	5	5
12	December	5	5	5	4	4	4	4
15	Total Days:	52	52	53	52	52	52	52

Рис. 6.5. На этом рабочем листе вычисляется число каждого дня недели в году

Вычисление праздничных дат

Определение дат, выпадающих на специфические праздничные дни, требует использования некоторых хитростей. Безусловно, это не касается таких дат, как Новый год или День Независимости США, поскольку это уже сами по себе даты, определенные издавна. Для них можно просто использовать функцию ДАТА, о которой мы уже говорили ранее в этой главе. Следующая формула отображает дату Нового года. Новый год всегда приходится на 1 января; требуемый год введите в ячейку А1:

```
=ДАТА ( А1 ; 1 ; 1 )
```

На самом деле просто определяются только праздники, относящиеся к определенным датам. А как быть с праздничными днями, дату которых определяет просто временной период. Например, определенный день недели одного из месяцев года. К таким праздникам относится День труда, который отмечается в первый понедельник сентября.

Обратите внимание, на рис. 6.6 отображена рабочая книга Excel, где вычисляются даты десяти праздничных дней США. Все формулы содержат ссылку на ячейку A1 — год, для которого вычисляются даты. Заметьте, что несмотря на то, что такие праздники, как Новый год, День Независимости и День Ветеранов, имеют четко установленные даты, функция ДАТА все же вычисляет их даты по заданному порядковому номеру дня определенного месяца.

Holiday	Description	Date	Weekday
New Year's Day	1st Day in January	Январь 1, 2002	вторник
Martin Luther King Jr. Day	3rd Monday in January	Январь 21, 2002	понедельник
Presidents' Day	3rd Monday in February	Февраль 18, 2002	понедельник
Memorial Day	Last Monday in May	Май 27, 2002	понедельник
Independence Day	4th Day of July	Июль 4, 2002	четверг
Labor Day	1st Monday in September	Сентябрь 2, 2002	понедельник
Veterans Day	11th Day of November	Ноябрь 11, 2002	понедельник
Columbus Day	2nd Monday in October	Октябрь 14, 2002	понедельник
Thanksgiving Day	4th Thursday in November	Ноябрь 28, 2002	четверг
Christmas Day	25th Day of December	Декабрь 25, 2002	среда

Рис. 6.6. Для того чтобы определить дату праздничного дня, воспользуйтесь специальными формулами

День Мартина Лютера Кинга

Этот праздничный день приходится на третий понедельник января. Давайте определим, на какую дату выпадет этот день в 2002 году, определенном в ячейке A1:

=ДАТА (A1 ; 1 ; 1) +ЕСЛИ (2<ДЕНЬНЕД (ДАТА (A1 ; 1 ; 1)) ; 7-
ДЕНЬНЕД (ДАТА (A1 ; 1 ; 1)) +2 ; 2-ДЕНЬНЕД (ДАТА (A1 ; 1 ; 1)) + ((3-1) *7)

День Президентов

День Президентов празднуется в третий понедельник февраля. Следующая формула вычисляет дату этого дня в 2002 году (ячейка A1):

=ДАТА (A1 ; 2 ; 1) +ЕСЛИ (2<ДЕНЬНЕД (ДАТА (A1 ; 2 ; 1)) ; 7-
ДЕНЬНЕД (ДАТА (A1 ; 2 ; 1)) +2 ; 2-ДЕНЬНЕД (ДАТА (A1 ; 2 ; 1)) + ((3-1) *7)

День памяти павших

День памяти павших в гражданской войне в США 1861–65 гг., в испано-американской и других войнах отмечается в США в последний понедельник мая. На какую дату он придется в 2002 году, (ячейка A1) определяется с помощью следующей формулы:

=ДАТА (A1 ; 6 ; 1) +ЕСЛИ (2<ДЕНЬНЕД (ДАТА (A1 ; 6 ; 1)) ; 7-
ДЕНЬНЕД (ДАТА (A1 ; 6 ; 1)) +2 ; 2-ДЕНЬНЕД (ДАТА (A1 ; 6 ; 1)) + ((1-1) *7) -7

Обратите внимание, что данная формула фактически вычисляет первый понедельник июня, а затем, чтобы определить предыдущий понедельник мая, из полученного результата вычитает 7 дней.

День Труда

День Труда — это первый понедельник сентября. Формула, приведенная ниже, определяет его дату в 2002 году (ячейка A1):

```
=ДАТА (A1 ; 9 ; 1) +ЕСЛИ (2<ДЕНЬНЕД (ДАТА (A1 ; 9 ; 1)) ; 7 -  
ДЕНЬНЕД (ДАТА (A1 ; 9 ; 1)) +2 ; 2-ДЕНЬНЕД (ДАТА (A1 ; 9 ; 1)) + ((1-1) *7)
```

День Колумба

Этот праздник выпадает на второй понедельник октября. Для вычисления его даты в 2002 году (ячейка A1) используется следующая формула:

```
=ДАТА (A1 ; 10 ; 1) +ЕСЛИ (2<ДЕНЬНЕД (ДАТА (A1 ; 10 ; 1)) ; 7 -  
ДЕНЬНЕД (ДАТА (A1 ; 10 ; 1)) +2 ; 2-ДЕНЬНЕД (ДАТА (A1 ; 10 ; 1)) + ((2-1) *7)
```

День Благодарения

День Благодарения празднуется в четвертый четверг ноября. Следующая формула вычисляет дату этого дня в 2002 году (ячейка A1):

```
=ДАТА (A1 ; 11 ; 1) +ЕСЛИ (5<ДЕНЬНЕД (ДАТА (A1 ; 11 ; 1)) ; 7 -  
ДЕНЬНЕД (ДАТА (A1 ; 11 ; 1)) +5 ; 5-ДЕНЬНЕД (ДАТА (A1 ; 11 ; 1)) + ((4-1) *7)
```

Определение дня Пасхи

Наверное, читатель уже обратил внимание, что Пасхальный день не рассматривается в предыдущем разделе. Дело в том, что Пасха — это необычный праздник, поскольку его дата определяется по лунному, а не по обычному календарю. Из-за этого, определение даты, на которую выпадает Пасха, становится несколько проблематичным.

Однажды один из пользователей Excel, Ганс Гербер (Hans Herber) из Германии, на своем Web-узле организовал своеобразное соревнование по созданию формулы вычисления даты Пасхи. Цель этого состязания состояла в том, чтобы создать самую, насколько это возможно, короткую формулу, которая бы правильно определяла дату Пасхи с 1900 до 2078 года.

На рассмотрение было представлено двадцать формул различной длины: от 44 до 154 символов. Среди этих формул есть такие, которые работают только с европейскими параметрами настройки даты. Ниже приведена самая короткая из всех представленных формул. Ее автор Томас Джансен (Thomas Jansen). Данная формула работает также с любыми параметрами настройки даты.

```
=РУБЛЬ ("4/"&A1) /7+ОСТАТ (19* ОСТАТ (A1 ; 19) -7 ; 30) *14% ; ) *7-6
```

Формула возвращает дату, на которую выпадает день Пасхи в указанном году (ячейка A1).

И, пожалуйста, не просите объяснить эту формулу. У меня нет ни единого предположения по поводу того, как она работает!

Определение даты последнего дня месяца

Для того чтобы определить дату, соответствующую последнему дню в месяце, можно воспользоваться функцией ДАТА. При этом число месяцев необходимо увеличить на единицу, а для дня использовать значение 0. Другими словами, в расчет принимается условие, что нулевой день следующего месяца — это последний день текущего месяца.

Приведенная ниже формула возвращает дату, которая соответствует последнему дню месяца. Данная формула предполагает, что дата содержится в ячейке A1.

=ДАТА (ГОД (A1) ; МЕСЯЦ (A1) +1 ; 0)

Эта формула может быть модифицирована, после чего она дает возможность вычислять количество дней, содержащихся в определенных месяцах. Следующая формула возвращает целое числовое значение, которое представляет собой количество дней в месяце, заданном в ячейке A1:

=ДЕНЬ (ДАТА (ГОД (A1) ; МЕСЯЦ (A1) +1 ; 0))

Определение високосного года?

Безусловно, для того чтобы определить, является ли високосным определенный год, можно написать формулу, которая определяет дату двадцать девятого дня февраля: 29 февраля или 1 марта. Но гораздо эффективнее воспользоваться формулой ДАТА, которая еще и откорректирует результат в том случае, если введен неправильный аргумент. Например, значение 29 для дня, в то время как в феврале этого года всего 28 дней.

Эта формула возвращает значение ИСТИНА, если дата, находящаяся в ячейке A1, относится к високосному году. В противном случае формула возвращает значение ЛОЖЬ:

=ЕСЛИ (МЕСЯЦ (ДАТА (ГОД (A1) ; 2 ; 29)) =2 ; ИСТИНА ; ЛОЖЬ)



Помните, что для 1900 года результат, возвращаемый данной функцией — ИСТИНА, будет неправильным. Для получения дополнительной информации см. раздел "Ошибка високосного года" ранее в этой главе.

Вычисление квартального периода

Наверное, всем знаком такой термин как квартальный отчет. Для многих финансовых работников квартальное отображение информации очень полезно. Формула, приведенная ниже, возвращает числовое значение от 1 до 4, которое соответствует календарному кварталу для даты, содержащейся в ячейке A1:

=ОКРУГЛВВЕРХ (МЕСЯЦ (A1) / 3 ; 0)

Данная формула делит количество месяцев на три, а затем округляет полученный результат.

Отображение года римскими цифрами

Любителям старых кинофильмов наверняка понравится эта особенность Excel. Следующая формула отображает 1945 год римскими цифрами и возвращает MCMXLV.

=РИМСКОЕ (1945)

Функция РИМСКОЕ работает только тогда, когда на компьютере установлена надстройка "Пакет анализа". Но имейте в виду, что данная функция возвращает текстовую строку, поэтому ее результат нельзя использовать для проведения вычислений! Кроме того, к сожалению, Excel не имеет обратной функции, которая позволила бы преобразовать римские цифры в арабские.

Создание календаря в отдельном диапазоне ячеек

Для создания календаря, который приводится здесь в качестве примера (рис. 6.7), использована одна формула (формула массива), позволяющая отображать календарь в определенном диапазоне ячеек. Полосы прокрутки связаны с ячейками, которые указывают месяц и год. В данном случае ячейка B2 содержит месяц (назовем его m), а ячейка D2 содержит год (названный y). Далее введите следующую формулу массива в диапазон ячеек B6:H11:

```
{=ЕСЛИ(МЕСЯЦ(ДАТА(y;m;1))<>МЕСЯЦ(ДАТА(y;m;1)-(ДЕНЬНЕД(ДАТА(y;m;1))-1)+{0;1;2;3;4;5}*7+{1;2;3;4;5;6;7}-1);"";ДАТА(y;m;1)-(ДЕНЬНЕД(ДАТА(y;m;1))-1)+{0;1;2;3;4;5}*7+{1;2;3;4;5;6;7}-1)}
```

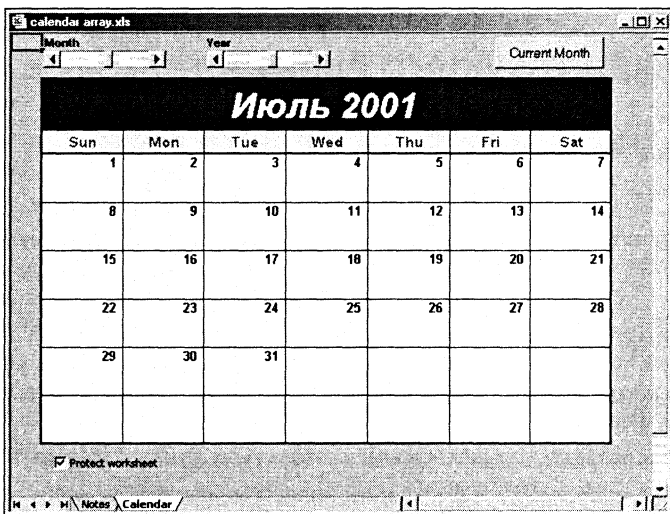


Рис. 6.7. Чтобы создать этот календарь, используйте одну формулу массива, поместив ее в 42 ячейки

Функции управления временем

В этом разделе мы будем рассматривать использование некоторых очень полезных функций управления временем. Как мы уже говорили, Excel позволяет пользователю работать с различными временными характеристиками, поэтому она содержит целый ряд функций, предназначенных для управления информацией о времени.

Давайте сразу систематизируем связанные со временем доступные в Excel функции (табл. 6.6). Все приведенные ниже функции можно найти в диалоговом окне Мастер функций в категории Дата и время.

Таблица 6.6. Функции, связанные со временем

Функция	Описание функции
ЧАС	Преобразует дату в числовом формате в часы
МИНУТЫ	Преобразует дату в числовом формате в минуты
МЕСЯЦ	Возвращает номер месяца (число в диапазоне от 1 до 12), соответствующий порядковому номеру даты в числовом формате
ТДАТА	Возвращает текущую дату и время в числовом формате
СЕКУНДЫ	Преобразует даты в числовом формате в секунды
ВРЕМЯ	Возвращает дату в числовом формате для заданного времени
ВРЕМЗНАЧ	Преобразует время из текстового формата в дату в числовом формате

Отображение текущего времени

Приведенная ниже формула возвращает текущее время как время в числовом формате или как порядковый номер времени безотносительно определенной даты:

`=ТДАТА() -СЕГОДНЯ()`



Для того чтобы ввести в заданную ячейку постоянную временную отметку, нажмите комбинацию клавиш <Ctrl+Shift+>.

Также как и для даты, для времени в Excel существует достаточно широкий выбор форматов. Например, вы можете отображать на экране время до и после полудня, используя следующий числовой формат:

чч:мм AM/PM

Время может объединяться с текстовой строкой. Следующая формула позволяет отобразить текст *“Текущее время 6:28 PM”*:

`=“Текущее время ” &ТЕКСТ(ТДАТА()); “ч:мм AM/PM”`



Имейте в виду, что эти формулы обновляются при пересчете рабочего листа.

Отображение времени

Ранее в этой главе, мы уже говорили о том, как вводить временные данные. Достаточно просто ввести в ячейку определенное время, используя для этого, по крайней мере, один знак : (двоеточие). Но есть и другой способ. Чтобы отобразить время, воспользуйтесь функцией ВРЕМЯ. Например, формула приведенная ниже, возвращает час, который содержит ячейка A1, минуты, введенные в ячейку B1, и секунды из ячейки C1:

`=ВРЕМЯ(A1; B1; C1)`

Так же как и для функции ДАТА, для функции ВРЕМЯ могут вводиться недопустимые аргументы, которые она затем приводит в соответствие самостоятельно. Например, давайте введем в следующую формулу заведомо неправильный аргумент 80. В этом случае формула

возвратит значение 10:20:15 AM. Неверные 80 минут будут просто добавлены к заданному часу, а оставшаяся их часть будет отображена как 20 минут.

=ВРЕМЯ(9;80;15)



Если ввести в качестве первого аргумента функции ВРЕМЯ значение, превышающее 24 часа, то результат будет несколько иным, чем можно было ожидать. Следуя логике, можно предположить, что такая формула должна вычислять числовое значение даты и времени равное 1,041667 (то есть один день и один час):

=ВРЕМЯ(25;0;0)

Но, фактически, эта формула имеет тот же смысл, что и следующая формула:

=ВРЕМЯ(1;0;0)

Тем не менее, функция ДАТА может быть использована вместе с функцией ВРЕМЯ в одной ячейке. Например, следующая формула вычисляет числовое значение 37 229,7708333333, которое соответствует 6:30 PM 4 декабря 2001 года:

=ДАТА(2001;12;4)+ВРЕМЯ(18;30;0)

Следующая функция — ВРЕМЗНАЧ преобразует текстовую строку, похожую на временное значение, в числовое представление заданного времени. Приведенная ниже формула возвращает числовое значение 0,2395833333 для времени 5:45:

=ВРЕМЗНАЧ("5:45")

Для того чтобы результат этой формулы отображался как время, необходимо применить к ячейке формат времени. Имейте в виду, что функция ВРЕМЗНАЧ принимает только форматы времени, поддерживаемые в приложении Excel. Например, следующая формула возвратит значение ошибки, поскольку Excel не воспринимает запись "a.m." как параметр времени:

=ВРЕМЗНАЧ("5:45 a.m.")

Суммирование времени, превышающего 24 часа

Давайте попробуем суммировать несколько значений времени, превышающих 24 часа. Вы, наверное, удивитесь, увидев, что Excel не может отобразить правильную сумму всех значений. Возьмем, например, диапазон ячеек B2:B8 (рис. 6.8), в которых введено несколько значений времени, число отработанных часов и минут за каждый день недели. Формула, вычисляющая суммарное значение времени, содержится в ячейке B9:

=СУММ(B2:B8)

Весьма очевидно, что формула возвращает неправильное значение (18 часов 30 минут). Ведь простой расчет показывает, что общее количество часов должно быть равным 42 часам 30 минутам. Проблема состоит в том, что, на самом деле, формула возвращает числовое значение даты и времени 1,770833, но формат ячейки не позволяет отображать часть этого значения, относящегося к дате.

Для того чтобы полученный результат отображался как время, превышающее 24 часа, необходимо изменить формат результирующей ячейки, а часть времени, представляющую со-

Day	Hours	Minutes
Sunday	0	
Monday	8:30	
Tuesday	8:00	
Wednesday	9:00	
Thursday	9:30	
Friday	4:30	
Saturday	3:00	
Total for Week:	18:30	

Рис. 6.8. Функция СУММ вычисляет общее количество отработанных часов. Однако данный ответ неверен, поскольку ячейка B9 имеет неправильный формат данных

бой *часы*, заключить в квадратные скобки. Применение данного формата к ячейке B9 позволит правильно отобразить сумму значений времени:

[ч] : мм

Рассмотрим другой пример расчета времени. На следующем рабочем листе (рис. 6.9) отслеживаются нормативные и сверхурочные часы, отработанные в течение недели.

Дата начала работы введена в ячейку D5. Столбец В содержит формулы, которые вычисляют дни недели, соответствующие каждой дате. Таким образом, определяются рабочие дни недели. Отработанное время вводится в диапазон ячеек D8 : G14. Следующий столбец — Н, содержит формулы, которые вычисляют общее количество часов, отработанное за каждый день. Например, формула, содержащаяся в ячейке H8, выглядит следующим образом:

=ЕСЛИ (E8<D8 ; E8+1-D8 ; E8-D8) +ЕСЛИ (G8<F8 ; G8+1-F8 ; G8-F8)

Weekday	Date	Start Work	Time Out (Lunch)	Time In (Lunch)	End Work	Total Hours	Weekly Hours	
понедельник	04.03.2002	8:00 AM	12:00 PM	1:00 PM	6:00 PM	9:00	9:00	
вторник	05.03.2002	10:00 AM	2:00 PM	2:30 PM	7:00 PM	8:30	17:30	
среда	06.03.2002	9:00 AM	12:00 PM	1:00 PM	6:30 PM	8:30	26:00	
четверг	07.03.2002	7:30 AM	11:30 AM	12:00 PM	6:30 PM	10:30	36:30	
пятница	08.03.2002	9:30 AM	2:00 PM	3:30 PM	5:00 PM	6:00	42:30	
суббота	09.03.2002					0:00	42:30	
воскресенье	10.03.2002					0:00	42:30	
WEEKLY TOTAL								
Total hours:							42:30	
Regular hours:							40:00	
Overtime hours:							2:30	

Рис. 6.9. Пример рабочей книги для расчета времени, отработанного служащими

В первой части данной формулы определяется количество часов, отработанных служащими до обеда. Время, содержащееся в столбце D, вычитается из времени в столбце E. Вторая часть формулы вычисляет часы, отработанные после обеда: столбец F вычитается из столбца G. Функция ЕСЛИ используется для того, чтобы учесть в расчете различные сдвиги во времени начала смены. Например, один из служащих может начинать работу в 10:00 PM, соответственно с перерывом на обед в 2:00 AM. Имейте в виду, что без функции ЕСЛИ формула может возвращать отрицательный результат.

Следующая формула, введенная в ячейку H17, подбивает итог по столбцу Н и вычисляет общее количество часов, отработанных в течение недели.

=СУММ (H8 : H14)

В этом примере принята 40-часовая рабочая неделя. Поэтому все часы, отработанные сверх 40 часов, рассматриваются как сверхурочные часы. Ячейка, отображающая *сверхурочное время*, не отображена на рис. 6.9. Эта ячейка вычисляет установленное нормами количество рабочих часов в неделю, используя следующую формулу:

=1+ВРЕМЯ (16 ; 0 ; 0)

Данная формула возвращает значение 40:00, то есть 24 часа плюс 16 часов. Однако, если стандартная рабочая неделя в расчетах пользователя содержит более или менее 40 часов, эта формула может быть изменена в соответствии с частными условиями.

Следующая ячейка (E18) содержит формулу, которая вычисляет нормативные часы. Формула возвращает меньшее из двух значений, введенных в качестве аргументов: общее, установленное стандартами, количество часов или сверхурочные часы.

=МИН (E17;Overtime)

И, наконец, последняя формула, содержащаяся в ячейке E19, вычитает нормативные часы из общего количества отработанных часов и возвращает количество сверхурочных часов:

=E17-E18

Все значения времени в диапазоне ячеек E17:E19 отображают время, превышающее 24 часа, поскольку используют собственный числовой формат:

[ч]:мм

Определение разницы между двумя временными значениями

Поскольку Excel позволяет преобразовать значение времени в числовой формат, мы можем легко вычитать более раннее время из более позднего, определяя таким образом разницу во времени. Например, если ячейка A2 содержит значение времени 5:30:00, а ячейка B2 — 14:00:00, то следующая формула возвратит значение времени 08:30:00, то есть разница во времени составляет 8 часов 30 минут:

=B2-A2

Если же разница во времени отрицательная, что является недопустимым результатом, Excel отображает несколько знаков “решетки” (#####), поскольку при отсутствии даты точка отсчета времени — это 0. Отрицательное время, соответственно, означает отрицательное числовое значение, что само по себе уже недопустимо.

В том случае, если направление вычитания времени не имеет для пользователя никакого значения, можно использовать функцию ABS, которая возвращает абсолютное значение:

=ABS (B2-A2)

Проблема “отрицательного времени” часто возникает при вычислении разницы между более ранним и более поздним временем. Вернемся к задаче вычисления количества отработанных часов. Чтобы определить его, достаточно просто вычесть начальное время работы из конечного времени. Это не представляет никакой проблемы, если два этих значения приходятся на один и тот же день. Но как быть, если рабочее время начинается незадолго до конца текущих суток, а заканчивается на следующие сутки. Например, служащий начинает работать в 10:00 PM, а заканчивает в 6:00 AM. В этом случае, результатом вычитания будет отрицательное время. Таким образом, определить отработанные часы для смены, начинающейся в одних сутках, а заканчивающейся в других, становится проблемой (рис. 6.10).

	A	B	C	D
1	Start Shift	End Shift	Hours Worked	
2	8:00 AM	5:30 PM	9:30	
3	10:00 PM	6:00 AM	#####	
4				
5				
6				
7				

Рис. 6.10. При определении количества отработанных часов при смене, продолжающейся за полночь, Excel возвращает ошибочное значение

Какой может быть выход из этой ситуации? Использование функции ABS и вычисление абсолютного значения в этом случае не подходит, поскольку формула возвращает неправильный результат — 16 часов. Тем не менее, приведенная ниже формула может сработать:

$= (B2 + (B2 < A2) - A2)$

Можно использовать другую, более простую формулу:

$= \text{ОСТАТ}(B2 - A2; 1)$



Как мы уже говорили, работа с отрицательным временем возможна, если рабочая книга использует систему дат 1904. Чтобы перейти к этой системе дат, выберите команду Сервис⇒Параметры и щелкните на вкладке Вычисления. Установите флажок опции система дат 1904. Будьте внимательны! При изменении системы дат, все даты, имеющиеся в рабочей книге, будут увеличены на четыре года.

Преобразование “военного времени”

Военное время, как правило, представляет собой четырехразрядное значение в пределах от 0000 до 2359. Например, 1:00 AM отображается как 0100 часов, а 3:30 PM как 1530 часов. Формула, приведенная ниже, преобразует это значение (ячейка A1) в общепринятый формат времени:

$= \text{ВРЕМЗНАЧ}(\text{ЛЕВСИМВ}(A1; 2) \& " " \& \text{ПРАВСИМВ}(A1; 2))$

Если ячейка A1, содержит больше или меньше четырех чисел, то результат, который возвратит данная формула, будет неверным. Следующая формула корректирует эту ошибку и возвращает допустимое значение времени для любого времени в военном формате от 0 до 2359:

$= \text{ВРЕМЗНАЧ}(\text{ЛЕВСИМВ}(\text{ТЕКСТ}(A9; "0000"); 2) \& " " \& \text{ПРАВСИМВ}(A9; 2))$

Следующая, более простая формула, использует функцию ТЕКСТ, которая форматирует строку как текст, а затем использует функцию ВРЕМЗНАЧ, чтобы отобразить результат в формате времени:

$= \text{ВРЕМЗНАЧ}(\text{ТЕКСТ}(A9; "00 \ : 00"))$

Преобразование часов, минут, секунд в десятичном исчислении

Иногда часы, минуты или секунды времени представляются в виде значений в десятичном исчислении. Для того чтобы преобразовать их в стандартный формат времени, просто разделите значение на 24. Предположим, что ячейка A1 содержит значение часа 9,25. Следующая формула возвращает 09 : 15 : 00 (9 часов 15 минут):

$= A1 / 24$

Подобная формула может использоваться и для преобразования минут в десятичном исчислении. Но в знаменателе должно быть число 1 440 — количество минут в сутках. Например, если ячейка A1 содержит значение 500 (минут), то следующая формула возвращает 08:20:00 (8 часов 20 минут):

=A1/1440

Для преобразования секунд в стандартный формат времени десятичные секунды делятся на 86 400 (число секунд в сутках). Например, если ячейка A1 содержит число 65 000 (секунд), то формула, приведенная ниже, возвращает 18:03:20 (18 часов 3 минуты и 20 секунд):

=A1/86400

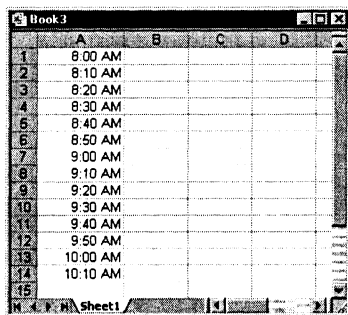
Добавление часов, минут или секунд к заданному времени

Функция ВРЕМЯ может также использоваться для добавления часов, минут или секунд к времени, содержащемуся в какой-либо ячейке. Например, если ячейка A1 уже содержит время, следующая формула позволяет добавить к нему 2 часа и 30 минут и отобразить полученный результат:

=A1+ВРЕМЯ(2;30;0)

Эта же функция используется для заполнения определенного диапазона ячеек возрастающими значениями времени. Обратите внимание, на рис. 6.11 отображен рабочий лист, в котором диапазон ячеек A1:A14 содержит значения времени, каждое из которых последовательно увеличивается на 10 минут. Только в одну из этих ячеек (A1) значение времени вводится вручную. В остальные ячейки просто копируется формула, введенная в ячейку A2:

=A1+ВРЕМЯ(0;10;0)



	A	B	C	D
1	8:00 AM			
2	8:10 AM			
3	8:20 AM			
4	8:30 AM			
5	8:40 AM			
6	8:50 AM			
7	9:00 AM			
8	9:10 AM			
9	9:20 AM			
10	9:30 AM			
11	9:40 AM			
12	9:50 AM			
13	10:00 AM			
14	10:10 AM			
15				

Рис. 6.11. Для заполнения ячеек несколькими временными значениями, последовательно возрастающими на 10 минут, использована всего одна формула

Работа с временными поясами

Представьте, что вам необходимо использовать рабочий лист, который содержит дату и время по Гринвичу (Greenwich Mean Time — GMT), которое иногда еще называется универсальным временем. Как перейти от этого времени ко времени текущего часового пояса? Для того чтобы привести дату и время в соответствие определенному часовому поясу, необходимо, в первую очередь, установить разницу между этими временными поясами. К примеру, разница между временем по Гринвичу и Стандартным центральноамериканским временем США (U.S. Central Standard Time) составляет –6 часов.

В данном случае, функцию ВРЕМЯ с отрицательным аргументом использовать нельзя, потому эта задача требует несколько иного подхода. Один час эквивалентен 1/24 суток. Таким образом, разницу во времени можно также разделить на 24, а затем полученный результат добавить ко времени исходного часового пояса.

	A	B	C
1	Conversion Factor:		5 hours
2			
3			
4		GMT	Local Time
5	01/05/2002 01:00 AM		01/04/2002 08:00 PM
6	01/05/2002 03:30 AM		01/04/2002 10:30 PM
7	01/05/2002 06:00 AM		01/05/2002 01:00 AM
8	01/05/2002 08:30 AM		01/05/2002 03:30 AM
9	01/05/2002 11:00 AM		01/05/2002 06:00 AM
10	01/05/2002 01:30 PM		01/05/2002 08:30 AM
11	01/05/2002 04:00 PM		01/05/2002 11:00 AM
12	01/05/2002 06:30 PM		01/05/2002 01:30 PM
13	01/05/2002 09:00 PM		01/05/2002 04:00 PM
14	01/05/2002 11:30 PM		01/05/2002 06:30 PM
15	01/06/2002 02:00 AM		01/05/2002 09:00 PM
16	01/06/2002 04:30 AM		01/05/2002 11:30 PM
17	01/06/2002 07:00 AM		01/06/2002 02:00 AM
18	01/06/2002 09:30 AM		01/06/2002 04:30 AM
19	01/06/2002 12:00 PM		01/06/2002 07:00 AM
20			
21			

Рис. 6.12. На этом рабочем листе дата и время переводится из одного часового пояса в другой

Обратите внимание, на рис. 6.12 приведен пример рабочего листа, в котором дата и время по Гринвичу приводятся в соответствие текущему часовому поясу. Ячейка B1 содержит разницу в часах между поясами. В данном случае –5 часов для Стандартного восточного времени США (U.S. East Standard Time). Ячейка B4 содержит следующую формулу, которая затем копируется вниз по столбцу:

$$=A4 + (\$B\$1 / 24)$$

Эта формула весьма эффективна, поскольку добавляет x часов к дате и времени, содержащимся в столбце A. Если ячейка B1 содержит отрицательное значение часа, это значение вычитается из даты и времени, содержащихся в столбце A. Заметьте, что, в некоторых случаях, это действие также изменяет и дату.

Округление времени

Иногда у пользователя может возникнуть необходимость округлить время до определенного значения. Например, округлить записи в отчете об использовании времени до ближайшего значения 15 минут. В этом разделе мы обсудим несколько возможных решений задачи округления временных значений.

Следующая формула округляет время, содержащееся в ячейке A1 до ближайшей минуты:

$$=ОКРУГЛ(A1 * 1440 ; 0) / 1440$$

Эта формула работает следующим образом: заданное время умножается на 1 440, чтобы вычислить общее количество минут. После этого, полученный результат обрабатывается с помощью функции ОКРУГЛ и делится на 1 440. Например, если ячейка A1 содержит 11:52:34, формула возвращает 11:53:00.

Следующая формула несколько похожа на предыдущую, за исключением того, что она округляет время в ячейке A1 до ближайшего часа:

$$=ОКРУГЛ(A1 * 24 ; 0) / 24$$

Если ячейка A1 содержит 5:21:31, данная формула возвращает 5:00:00.

Формула, приведенная ниже, округляет время, содержащееся в ячейке A1 до ближайших 15 минут:

$$=ОКРУГЛ(A1 * 24 / 0,25 ; 0) * (0,25 / 24)$$

В этой формуле значение 0,25 означает часть часа, то есть 15 минут. Аналогично, чтобы округлить время до ближайших 30 минут, просто измените 0,25 на 0,5, как это сделано в следующей формуле:

$$=ОКРУГЛ(A1 * 24 / 0,5 ; 0) * (0,5 / 24)$$

Работа со значениями, не имеющими отношения ко времени суток

Вы когда-нибудь сталкивались с задачей составления списка результатов гоночных соревнований или, например, подсчета времени ежедневной пробежки? Эти временные значения, фактически, никак не связаны с определенным временем дня, а, скорее, характеризуются длительностью определенного события в часах, минутах, секундах. Как работать с этими значениями? Предположим, например, что время, отведенное на тест, составляет 35 минут 40 секунд. Это значение может быть введено в ячейку следующим образом:

00:35:45

Excel воспринимает эту запись достаточно хорошо и интерпретирует ее как 12:35:45 AM. Остается только установить для этой ячейки подходящий формат времени. Однако имейте в виду, что при вводе времени, которое не содержит часа, для значения часа необходимо указать хотя бы один ноль. В противном случае, если данные часа будут опущены, Excel интерпретирует эту запись как 35 часов 45 минут.

Обратите внимание, на рис. 6.13 приведен пример рабочего листа, в котором отслеживаются тренировки по бегу спортсмена. Столбец A содержит обыкновенные даты. Столбец B — расстояние в милях. А столбец C содержит время, которое потребовалось на преодоление каждой соответствующей дистанции. Формулы, содержащиеся в столбце D, вычисляют скорость спортсмена в милях за час. Например, формула в ячейке D2 выглядит следующим образом:

=B2 / (C2 * 24)

	A	B	C	D	E	F	G	H
1	Date	Distance	Time	Speed (mph)	Pace (min/mile)	YTD Distance	Cumulative Time	
2	01/01/2002	1.50	00:18:45	4.80	12.50	1.50	00:18:45	
3	02/01/2002	1.50	00:17:40	5.09	11.78	3.00	00:36:25	
4	03/01/2002	2.00	00:21:30	5.58	10.75	5.00	00:57:55	
5	04/01/2002	1.50	00:15:20	5.87	10.22	6.50	01:13:15	
6	05/01/2002	2.40	00:25:05	5.74	10.45	8.90	01:38:20	
7	06/01/2002	3.00	00:31:06	5.79	10.37	11.90	02:09:26	
8	07/01/2002	3.80	00:41:06	5.55	10.82	15.70	02:50:32	
9	08/01/2002	5.00	01:09:00	4.35	13.80	20.70	03:59:32	
10	09/01/2002	4.00	00:45:10	5.31	11.29	24.70	04:44:42	
11	10/01/2002	3.00	00:29:06	6.19	9.70	27.70	05:13:48	
12	11/01/2002	5.50	01:06:30	4.82	12.45	33.20	06:22:18	
13								
14								
15								
16								

Рис. 6.13. На этом рабочем листе приведены временные значения, не связанные с определенным временем суток

В столбце E содержатся формулы, которые определяют темп спортсмена, в минутах за милю. Например, формула в ячейке E2:

= (C2 * 60 * 24) / B2

Ячейки в столбцах F и G содержат формулы, которые вычисляют кумулятивные итоги пройденного расстояния (данные взяты из столбца B) и использованного времени (данные взяты из столбца C). Все ячейки столбца G используют числовой формат, который позволяет отображать время, превышающее 24 часа:

[чч]:мм:сс

Резюме

В этой главе мы рассмотрели возможности приложения Excel, позволяющие работать с данными даты и времени. В общих чертах мы обсудили существующие системы дат и работу с последовательными датами и временем. В данной главе приведено множество примеров формул, использующихся для работы с датой и временем.

В следующей главе мы рассмотрим различные возможные методы расчета данных в электронной таблице.

Способы суммирования и подсчета данных

В этой главе...

- ◆ Подсчет и суммирование ячеек электронной таблицы
- ◆ Подсчет или суммирование записей в базах данных и сводных таблицах
- ◆ Основные формулы подсчета
- ◆ Дополнительные формулы подсчета ячеек
- ◆ Формулы суммирования
- ◆ Условное суммирование по одному критерию
- ◆ Условные суммы на основе множества критериев
- ◆ Использование функций VBA для подсчета и суммирования данных
- ◆ Резюме

Большая часть вопросов, чаще всего возникающих при создании электронных таблиц, как правило, касается подсчета и суммирования значений или других элементов рабочего листа. Поэтому многие пользователи заинтересованы в специальных формулах подсчета и суммирования данных. Работая над этой главой, мы постарались помочь найти решение большинства задач, стоящих практически перед любым пользователем Excel.

Подсчет и суммирование ячеек электронной таблицы

Формулы подсчета, как правило, возвращают число ячеек заданного диапазона, отвечающих определенному условию. Суммирующие формулы возвращают сумму значений в ячейках заданного диапазона, также отвечающих определенному условию. В свою очередь, диапазон ячеек, которые должны быть подсчитаны или суммированы, может либо содержать данные электронной таблицы, либо просто пустые ячейки.



Более подробно о формулах массива, используемых для подсчета и суммирования ячеек, см. в части V этой книги.

В табл. 7.1 приведены функции, которые используются при создании формул суммирования и подсчета. Если ни одна из приведенных функций не подходит для решения вашей задачи, вы можете использовать формулы массива, которые не рассматриваются в этой главе.

Таблица 7.1. Функции суммирования и подсчета в приложении Excel

Функция	Описание функции
СЧЁТ	Возвращает количество чисел в списке аргументов
СЧЁТЗ	Возвращает количество значений в списке аргументов и непустых ячеек
СЧИТАТЬПУСТОТЫ	Возвращает количество пустых ячеек в диапазоне
СЧЁТЕСЛИ	Подсчитывает количество непустых ячеек в диапазоне, удовлетворяющих заданному условию
ВСЧЁТ	Подсчитывает количество числовых ячеек в выборке из заданной базы данных по заданному критерию
ВСЧЁТА	Подсчитывает количество непустых ячеек в выборке из заданной базы данных по заданному критерию
КВАДРОТКЛ	Возвращает сумму квадратов отклонений точек данного от среднего по выборке; используется преимущественно в статистических формулах
БДСУММ	Суммирует числа в поле столбца записей базы данных, удовлетворяющих условию
ЧАСТОТА	Возвращает распределение частот в виде вертикального массива; используется только в формулах массивов со множеством ячеек
ПРОМЕЖУТОЧНЫЕ . ИТОГИ	При использовании значений от 2 до 3 в качестве первого аргумента возвращает количество ячеек, содержащихся в промежуточных итогах списка или базы данных; при использовании в качестве первого аргумента значения 9 возвращает сумму значений ячеек промежуточных итогов
СУММ	Возвращает сумму аргументов
СУММЕСЛИ	Суммирует ячейки, специфицированные заданным критерием
СУММПРОИЗВ	Возвращает сумму произведений соответствующих элементов массивов
СУММКВ	Возвращает сумму квадратов аргументов. Аргументами могут являться числа, массивы, имена или ссылки на ячейки, содержащие числа. Используется преимущественно в статистических формулах
СУММСУММКВ	Возвращает сумму сумм квадратов соответствующих элементов двух массивов; используется преимущественно в статистических формулах
СУММКВРАЗН	Возвращает сумму квадратов разностей соответствующих значений в двух массивах; используется преимущественно в статистических формулах
СУММРАЗНКВ	Возвращает сумму разностей квадратов соответствующих значений в двух массивах; используется преимущественно в статистических формулах

Подсчет или суммирование записей в базах данных и сводных таблицах

Приложение Excel предоставляет в распоряжение пользователя специальные функции подсчета и суммирования значений, содержащихся в базах данных и сводных таблицах. Функции БДСЧЁТ и БДСУММ — функции базы данных, которые работают с базой данных электронной таблицы и требуют указания определенного диапазона ячеек, удовлетворяющего условиям суммирования и подсчета.



Информацию о функциях базы данных и использовании формул подсчета и суммирования базы данных электронной таблицы см. в главе 9.

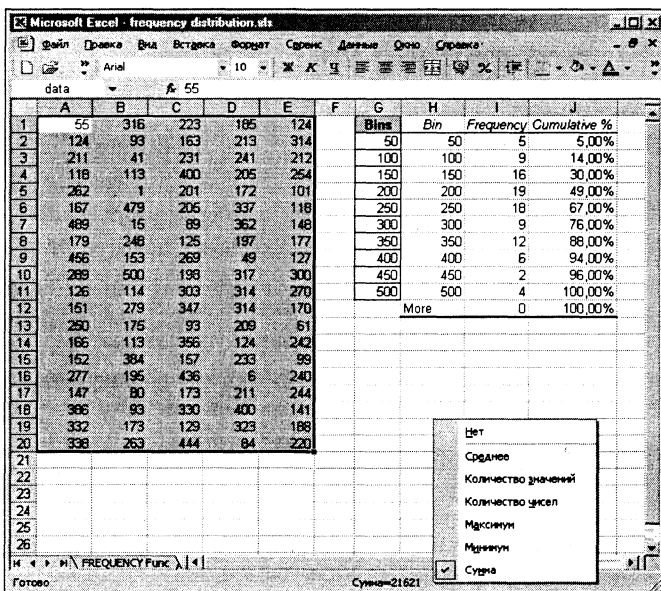
Эффективный способ подсчета или суммирования элементов электронной таблицы — это создание сводной таблицы. Как и функция базы данных, сводная таблица используется в том случае, если данные представлены в форме базы данных.



Для получения дополнительной информации о сводных таблицах обратитесь к главе 18.

Быстрый подсчет и суммирование

Начиная с Excel 97, компания Microsoft включила в приложение функциональную особенность под названием Автосумма, которая отображает в строке состояния информацию о выбранном диапазоне. Как правило, это сумма значений выбранного диапазона. Но вы можете изменить эту опцию. Для этого, правой кнопкой мыши щелкните на панели автосуммы в строке состояния и выберите из контекстного меню необходимую команду.



При выборе команды Количество значений, строка состояния будет отображать число пустых ячеек в выбранном диапазоне. При выборе команды Количество чисел — число ячеек в числовом формате в выбранном диапазоне.

Основные формулы подсчета

В этом разделе мы рассмотрим основные формулы подсчета, которые предоставляет пользователю приложение Excel. Все эти формулы достаточно просты в использовании и демонстрируют принципы применения функций, подсчитывающих количество ячеек заданного

диапазона, удовлетворяющих определенным условиям. Обратите внимание, на рис. 7.1 показан рабочий лист, где для суммирования содержимого 20-ти ячеек диапазона A1 : B10 под названием *Данные* используются специальные формулы (столбец E).

	A	B	C	D	E	F	G
1	Jan	Feb		Total cells:	20		
2	525	718		Blank cells:	6		
3				Nonblank cells:	14		
4	3			Numeric values:	7		
5	552	911		Non-text cells:	17		
6	250	98		Text cells:	3		
7				Logical values:	2		
8	ИСТИНА	ЛОЖЬ		Error values:	2		
9		#ДЕЛ/0!		#N/A errors:	0		
10	Total	#ИМЯ?		#NULL! errors:	0		
11				#DIV/0! errors:	1		
12				#VALUE! errors:	0		
13				#REF! errors:	0		
14				#NAME? errors:	1		
15				#NUM! errors:	0		
16							

Рис. 7.1. Формулы в столбце E могут быть использованы для проведения различных расчетов данных в диапазоне A1 : B10

Подсчет общего количества ячеек

Чтобы подсчитать общее количество ячеек в диапазоне, используйте формулу, приведенную ниже. Данная формула возвращает число ячеек в диапазоне *Данные*, умножая количество строк (возвращенное функцией ЧСТРОК) на количество столбцов (возвращенное функцией ЧИСЛСТОЛБ).

=ЧСТРОК (Данные) * ЧИСЛСТОЛБ (Данные)

Подсчет пустых ячеек

Следующая формула возвращает количество пустых ячеек в диапазоне *Данные*:

=СЧИТАТЬПУСТОТЫ (Данные)

Несколько слов о примерах, приведенных в этой главе

Многие примеры, приведенные в этой главе, содержат формулы массива. Формулы массива (как это будет рассмотрено далее в главе 14) относятся к специальному типу формул и при отображении в строке формул заключаются в фигурные скобки. Например:

{=Data*2}

Чтобы сообщить Excel, что введенная формула представляет собой формулу массива, после ввода формулы вместо обычного <Enter> нажмите комбинацию клавиш <Ctrl+Shift+Enter>. Вы также не должны вводить фигурные скобки, Excel сделает это за вас.

Функция СЧИТАТЬПУСТОТЫ также подсчитывает ячейки, содержащие формулу, которая возвращает пустую строку. К примеру, формула, приведенная ниже, возвращает пустую строку, если ячейка A1 содержит значение больше 5. В том случае, если ячейка A1 выполняет это условие, функция СЧИТАТЬПУСТОТЫ рассматривает ячейку с данной формулой как пустую строку.

=ЕСЛИ (A1>5; " "; A1)



Имейте в виду, что функция СЧИТАТЬПУСТОТЫ не учитывает ячейки, содержащие нулевое значение, даже в том случае, если в поле Нулевые значения диалогового окна Параметры не установлен флажок (чтобы проверить это, выберите команду меню Сервис⇒Параметры, а затем перейдите вкладку Вид).

Функция СЧИТАТЬПУСТОТЫ может использовать аргумент, представляющий собой целую строку или столбец. Например, приведенная ниже формула возвращает количество пустых ячеек в столбце А:

=СЧИТАТЬПУСТОТЫ(А:А)

Следующая формула возвращает количество пустых ячеек в пределах всего рабочего листа Лист1. Данная формула должна быть введена в другой лист, в противном случае будет создана циклическая ссылка:

=СЧИТАТЬПУСТОТЫ(Лист1!1:65536)

Подсчет непустых ячеек

Следующая формула использует функцию СЧЁТЗ и возвращает количество непустых ячеек в диапазоне *Данные*:

=СЧЁТЗ(Данные)

Функция СЧЁТЗ подсчитывает ячейки, содержащие значения, текст или логические значения (ИСТИНА или ЛОЖЬ).



Если ячейка содержит формулу, возвращающую пустую строку, она включается в итог, возвращенный функцией СЧЁТЗ, даже в том случае, если эта ячейка не содержит данных.

Подсчет ячеек, содержащих числовые значения

Следующая формула использует функцию СЧЁТ и позволяет подсчитать количество ячеек диапазона *Данные*, содержащих только числовые значения:

=СЧЁТ(Данные)

Ячейки, содержащие значения даты и времени, также рассматриваются как числовые значения. Ячейки, содержащие логические значения (ИСТИНА или ЛОЖЬ), как числовые значения не рассматриваются.

Подсчет ячеек, содержащих нетекстовые значения

Приведенная ниже формула массива использует функцию Excel ЕНТЕКСТ, которая возвращает значение ИСТИНА в том случае, если ее аргумент ссылается на ячейку, не содержащую текстовое значения, включая также пустые ячейки. Сама по себе формула возвращает количество ячеек, не содержащих текст (включая пустые ячейки):

{=СУММ(ЕСЛИ(ЕНТЕКСТ(Данные);1))}

Подсчет ячеек, содержащих текстовое значение

Для того чтобы подсчитать количество ячеек в диапазоне, содержащих текст, воспользуйтесь формулой массива, которая возвращает число ячеек, содержащих текстовое значение, в диапазоне *Данные*:

```
{=СУММ(ЕСЛИ(ЕТЕКСТ(Данные);1))}
```

Подсчет логических значений

Следующая формула массива возвращает количество логических значений (ИСТИНА или ЛОЖЬ) в диапазоне *Данные*:

```
{=СУММ(ЕСЛИ(ЕЛОГИЧ(Данные);1))}
```

Значения ошибки диапазона данных

Приложение Excel содержит три функции, которые позволяют пользователю определить, содержит ли ячейка ошибки:

- ◆ ЕОШИБКА возвращает значение ИСТИНА в том случае, если ячейка содержит любое значение ошибки (#Н/Д, #ССЫЛКА!, #ЗНАЧ!, #ДЕЛ/0!, #ИМЯ?, #НУЛЬ!).
- ◆ ЕОШ возвращает значение ИСТИНА, если ячейка содержит любое значение ошибки, за исключением #Н/Д.
- ◆ ЕНД возвращает значение ИСТИНА, если значение является значением ошибки #Н/Д.

Эти формулы используются в формуле массива для подсчета значений ошибки в определенном диапазоне. Например, следующая формула массива возвращает общее количество ячеек, содержащих значения ошибки, в диапазоне *Данные*:

```
{=СУММ(ЕСЛИ(ЕОШИБКА(Данные);1))}
```

При необходимости вместо функции ЕОШИБКА пользователь может использовать функции ЕОШ и ЕНД.

Чтобы подсчитать количество ошибок одного определенного типа, воспользуйтесь функцией СЧЁТЕСЛИ. Например, следующая формула возвращает количество ячеек с типом ошибки #ДЕЛ/0!, содержащихся в диапазоне *Данные*:

```
=СЧЁТЕСЛИ(Данные;"#ДЕЛ/0!")
```

Дополнительные формулы подсчета ячеек

Большинство примеров, представленных в предыдущем разделе, используют функции и формулы, осуществляющие подсчет ячеек, удовлетворяющих определенным условиям. В этом разделе мы рассмотрим более сложные примеры подсчета ячеек рабочего листа с использованием дополнительных формул, использующих различные виды условий.

Подсчет ячеек с использованием функции СЧЁТЕСЛИ

Использование функции Excel СЧЁТЕСЛИ наиболее эффективно в формулах, ограниченных по одному условию. Функция СЧЁТЕСЛИ использует два аргумента:

- ◆ *Диапазон*, содержащий значение, определяющее ячейки, включающиеся в расчет.
- ◆ *Логическое условие*, которое определяет подсчитываемые ячейки.

Ниже приведено несколько примеров формул, использующих функцию СЧЁТЕСЛИ. Все эти формулы используют для работы диапазон *Данные*. Обратите внимание, что аргумент *условие* может представлять собой любой тип данных: константы, выражения, функции, ссылки на ячейку и даже групповые символы (* и ?).

Следующая формула возвращает количество ячеек, содержащих значение 12:

=СЧЁТЕСЛИ(Данные; 12)

Данная формула возвращает количество ячеек, содержащих отрицательное значение:

=СЧЁТЕСЛИ(Данные; "<0")

Следующая формула возвращает количество ячеек, значение которых не равно 0:

=СЧЁТЕСЛИ(Данные; "<>0")

Формула возвращает количество ячеек, значение которых больше 5:

=СЧЁТЕСЛИ(Данные; ">5")

Формула возвращает количество ячеек, значение которых равно содержимому ячейки A1:

=СЧЁТЕСЛИ(Данные; A1)

Формула возвращает количество ячеек, значение которых больше, чем значение ячейке A1:

=СЧЁТЕСЛИ(Данные; ">"&A1)

Формула возвращает количество ячеек, содержащих текст:

=СЧЁТЕСЛИ(Данные; "*")

Формула возвращает количество ячеек, содержащих ровно три символа:

=СЧЁТЕСЛИ(Данные; "???")

Формула возвращает количество ячеек, содержащих слово *бюджет* (без учета регистра):

=СЧЁТЕСЛИ(Данные; "бюджет")

Формула возвращает количество ячеек, содержащих слово *бюджет* с любым другим текстом:

=СЧЁТЕСЛИ(Данные; "*"бюджет*")

Формула возвращает количество ячеек, содержащих текст, который начинается с символа *A* (без учета регистра):

=СЧЁТЕСЛИ(Данные; "A*")

Формула возвращает количество ячеек, содержащих текущую дату:

=СЧЁТЕСЛИ(Данные; СЕГОДНЯ ())

Формула возвращает количество ячеек, значение которых больше среднего:

=СЧЁТЕСЛИ(Данные; ">"&СРЗНАЧ(Данные))

Формула возвращает количество ячеек, значение которых превышает три стандартных отклонения от среднего:

=СЧЁТЕСЛИ (Данные; ">"&СРЗНАЧ (Данные) +СТАНДОТКЛОН (Данные) *3)

Формула возвращает количество ячеек, содержащих значение 3 или -3:

=СЧЁТЕСЛИ (Данные; 3) +СЧЁТЕСЛИ (Данные; -3)

Формула возвращает количество ячеек, содержащих значение ИСТИНА:

=СЧЁТЕСЛИ (Данные; ИСТИНА)

Формула возвращает количество ячеек, содержащих логическое значение (ИСТИНА или ЛОЖЬ):

=СЧЁТЕСЛИ (Данные; ИСТИНА) +СЧЁТЕСЛИ (Данные; ЛОЖЬ)

Формула возвращает количество ячеек, содержащих значение ошибки #Н/Д:

=СЧЁТЕСЛИ (Данные; "#Н/Д")

Подсчет ячеек с использованием нескольких условий

Во многих случаях, формула подсчета использует только те ячейки, которые удовлетворяют двум или более условиям. Выбор условий основывается на ячейках, которые уже подсчитаны или на диапазоне ячеек, удовлетворяющих определенным условиям.

Все примеры в этом разделе используют простой рабочий лист, отображенный на рис. 7.2, в котором в столбцах Month, SalesRep и Type приведены данные о продажах фирмы. Диапазоны ячеек данного рабочего листа соответствуют значениям, введенным в строку 1.

Использование оператора И

Используя оператор И, вы можете подсчитать все ячейки диапазона, отвечающие определенным условиям. Типичным примером использования этого условия является формула, которая подсчитывает количество значений, содержащихся в пределах числового диапазона. Предположим, что вам необходимо подсчитать ячейки, содержащие значения, которые больше 0 и меньше или равны 12. В данном случае в итог будет включена любая ячейка, содержащая положительное значение, не превышающее 12. Функция СЧЁТЕСЛИ выполняет следующее:

=СЧЁТЕСЛИ (Данные; ">0") -СЧЁТЕСЛИ (Данные; ">12")

Данная формула подсчитывает число значений больше 0, а затем из полученного итога вычитает число значений больше 12. Результат — количество ячеек, которые содержат значение больше 0 и меньше или равно 12.

Возможно, вид данной формулы может ввести вас в заблуждение, поскольку, несмотря на то, что ее цель состоит в том, чтобы подсчитать значения меньше или равные 12, здесь используется условие ">12". Существует еще одна методика, использующая формулу массива, которая представляет собой тип формулы в некоторой степени проще предыдущей:

{=СУММ((Данные>0) *Данные<=12))}

В некоторых случаях условия подсчета могут быть заданы не только для подсчитываемых, но и для других ячеек. Например, вы можете подсчитать количество продаж в ячейках, отвечающих следующим условиям:

- ◆ Месяц — январь *и*
- ◆ продавец — Брукс *и*
- ◆ количество продаж более 1000

Следующая формула массива возвращает число ячеек массива, отвечающих всем этим трем условиям:

```
{=СУММ(Мoнtн="January")*(SalesRep="Brooks")*(Amount>1000))}
```

Использование оператора ИЛИ

Чтобы подсчитать количество ячеек, используя оператор ИЛИ, в некоторых случаях задействовано множество функций СЧЁТЕСЛИ. Например, следующая формула подсчитывает количество значений 1, 3 и 5 в диапазоне *Данные*:

```
=СЧЁТЕСЛИ(Данные; 1)+СЧЁТЕСЛИ(Данные; 3)+СЧЁТЕСЛИ(Данные; 5)
```

Функция СЧЁТЕСЛИ также используется в формуле массива. К примеру, формула массива, приведенная ниже, возвращает тот же результат, что и предыдущая формула:

```
{=СУММ(СЧЁТЕСЛИ(Данные; {1; 3; 5}))}
```

Если же условие ИЛИ применяется не к подсчитываемым ячейкам, а к ячейкам другого диапазона, функция СЧЁТЕСЛИ работать не будет. Вернемся снова к рис. 7.2 и предположим, что вам необходимо подсчитать количество продаж, отвечающих следующим условиям:

- ◆ Месяц — январь *или*
- ◆ продавец — Брукс *или*
- ◆ количество продаж более 1000

Следующая формула массива возвращает правильный результат подсчета:

```
{=СУММ(ЕСЛИ((Month="January")+(SalesRep="Brooks")+(Amount>1000); 1))}
```

Совместное использование условий И и ИЛИ

При подсчете ячеек условия И и ИЛИ можно объединять. Предположим, что вы хотите подсчитать продажи, удовлетворяющие следующим условиям:

- ◆ Месяц — январь *и*
- ◆ продавец — Брукс *или* продавец — Кук

Следующая формула массива возвращает число продаж, удовлетворяющих этим условиям:

```
{=СУММ((Month="January")*ЕСЛИ((SalesRep="Brooks")+(SalesRep="Cook"); 1))}
```

Подсчет часто встречающихся записей

Функция МОДА возвращает значение, наиболее часто встречающееся в диапазоне ячеек. Обратите внимание, на рис. 7.3 показан рабочий лист, содержащий диапазон ячеек A1:A10 (под названием *Данные*). Формула, приведенная ниже, возвращает значение 10, поскольку это значение в данном диапазоне ячеек встречается наиболее часто:

```
=МОДА(Данные)
```

	A	B	C	D
1	Month	SalesRep	Type	Amount
2	January	Albert	New	85
3	January	Albert	New	675
4	January	Brooks	New	130
5	January	Cook	New	1350
6	January	Cook	Existing	685
7	January	Brooks	New	1350
8	January	Cook	New	475
9	January	Brooks	New	1205
10	February	Brooks	Existing	450
11	February	Albert	New	495
12	February	Cook	New	210
13	February	Cook	Existing	1050
14	February	Albert	New	140
15	February	Brooks	New	900
16	February	Brooks	New	900
17	February	Cook	New	95
18	February	Cook	New	780
19	March	Brooks	New	900
20	March	Albert	Existing	875
21	March	Brooks	New	50
22	March	Brooks	New	875
23	March	Cook	Existing	225
24	March	Cook	New	175
25	March	Brooks	Existing	400
26	March	Albert	New	840
27	March	Cook	New	132
28				

Рис. 7.2. На этом рабочем листе приведены различные варианты подсчета, использующие множество условий

Чтобы подсчитать количество наиболее часто встречающихся в заданном диапазоне значений (иными словами, частоту моды), воспользуйтесь следующей формулой:

=СЧЁТЕСЛИ (Данные; МОДА (Данные))

	A	B	C	D	E	F	G
1	1		10	<- Mode			
2		4					
3		7		3	<- Frequency of the Mode		
4		10					
5		10					
6		10					
7		12					
8		14					
9		16					
10		9					

Рис. 7.3. Функция МОДА возвращает значение, встречающееся наиболее часто в заданном диапазоне ячеек

	A	B	C	D	E	F	G
1	aa	Alpha					
2	Alpha						
3	AAA						
4	aaa						
5	Beta						
6	B						
7	BBB						
8	Alpha Beta						
9	AB						
10	alpha						
11		2 Entire cell (not case-sensitive)					
12		1 Entire cell (case-sensitive)					
13							
14							
15		3 Part of cell (not case-sensitive)					
16		2 Part of cell (case-sensitive)					
17							
18		3 Total occurrences in range (not case-sensitive)					
19		2 Total occurrences in range (case-sensitive)					
20							
21							

Рис. 7.4. Вы можете использовать различные способы подсчета символов в заданном диапазоне ячеек

Данная формула возвращает значение 3, поскольку значение моды (10) встречается в диапазоне *Данные* три раза.

Имейте в виду, что функция МОДА работает только с числовыми значениями. В том случае, если заданная ячейка содержит текст, функция ее игнорирует. Для того чтобы найти текстовую строку, наиболее часто встречающуюся в заданном диапазоне, используйте формулу массива.

Чтобы подсчитать количество элементов, наиболее часто встречающихся в заданном диапазоне *Данные*, будь-то текст или числовое значение, используйте следующую формулу массива:

{=МАКС (СЧЁТЕСЛИ (Данные ; Данные)) }

Следующая формула массива работает подобно функции МОДА, за исключением того, что здесь могут использоваться как числовые, так и текстовые значения:

{=ИНДЕКС (Данные ; ПОИСКПОЗ (МАКС (СЧЁТЕСЛИ (Данные ; Данные)) ; СЧЁТЕСЛИ (Данные ; Данные) ; 0)) }

Подсчет количества экземпляров определенного текста

В этом разделе мы рассмотрим различные способы подсчета количества экземпляров определенной строки символов или текстовой строки в заданном диапазоне ячеек. Все примеры в этом разделе используют рабочий лист, отображенный на рис. 7.4, который содержит в диапазоне ячеек A1:A10 (*Данные*) различную текстовую информацию; ячейка B1 носит название *Текст*.

Содержимое всей ячейки

Чтобы подсчитать количество ячеек, содержащих исключительно значение, введенное в ячейку *Текст* (и не содержащих никаких других символов), воспользуйтесь функцией СЧЁТЕСЛИ. Для этого создайте следующую формулу:

```
=СЧЁТЕСЛИ(Данные;Текст)
```

Например, если ячейка *Текст* содержит текстовую строку Alpha, формула возвращает значение 2, поскольку этот текст содержат две ячейки заданного диапазона *Данные*. Обратите внимание, что данная формула не учитывает регистр символов и воспринимает содержимое ячейки A2 и ячейки A10 как идентичные текстовые строки. Тем не менее, содержимое ячейки A8 данная формула игнорирует.

Действие следующей формулы массива подобно предыдущей, за исключением того, что она учитывает регистр символов:

```
{=СУММ(ЕСЛИ(СОВПАД(Данные;Текст);1))}
```

Часть содержимого ячейки

Чтобы подсчитать количество ячеек, содержащих строку, которая включает слово *Текст*, используется следующая формула:

```
=СЧЁТЕСЛИ(Данные;"*"&Текст&"*")
```

К примеру, если ячейка *Текст* содержит текст Alpha, данная формула возвращает значение 3, поскольку в диапазоне *Данные* текст Alpha содержат три ячейки: A2, A8 и A10. Обратите внимание, что регистр в данном случае не учитывается.

Для того чтобы подсчитать количество ячеек, включающих заданный текст, используйте следующую формулу массива:

```
{=СУММ(ЕСЛИ(ДЛСТР(Данные)-ДЛСТР(ПОДСТАВИТЬ(Данные;Текст;""))>0;1))}
```

Если ячейка *Текст* содержит текст Alpha, предыдущая формула возвращает значение 2, поскольку данная текстовая строка встречается в двух ячейках диапазона (A2 и A8).

Общее количество экземпляров строки в заданном диапазоне

Чтобы подсчитать общее количество экземпляров строки в заданном диапазоне ячеек, используйте следующую формулу массива:

```
{=СУММ(ДЛСТР(Данные))-СУММ(ДЛСТР(ПОДСТАВИТЬ(Данные;Текст;"")))/ДЛСТР(Текст)}
```

Если ячейка *Текст* содержит символ B, данная формула возвращает значение 7 — этот символ содержит семь ячеек заданного диапазона. Имейте в виду, что эта формула учитывает регистр символа. Чтобы игнорировать регистр, используйте следующую измененную формулу массива:

```
{=СУММ(ДЛСТР(Данные))-СУММ(ДЛСТР(ПОДСТАВИТЬ(ПРОПИСН(Данные);ПРОПИСН(Текст);"")))/ДЛСТР(Текст)}
```

Подсчет количества одинаковых значений

Следующая формула массива возвращает число одинаковых значений в диапазоне *Данные*:

```
{=СУММ(1/СЧЁТЕСЛИ(Данные;Данные))}
```

Для полного понимания того, как работает эта формула, вам необходимо усвоить основные принципы работы формул массива. Обратите внимание, на рис. 7.5 отображен рабочий

лист, содержащий два диапазона ячеек: диапазон A1:A12 (*Данные*) и диапазон C1:C12, все ячейки которого содержат следующую формулу массива:

{=СЧЁТЕСЛИ(Данные; Данные)}

	A	B	C	D	E	F	G
1	100	3	0,333333				
2	100	3	0,333333				
3	100	3	0,333333				
4	200	2	0,5				
5	200	2	0,5				
6	300	1	1				
7	400	2	0,5				
8	400	2	0,5				
9	500	4	0,25				
10	500	4	0,25				
11	500	4	0,25				
12	500	4	0,25				
13				5 <- Unique items in Column A			
14							
15				5 <- Single formula			
16							
17							
18							

Рис. 7.5. Формулы массива используются для подсчета количества одинаковых значений в заданном диапазоне ячеек

Массив в диапазоне C1:C12 содержит итоги, подсчитанные для каждого значения в массиве *Данные*. Например, число 100 появляется в массиве *Данные* три раза. Таким образом, каждый элемент массива C1:C12, соответствующий значению 100 в диапазоне *Данные*, будет содержать значение 3.

Диапазон D1:D12 содержит следующую формулу массива:

{=1/C1:C12}

Данный массив состоит из значений, полученных путем деления 1 на значения массива в диапазоне ячеек C1:C12. Например, если какая-либо из ячеек исходного диапазона *Данные* содержит значение 200, то соответствующая ей ячейка в диапазоне D1:D12 будет содержать значение 0,05.

Суммирование значений диапазона D1:D12 дает число одинаковых элементов, содержащихся в массиве *Данные*. По существу, формула, приведенная в начале этого раздела, создает массив значений в диапазоне ячеек D1:D12 и суммирует значения этого массива.

Но эта формула имеет серьезное ограничение: в том случае, если диапазон ячеек содержит пустые ячейки, формула возвращает ошибку. Приведенная ниже формула массива решает эту проблему:

{=СУММ(ЕСЛИ(СЧЁТЕСЛИ(Данные; Данные)=0; " "; 1/СЧЁТЕСЛИ(Данные; Данные)))}



Чтобы создать формулу массива, возвращающую список одинаковых значений в заданном диапазоне ячеек, см. главу 15.

Распределение частот

Основной составляющей распределения частот является итоговая таблица, которая отображает частоту повторения каждого значения диапазона. Например, преподаватель школы или ВУЗа может создать распределение частот экзаменационных оценок, то есть таблицу, в

которой отображалось бы количество значений А, В, С и др. Чтобы создать распределение частот, вы можете воспользоваться одним из следующих способов, которые Excel предоставляет в распоряжение пользователя:

- ◆ Использовать функцию ЧАСТОТА
- ◆ Создать свою собственную формулу
- ◆ Использовать надстройку Excel Analysis ToolPak



Если ваши данные находятся в форме базы данных, для создания распределения частот вы можете также использовать сводную таблицу.

Функция ЧАСТОТА

Самый простой способ создания распределения частот в приложении Excel — это использовать функцию ЧАСТОТА. Эта функция возвращает массив, поэтому она всегда должна использоваться в формуле массива, введенной в диапазон множества ячеек.

Обратите внимание, на рис. 7.6 в диапазон ячеек А1 : Е20 (Данные) введены значения от 1 до 500. Диапазон G2 : G11 задает уровни дискретизации, используемые при распределении частот. Каждая ячейка этого диапазона представляет собой верхний предел уровня. В данном случае используются уровни: 1-50, 51-100, 101-150 и так далее. Чтобы найти простой способ создания уровней дискретизации, обратитесь к врезке “Дискретизация в распределении частот”.

Для того чтобы создать распределение частот, выберите диапазон ячеек, соответствующий количеству ячеек в диапазоне уровней, после чего введите следующую формулу массива:

{=ЧАСТОТА (Данные ; G2 : G11) }

	A	B	C	D	E	F	G
1	55	316	223	185	124		
2	124	93	163	213	314		50
3	211	41	231	241	212		100
4	118	113	400	205	254		150
5	262	1	201	172	101		200
6	167	479	205	337	118		250
7	489	15	89	362	148		300
8	179	248	125	197	177		350
9	456	153	269	49	127		400
10	289	500	198	317	300		450
11	126	114	303	314	270		500
12	151	279	347	314	170		
13	250	175	93	209	61		
14	165	113	356	124	242		
15	152	384	157	233	99		
16	277	195	436	6	240		
17	147	80	173	211	244		
18	386	93	330	400	141		
19	332	173	129	323	188		
20	338	263	444	84	220		

Рис. 7.6. Создание распределения частот для диапазона ячеек А1 : Е20

Данная формула массива подсчитывает количество значений в диапазоне Данные, попадающих в каждый уровень дискретизации. Чтобы создать распределение частот в виде процентного распределения, используйте следующую функцию:

{=ЧАСТОТА (Данные ; G2 : G10) / СЧЁТ (Данные) }

Обратите внимание, на рис. 7.7 отображены два столбца распределения частот: один в числовых значениях, а другой в процентах. Кроме того, на рис. 7.7 отображена диаграмма (или гистограмма), созданная на основе распределения частот.

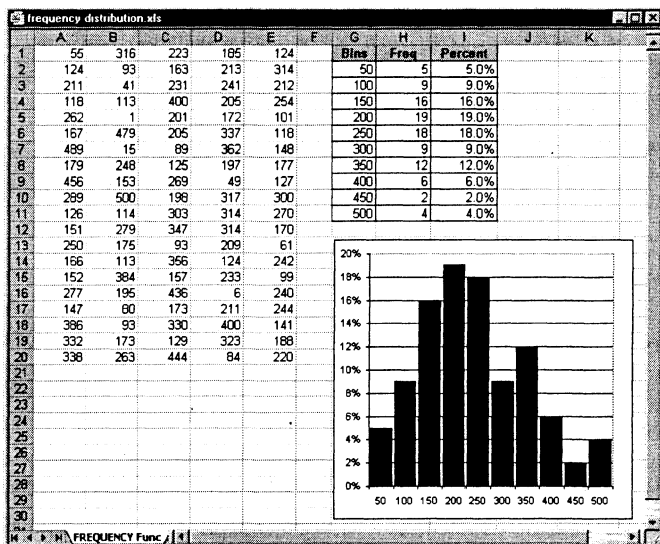


Рис. 7.7. Создание распределения частот с помощью функции ЧАСТОТА

Использование формул для создания распределения частот

На рис. 7.8 отображен рабочий лист, который содержит экзаменационные оценки 50 студентов (столбец В — *Grade*). Формулы в столбцах G и H вычисляют распределение оценок студентов. Минимальные и максимальные баллы для каждой оценки отображаются в столбцах D и E. К примеру, экзаменационная оценка, находящаяся в интервале между 80 и 89 (включительно), помещается в столбец В диаграммы.

Дискретизация в распределении частот

Прежде чем начать создавать распределение частот, в диапазоне значений сначала необходимо определить уровни дискретизации. Количество элементов дискретизации задает число категорий в распределении. В большинстве случаев каждый из элементов представляет равнозначный диапазон значений.

Для того чтобы создать 10 равномерно удаленных друг от друга уровней дискретизации для значений диапазона *Данные*, в диапазон, состоящий из 10 ячеек одного столбца, введите следующую формулу массива:

```
{МИН(Данные) + (СТРОКА(ДВССЫЛ("1:10")) * (МАКС(Данные) - МИН(Данные) + 1) / 10) - 1}
```

Данная формула создает 10 элементов дискретизации на основе значений диапазона *Данные*. Верхнее значение элемента всегда соответствует максимальному значению диапазона.

Для того чтобы создать большее или меньшее количество уровней дискретизации, необходимо использовать другое значение, отличное от 10, а саму формулу массива ввести в диапазон, содержащий такое же количество ячеек. Например, чтобы создать пять уровней дискретизации, в диапазон, состоящий из пяти расположенных вертикально ячеек, введите следующую формулу массива:

```
{МИН(Данные) + (СТРОКА(ДВССЫЛ("1:5")) * (МАКС(Данные) - МИН(Данные) + 1) / 5) - 1}
```

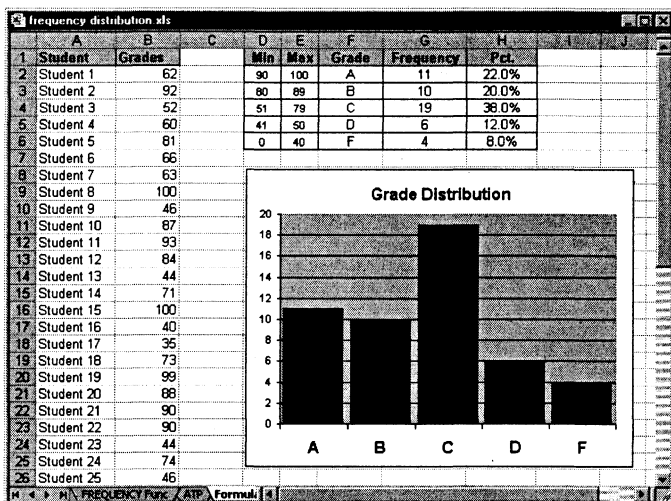


Рис. 7.8. Создание распределения экзаменационных отметок

Формула в ячейке G2, приведенная ниже, представляет собой формулу массива, которая подсчитывает количество баллов, относящихся к оценке A:

$\{=СУММ((Grades \geq D2) * (Grades \leq E2))\}$

Вспомните, мы уже использовали эту формулу в предыдущем разделе этой книги (см. раздел “Подсчет ячеек с использованием нескольких условий”). В данном примере мы просто скопировали ее на две ячейки ниже ячейки G2.

Формулы в столбце H вычисляют процентное соотношение оценок. Формула, содержащаяся в ячейке H2, скопирована на четыре ячейки ниже H2:

$=G2 / СУММ(\$G\$2 : \$G\$6)$

Использование надстройки “Пакет анализа” для создания распределения частот

Если вы предварительно установили надстройку “Пакет анализа”, для создания распределения частот вы можете использовать опцию Гистограмма. Перед тем как начать работу с этой опцией, введите в диапазон необходимые значения уровней дискретизации. После этого выберите команду Сервис⇒Анализ данных. На экране отобразится диалоговое окно Анализ данных. Выберите из списка опцию Гистограмма и щелкните на кнопке ОК, после чего на экране отобразится диалоговое окно Гистограмма, показанное на рис. 7.9.

В полях Входной интервал, Интервал карманов и Выходной интервал установите соответственно диапазоны входных данных, элементов дискретизации и выходные данные, после чего задайте дополнительные опции. Обратите внимание, на рис. 7.10 отображено распределение частот (и диаграмма), созданные с помощью опции Гистограмма.

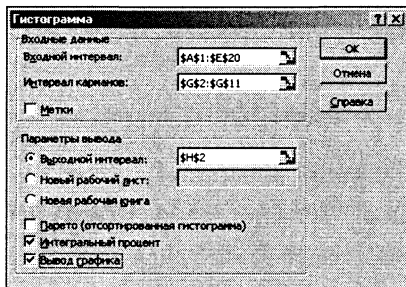


Рис. 7.9. Диалоговое окно Гистограмма надстройки “Пакет анализа”

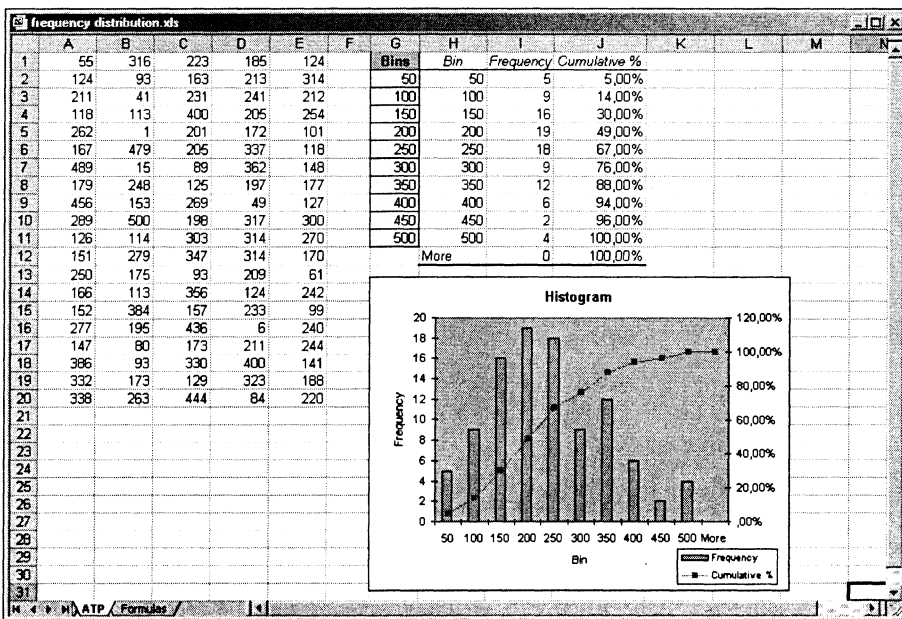


Рис. 7.10. Данные распределения частоты и диаграмма созданы с помощью опции Гистограмма надстройки "Пакет анализа"



Имейте в виду, что распределение частот строится не на основе формул, а значений. Поэтому, при внесении изменений во входные данные, вы должны повторно запустить функцию Гистограмма, чтобы модифицировать графические результаты.

Использование настраиваемых элементов дискретизации для создания гистограмм

На рис. 7.11 показан рабочий лист, в котором в столбце В перечислены оценки студентов (общее количество студентов — 67 человек). Столбцы D и E содержат формулы, которые вычисляют верхние и нижние пределы уровней дискретизации на основе данных ячейки E1 (под названием *BinSize*). Например, если ячейка *BinSize* содержит значение 10 (см. рис. 7.11), каждый следующий элемент будет отстоять от предыдущего на 10 единиц (1-10, 11-20 и так далее).

Данная диаграмма использует в формуле РЯД два динамических имени. Имя *Categories* может быть определено с помощью следующей формулы:

=СМЕЩ(Лист1!\$E\$4;0;0;ОКРУГЛВВЕРХ(100/BinSize;0))

Имя *Frequency* определяется формулой:

=СМЕЩ(Лист1!\$F\$4;0;0;ОКРУГЛВВЕРХ(100/BinSize;0))

В результате, после изменения ячейки *BinSize*, диаграмма будет корректироваться автоматически.



Дополнительная информация о создании диаграмм, использующих динамические имена в формуле РЯД, приведена в главе 17.

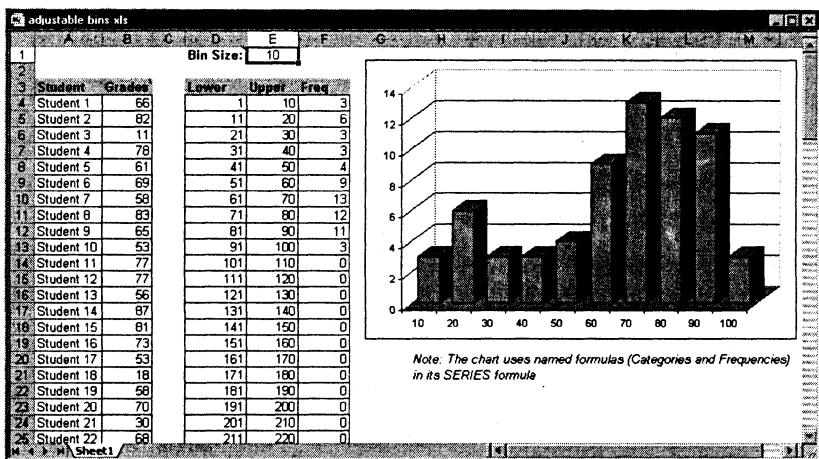


Рис. 7.11. Диаграмма оценок студентов, представленная в виде гистограммы; шаг между уровнями дискретизации значений устанавливает значение ячейки E1

Формулы суммирования

В этом разделе мы коснемся основных принципов суммирования ячеек, а также рассмотрим примеры формул суммирования: начиная с простейших формул суммирования и заканчивая сложными формулами массивов, которые вычисляют значения на основе множества критериев.

Суммирование всех ячеек диапазона

Нет ничего проще, чем прямое суммирование всех ячеек диапазона. Например, следующая формула возвращает сумму всех значений диапазона *Данные*:

=СУММ(Данные)

Функция СУММ может использовать до 32 аргументов. Следующая формула возвращает сумму значений в пяти диапазонах, расположенных в нескольких несмежных областях:

=СУММ(A1:A9;C1:C9;E1:E9;G1:G9;I1:I9)

В некоторых случаях в качестве аргумента функции СУММ могут использоваться отдельные строки или столбцы электронной таблицы. Например, формула, приведенная ниже, возвращает сумму всех значений в столбце A. Однако, если эту формулу ввести в ячейку столбца A, это сгенерирует ошибку циклической ссылки.

=СУММ(A:A)

Следующая формула возвращает сумму всех значений, содержащихся на рабочем листе Лист1. Чтобы при создании этой формулы избежать циклической зависимости, формула должна использоваться в другом рабочем листе, отличном от Лист1.

=СУММ(Лист1!1:65536)

Функция СУММ достаточно универсальна. В качестве аргументов этой функции используются числовые значения, ячейки, диапазоны, текстовые представления чисел (которые интерпретируются как значения), логические значения и даже вложенные функции. Например, обратите внимание на следующую формулу:

=СУММ(B1;5;"6"; ;КОРЕНЬ(4);A1:A5;ИСТИНА)

Данная формула вполне допустима для использования и содержит типы аргументов, которые перечислены ниже в порядке их представления:

- ◆ простая ссылка на одну ячейку;
- ◆ значение, представленное константой;
- ◆ строка, похожая на числовое значение;
- ◆ аргумент отсутствует;
- ◆ выражение, использующее другую функцию;
- ◆ ссылка на диапазон значений;
- ◆ логическое значение ИСТИНА.



Несмотря на всю универсальность, функция СУММ в отношении логических значений (ИСТИНА или ЛОЖЬ) ведет себя достаточно противоречиво. Например, логические значения, сохраненные в ячейках, всегда обрабатываются как 0, однако при использовании логического значения ИСТИНА в качестве аргумента функции СУММ оно обрабатывается как 1.

Общая сумма

В некоторых случаях, если это необходимо, пользователь может вычислить общую сумму значений заданного диапазона, иногда еще называемой “текущей суммой”. На рис. 7.12 приведен пример такого вычисления. В столбце В приведены суммы по месяцам, а в столбце С вычисляется общая сумма, которая равна общей сумме на текущий месяц.

Month	Amount	Year-to-Date
January	850	850
February	900	1,750
March	750	2,500
April	1,100	3,600
May	600	4,200
June	500	4,700
July	1,200	5,900
August		5,900
September		5,900
October		5,900
November		5,900
December		5,900
TOTAL	5,900	

Рис. 7.12. С помощью простой формулы, введенной в столбец С, в нем представлена кумулятивная сумма значений из столбца В

Month	Amount	Year-to-Date
January	850	850
February	900	1,750
March	750	2,500
April	1,100	3,600
May	600	4,200
June	500	4,700
July	1,200	5,900
August		5,900
September		
October		
November		
December		
TOTAL	5,900	

Рис. 7.13. Чтобы скрыть кумулятивные суммы для строк с отсутствующими данными, используйте функцию ЕСЛИ

В ячейку С2 введена следующая формула:

=СУММ (В\$2 : В2)

Обратите внимание, что данная формула использует смешанную ссылку. Первой ячейкой диапазона, на которую будет ссылаться формула, всегда будет ячейка в строке 2. При копировании этой формулы вниз по столбцу этот диапазон будет изменяться, но таким образом, что суммирование ячеек всегда будет начинаться со строки 2 и заканчиваться текущей строкой. Например, после копирования этой формулы вниз по столбцу С, ячейка С8 будет содержать следующую формулу:

=СУММ (В\$2 : В8)

Чтобы скрыть общие суммы для строк, данные в которые еще не введены, используйте функцию ЕСЛИ, как это сделано в формуле, которая введена в ячейку С2 и скопирована вниз по столбцу:

```
=ЕСЛИ (B2<>" ";СУММ (B$2:В2) ; " ")
```

На рис. 7.13 показано, как работает эта формула.

Суммирование *n*-го количества наибольших или наименьших значений

В отдельных случаях у пользователя может возникнуть необходимость суммировать некоторое *n*-е количество самых больших значений заданного диапазона, например, десяти. Один из способов решения этой проблемы — это сортировать диапазон ячеек в порядке убывания, а затем использовать функцию СУММ и первые *n* значений в сортируемом диапазоне в качестве ее аргументов. Следующая формула массива выполняет эту задачу, не требуя предварительной сортировки значений:

```
{=СУММ (НАИБОЛЬШИЙ (Данные ; {1;2;3;4;5;6;7;8;9;10} ) ) }
```

Эта формула суммирует 10 наибольших значений в диапазоне *Данные*. Для того чтобы суммировать 10 наименьших значений, вместо функции НАИБОЛЬШИЙ просто используйте функцию НАИМЕНЬШИЙ:

```
{=СУММ (НАИМЕНЬШИЙ (Данные ; {1;2;3;4;5;6;7;8;9;10} ) ) }
```

Обе эти формулы используют константу массива, представляющую собой аргумент функций НАИБОЛЬШИЙ или НАИМЕНЬШИЙ. В том случае, если вы имеете дело с диапазоном несоразмерно большим, чем запрашиваемое количество наибольших значений, вы можете воспользоваться формулой, приведенной ниже, которая возвращает сумму наибольших 30 значений диапазона *Данные*. При необходимости значение 30 может быть заменено на любое другое значение.

```
{=СУММ (НАИБОЛЬШИЙ (Данные ; СТРОКА (ДВССЫЛ ("1:30" ) ) ) ) }
```

Условное суммирование по одному критерию

Достаточно часто, в расчетах возникает необходимость использовать *условную сумму*. Что она из себя представляет? Это сумма, которая складывается из значений диапазона, отвечающих одному или более условиям. В этом разделе мы будем рассматривать примеры условного суммирования, удовлетворяющего только одному критерию.

Функция СУММЕСЛИ весьма полезна для суммирующих формул, использующих одно условие. Функция использует три аргумента:

- ♦ *Диапазон*, устанавливающий ячейки со значениями, включаемыми в сумму.
- ♦ *Условие* в форме числа, выражения или текста, которое определяет добавляемую ячейку.
- ♦ *Диапазон суммирования* — необязательный аргумент. Устанавливает диапазон ячеек, которые анализируются и подлежат суммированию. Если этот аргумент опускается, функция использует диапазон ячеек, указанный в первом аргументе.

Приведенные ниже примеры демонстрируют принцип использования функции СУММЕСЛИ. Все формулы для расчета используют данные рабочего листа, показанного на рис. 7.14, в котором столбец F содержит формулу, которая вычитает дату, указанную в столбце E, из даты в столбце D.

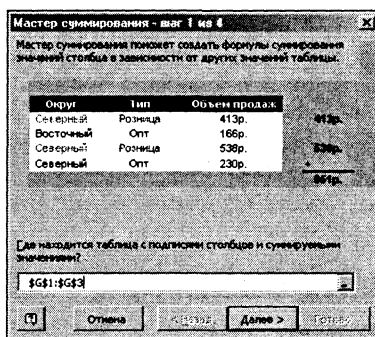
Отрицательное значение в столбце F свидетельствует о том, что оплата просрочена. Рабочий лист использует несколько диапазонов, именуемых в соответствии со значениями строки 1.

	A	B	C	D	E	F	G	H
1	InvoiceNum	Office	Amount	DateDue	Today	Difference		
2	AG-0145	Oregon	\$5 000,00	03 apr	06 май	-33		
3	AG-0189	California	\$450,00	21 apr	06 май	-15		
4	AG-0220	Washington	\$3 211,56	30 apr	06 май	-6		
5	AG-0310	Oregon	\$250,00	02 май	06 май	-4		
6	AG-0355	Washington	\$125,50	06 май	06 май	0		
7	AG-0409	Washington	\$3 000,00	12 май	06 май	6		
8	AG-0581	Oregon	\$2 100,00	25 май	06 май	19		
9	AG-0600	Oregon	\$335,39	25 май	06 май	19		
10	AG-0602	Washington	\$65,00	30 май	06 май	24		
11	AG-0633	California	\$250,00	01 июн	06 май	26		
12	TOTAL		\$14 787,45			36		
13								

Рис. 7.14. Отрицательное значение в столбце F свидетельствует о том, что оплата просрочена

Мастер создания формул

Начиная с версии Excel 97, в приложение Excel встроена надстройка под названием Мастер суммирования. Для того чтобы вызвать мастер суммирования, предварительно установив эту надстройку, выберите команду Сервис⇒Мастер⇒Частичная сумма.



Используя мастер суммирования вы можете устанавливать различные условия суммирования, а также создавать формулы суммирования, которые всегда будут формулами массива. Несмотря на то, что мастер суммирования представляет собой весьма удобный инструмент, тем не менее, он не универсален. Например, вы можете объединить несколько условий, используя условие И, однако не можете использовать условие ИЛИ.

Имейте в виду, что таблица данных, отображенная в центральной части мастера, никоим образом не связана с данными пользователя.

Суммирование только отрицательных значений

Следующая формула возвращает сумму отрицательных значений, содержащихся в столбце F. Выразаясь другими словами, функция возвращает общее количество просроченных дней оплаты всех счетов. В нашем случае формула возвращает значение -58.

=СУММЕСЛИ(Difference; "<0")

Поскольку, в данном случае, третий аргумент опускается, второй аргумент функции ("<0") обращается к значениям диапазона *Difference*.



Для того чтобы суммировать отрицательные значения диапазона *Difference*, можно также использовать следующую формулу массива:

```
{=СУММ(ЕСЛИ(Difference<0;Difference))}
```

Нет никакой необходимости записывать сложный код, чтобы ввести в формулу аргумент функции СУММЕСЛИ. Например, следующая формула создана со ссылкой на содержимое ячейки G2, которая и представляет собой аргумент функции:

```
=СУММ(Difference;G2)
```

В том случае, если аргумент в ячейке G2 изменяется, формула возвращает новый результат.

Суммирование значений на основе диапазона *Difference*

Следующая формула возвращает сумму всех просроченных счетов оплаты (из столбца C):

```
=СУММЕСЛИ(Difference;"<0";Amount)
```

Данная формула использует значения диапазона *Difference*, чтобы определить, какие из значений диапазона *Amount* участвуют в формировании суммы проведенных платежей.



Чтобы вычислить сумму значений диапазона *Amount*, для которых соответствующие значения диапазона *Difference* являются отрицательными, вы можете воспользоваться следующей формулой:

```
{=СУММ(ЕСЛИ(Difference<0;Amount))}
```

Суммирование значений на основе сравнения текста

Следующая формула возвращает общую сумму счетов офиса Штата Орегон:

```
=СУММЕСЛИ(Office;"=Oregon";Amount)
```

Использование знака равенства необязательно. Следующая формула возвращает аналогичный результат:

```
=СУММЕСЛИ(Office;"Oregon";Amount)
```

Чтобы суммировать общие суммы счетов всех офисов кроме Штата Орегон, используйте следующую формулу:

```
=СУММЕСЛИ(Office;"<>Oregon";Amount)
```

Суммирование значений на основе сравнения даты

Следующая формула возвращает общую сумму счетов, дата оплаты которых после 1 июня 1999 года:

```
=СУММЕСЛИ(DateDue;">="&ДАТА(1999;1;6);Amount)
```

Обратите внимание, что второй аргумент функции СУММЕСЛИ представляет собой выражение, использующее функцию ДАТА, которая возвращает значение даты. Заключенный в ка-

вычки оператор сравнения, используя оператор &, объединяет значение с результатом, полученным функцией ДАТА.

Формула, приведенная ниже, возвращает общую сумму счетов, дата оплаты которых приходится на будущий период времени (включая текущую дату):

=СУММЕСЛИ(DateDue; ">="&СЕГОДНЯ(); Amount)

Условные суммы на основе множества критериев

Все примеры, рассмотренные нами в предыдущем разделе, использовали только одно условие сравнения данных. В этом разделе мы будем работать с формулами суммирования, использующими множество условий. Имейте в виду, что функция СУММЕСЛИ не работает со множеством условий, поэтому здесь необходимо использовать формулу массива. На рис. 7.15 показан рабочий лист, аналогичный использовавшемуся в предыдущем разделе, данные которого мы будем использовать.

	A	B	C	D	E	F	G	H
1	InvoiceNum	Office	Amount	DateDue	Today	Difference		
2	AG-0145	Oregon	\$5 000.00	03 апр	06.май	-33		
3	AG-0189	California	\$450.00	21 апр	06.май	-15		
4	AG-0220	Washington	\$3 211.56	30 апр	06.май	-6		
5	AG-0310	Oregon	\$250.00	02.май	06.май	-4		
6	AG-0355	Washington	\$125.50	06.май	06.май	0		
7	AG-0409	Washington	\$3 000.00	12.май	06.май	6		
8	AG-0581	Oregon	\$2 100.00	25.май	06.май	19		
9	AG-0600	Oregon	\$335.39	25.май	06.май	19		
10	AG-0602	Washington	\$65.00	30.май	06.май	24		
11	AG-0633	California	\$250.00	01.июн	06.май	26		
12	TOTAL		\$14 767.45			36		
13								

Рис. 7.15. На этом рабочем листе суммирование проводится с использованием множества условий

Использование оператора И

Предположим, что вам необходимо подсчитать общую сумму по счетам, оплата которых просрочена и которые, в то же время, относятся к счетам офиса штата Орегон. Иными словами, значения, содержащиеся в диапазоне *Amount* будут суммироваться только в том случае, если они одновременно соответствуют следующим условиям:

- ♦ В диапазоне *Difference* им соответствуют отрицательные значения.
- ♦ Соответствующий им текст в диапазоне *Office* — это текст *Oregon*.

Приведенная ниже формула массива выполняет следующее:

{=СУММ((Difference<0)*Office="Oregon")*Amount)}

Данная формула создает в памяти системы два новых массива:

- ♦ Булев массив, содержащий значение ИСТИНА в том случае, если соответствующее значение диапазона *Difference* меньше нуля, и значение ЛОЖЬ — если значение диапазона больше нуля.

- ♦ Булев массив, содержащий значение ИСТИНА в том случае, если соответствующее значение диапазона *Office* равно значению Oregon, и значение ЛОЖЬ в обратном случае.

Умножение булевых значений дает следующие результаты:

ИСТИНА * ИСТИНА = 1

ИСТИНА * ЛОЖЬ = 0

ЛОЖЬ * ЛОЖЬ = 0

Значение диапазона *Amount*, соответствующее заданным условиям, возвращается как ненулевое значение только в том случае, если оба соответствующих ему значения, находящихся в массивах памяти, имеют значение ИСТИНА. Результатом является сумма значений диапазона *Amount*, удовлетворяющих определенным условиям.



Можно предположить, что предыдущая формула массива переписана с использованием функции СУММПРОИЗВ, выполняющей умножение и сложение элементов массивов:

```
=СУММПРОИЗВ((Difference<0);(Office="Oregon");Amount)
```

Однако, в силу некоторых причин, функция СУММПРОИЗВ обрабатывает булевы значения некорректно, поэтому данная формула не работает. Тем не менее, можно использовать другую формулу, которая умножает булевы значения на 1:

```
=СУММПРОИЗВ(1*(Difference<0);1*(Office="Oregon");Amount)
```

Использование оператора ИЛИ

Предположим, что вам необходимо получить общую сумму просроченных счетов или счетов офиса штата Орегон. Иными словами, вы должны суммировать те значения диапазона *Amount*, которые отвечают хотя бы одному из следующих условий:

- ♦ В диапазоне *Difference* им соответствуют отрицательные значения.
- ♦ Соответствующий им текст в диапазоне *Office* — это текст Oregon.

Приведенная ниже формула массива выполняет следующее:

```
{=СУММ(ЕСЛИ((Office="Oregon")+(Difference<0);1;0)*Amount)}
```

Знак (+) означает объединение условий, количество которых не ограничивается двумя, как это показано в данном примере, а может быть и больше.

Использование условий И и ИЛИ

Как правило, все становится несколько сложнее, когда пользователь использует условие, состоящее из двух операторов И и ИЛИ. Например, предположим, что нам необходимо суммировать значения диапазона *Amount*, отвечающие следующим условиям:

- ♦ В диапазоне *Difference* им соответствуют отрицательные значения.
- ♦ Соответствующий им текст в диапазоне *Office* — это текст Oregon или California.

Обратите внимание, что второе условие в действительности состоит из двух условий, объединенных с помощью оператора ИЛИ. Эта задача выполняется с помощью следующей формулы:

```
{СУММ((Difference<0)*ЕСЛИ((Office="Oregon")+(Office="California");1)*Amount)}
```


Использование функций VBA для подсчета и суммирования данных

Иногда вы можете столкнуться с тем, что некоторые виды задач подсчета и суммирования данных просто невозможно выполнить с помощью функций, встроенных в Excel, а также формул массива. К счастью, на этот случай приложение Excel имеет мощный инструмент — язык программирования Visual Basic for Application (VBA), предназначенный для создания пользовательских функций, которые, в отличие от встроенных, позволяют решить эти задачи.

Более подробно функции VBA будут рассмотрены в части IV этой книги. В главе 25 приведены примеры некоторых пользовательских функций подсчета и суммирования данных. Ниже приводится краткое описание этих функций:

- ◆ Функция COUNTBETWEEN. Возвращает количество ячеек, которые содержат значение, расположенное между двумя заданными величинами.
- ◆ Функция COUNTVISIBLE. Возвращает количество видимых ячеек заданного диапазона.
- ◆ Функция DATATYPE. Возвращает строку с описанием типа данных, содержащихся в ячейке. Данная функция позволяет подсчитать ячейки, которые содержат даты (что, как правило, не возможно).
- ◆ Функции ISBOLD, ISITALIC, FILLCOLOR. Функции возвращают значение ИСТИНА в том случае, если указанная ячейка имеет определенный тип форматирования (полужирный шрифт, курсив или шрифт определенного цвета). Данные функции могут быть использованы для суммирования и подсчета ячеек на основе их форматирования.
- ◆ Функция NUMBERFORMAT. Возвращает числовой формат ячейки. Данная функция позволяет подсчитать или суммировать ячейки на основе их числового формата.
- ◆ Функция SUMVISIBLE. Возвращает сумму видимых ячеек заданного диапазона.

Резюме

В этой главе мы рассмотрели множество примеров функций и формул, предназначенных для подсчета и суммирования ячеек, отвечающих некоторым условиям. Многие из этих формул представляют собой формулы массива.

В следующей главе мы обсудим использование формул для поиска информации в таблицах или диапазонах данных.

Глава 8

Поиск данных

В этой главе...

- ◆ Знакомство с формулой поиска данных
- ◆ Функции, используемые для поиска данных
- ◆ Основные формулы поиска данных
- ◆ Специальные формулы поиска данных
- ◆ Резюме

В этой главе мы обсудим различные методы, используемые для поиска значений в таблицах. Для выполнения этой задачи приложение Excel предоставляет пользователю три функции: ПРОСМОТР, ВПР и ГПР. Несмотря на то, что именно эти функции рассматриваются как основные функции поиска информации, они все же не достаточно универсальны. В этой главе приведено множество примеров поиска информации как стандартными, поддерживаемыми обычными средствами Excel, так и другими более специализированными инструментами поиска данных.

Знакомство с формулой поиска данных

Действительно, формула поиска возвращает значение из таблицы, отыскивая при этом другое, соответствующее ему значение. Хорошим примером, в данном случае, служит обычный телефонный справочник. Например, если вы хотите найти номер телефона человека, вы сначала ищите имя и фамилию этого человека, а затем записываете соответствующий номер телефона.

	A	B	C	D	E	F	G
1			Last Name	First Name	Department	Extension	Date Hired
2	Enter a Name ->	=ПРОСМОТР(C2;A9:F18;E9)	Davis	Rita	Administration	4441	09/19/98
3							
4							
5							
6							
7							
8							
9			Last Name	First Name	Department	Extension	Date Hired
10			Allen	Yolanda	Sales	4465	03/05/98
11			Baker	Nancy	Operations	4498	11/24/98
12			Bunnel	Ken	Marketing	4422	05/12/98
13			Charles	Larry	Administration	3968	06/00/97
14			Cramden	Oliver	Administration	4421	01/04/01
15			Davis	Rita	Administration	4441	09/19/98
16			Dunwell	James	Operations	3321	04/16/99
17			Ellis	Pamela	Data Processing	3398	02/01/00
18			Endow	Ed	Data Processing	4448	02/01/99

Рис. 8.1. Формулы поиска, введенные в строку 2, возвращают необходимую информацию служащего, имя которого введено в ячейку C2

Обратите внимание, на рис. 8.1 отображен простейший рабочий лист, который содержит несколько формул поиска и таблицу, содержащую данные о служащем

(EmpData), начиная с 9 строки. Если в ячейку C2 ввести интересующее вас имя, формулы поиска, введенные в диапазон ячеек D2 :G2, отыщет в таблице необходимую информацию. Приведенные ниже формулы используют функцию поиска ВПР:

D2 =ВПР (C2 ; EmpData ; 2 ; ЛОЖЬ)
E2 =ВПР (C2 ; EmpData ; 3 ; ЛОЖЬ)
F2 =ВПР (C2 ; EmpData ; 4 ; ЛОЖЬ)
G2 =ВПР (C2 ; EmpData ; 5 ; ЛОЖЬ)

В этом примере, для того чтобы вернуть информацию из диапазона EmpData, используется четыре формулы. Однако, как правило, пользователю необходимо вернуть только одно значение таблицы, поэтому достаточно часто используется только одна формула.

Функции, используемые для поиска данных

Для написания формул поиска информации в таблице данных используются следующие функции Excel (табл. 8.1):

Таблица 8.1. Функции, используемые для поиска данных в таблице

Функция	Описание функции
ВЫБОР	Возвращает определенное значение из списка значений (до 29), введенных в качестве аргументов.
ГПР	Функция горизонтального просмотра. Просматривает верхнюю строку таблицы и возвращает значение из указанной ячейки.
ИНДЕКС	Использует индекс для выбора значения из таблицы или массива.
ПРОСМОТР	Возвращает значение из диапазона, состоящего из одного столбца или одной строки.
ПОИСКПОЗ	Возвращает относительное положение элемента диапазона, который соответствует указанному значению в указанном порядке.
СМЕЩ	Возвращает ссылку на диапазон, смещенную относительно заданной ячейки или диапазона на определенное количество строк и столбцов.
ВПР	Вертикальный поиск. Ищет значение в первом столбце массива и возвращает значение из ячейки в найденной строке и указанном столбце.

Все примеры, приведенные в этой главе, используют функции, перечисленные в табл. 8.1.

Основные формулы поиска данных

Базовые функции поиска данных используются, в основном, для поиска определенных строк и столбцов таблицы и, в результате, возвращают другое, соответствующее им значение таблицы. Приложение Excel предоставляет в распоряжение пользователя три базовых функции просмотра: ГПР, ВПР и ПРОСМОТР. Функции ПОИСКПОЗ и ИНДЕКС часто используются совместно с ними и возвращают ячейку или ссылку на ячейку с искомым значением.

Функция ВПР

Функция ВПР осуществляет поиск значения в первом столбце просматриваемой таблицы и возвращает значение из ячейки, расположенной на пересечении найденной строки и ука-

занного столбца таблицы. Данная функция осуществляет вертикальный поиск по таблице. Синтаксис функции ВПР следующий:

ВПР (искомое_значение; массив_табл; номер_индекса_столбца; диапазон_просмотра)

Функция ВПР использует следующие аргументы:

- ◆ **искомое_значение** — значение, которое необходимо найти в первом столбце таблицы (значение, ссылка или строка).
- ◆ **массив_табл** — диапазон, содержащий таблицу, в которой осуществляется поиск данных.
- ◆ **номер_индекса_столбца** — номер столбца в таблице, в котором необходимо найти соответствующее значение.
- ◆ **диапазон_просмотра** — необязательный аргумент. В том случае, если данный аргумент содержит значение ИСТИНА или опущен, функция возвращает значение, найденное путем приближенного сопоставления искомого значения (если не найдено точное значение, функция возвращает ближайшее самое большое значение, но меньшее, чем заданный аргумент **искомое_значение**). В случае, если аргумент содержит логическое значение ЛОЖЬ, функция будет искать точное соответствие. Если точное соответствие не найдено, функция возвращает ошибку #Н/Д.



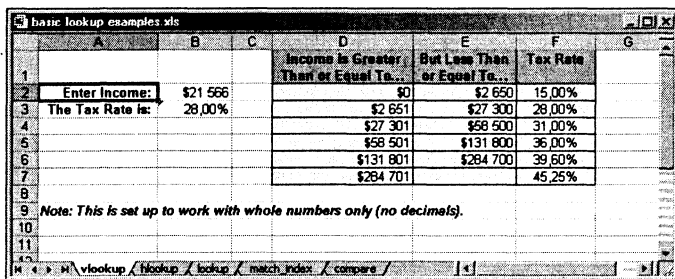
Если аргумент **диапазон_просмотра** содержит значение ИСТИНА или опущен, таблица будет отсортирована в порядке возрастания. В случае, если аргумент **искомое_значение** меньше самого меньшего значения, содержащегося в первом столбце диапазона **массив_табл**, функция ВПР возвращает ошибку #Н/Д. Если аргумент **диапазон_просмотра** содержит значение ЛОЖЬ, первый столбец таблицы поиска не требует сортировки по возрастанию. Тогда, если точное соответствие не найдено, функция возвращает ошибку #Н/Д.



Несмотря на то, что эта информация не включена в справочные сведения о функции, имейте в виду, что, если аргумент **искомое_значение** содержит текстовое значение, в нем можно использовать групповые символы * и ?.

Классическим примером применения формулы поиска может служить таблица с процентными ставками подоходного налога (рис. 8.2). В шкалу ставок подоходного налога внесены ставки налога, каждая из которых соответствует определенному уровню дохода служащего. Следующая формула, введенная в ячейку В3, возвращает ставку подоходного налога для уровня дохода в ячейке В2:

=ВПР (В2 ; D2 : F7 ; 3)



	A	B	C	D	E	F	G
				Income Is Greater Than or Equal To...	But Less Than or Equal To...	Tax Rate	
1							
2	Enter Income:	\$21 566		\$0	\$2 650	15,00%	
3	The Tax Rate is:	28,00%		\$2 651	\$27 300	28,00%	
4				\$27 301	\$58 500	31,00%	
5				\$58 501	\$131 800	36,00%	
6				\$131 801	\$284 700	39,60%	
7				\$284 701		45,25%	
8							
9	Note: This is set up to work with whole numbers only (no decimals).						
10							
11							

Рис. 8.2. Чтобы найти ставку налога, воспользуйтесь функцией ВПР

Обратите внимание, анализируемая таблица находится в диапазоне, состоящем из трех столбцов (D2:F7). Последний аргумент функции ВПР — 3, поэтому формула возвращает соответствующее значение, содержащееся в третьем столбце таблицы.

В данном случае, точного соответствия не требуется. Если точное соответствие не найдено в первом столбце просматриваемой таблицы, функция ВПР будет использовать ближайшее наибольшее значение, которое меньше искомого в таблице значения. Другими словами, функция использует строку, содержащую значение, которое меньше или равно искомому, а значение следующей за ней строки больше искомого значения. В таблице налоговых ставок — это вторая строка.

Функция ГПР

Функция ГПР работает по аналогии с функцией ВПР, за исключением того, что таблица просматривается не вертикально, а горизонтально. Функция просматривает верхнюю строку таблицы и возвращает значение, соответствующее указанной строке таблицы.

Синтаксис функции ГПР имеет следующий вид:

ГПР (искомое_значение ; таблица ; номер_строки ; диапазон_просмотра)

Функция ГПР использует следующие аргументы:

- ◆ **искомое_значение** — значение, которое требуется найти в первой строке таблицы.
- ◆ **таблица** — таблица, в которой производится поиск данных.
- ◆ **номер_строки** — номер строки в таблице, из которой возвращается сопоставляемое значение.
- ◆ **диапазон_просмотра** — необязательный аргумент. Если значение этого аргумента ИСТИНА или опущено, функция возвращает значение, найденное путем приближенного сопоставления искомого значения (если точное соответствие не будет найдено, функция возвратит ближайшее наибольшее значение, меньшее чем значение аргумента **искомое_значение**). В случае, если аргумент содержит логическое значение ЛОЖЬ, функция ГПР будет искать точное соответствие. Если точное соответствие не найдено, функция возвращает ошибку #Н/Д.



Эта информация не включена в интерактивную справку функции, однако, имейте в виду, что, если аргумент **искомое_значение** содержит текстовое значение, в нем можно использовать групповые символы * и ?.

На рис. 8.3 показана таблица со ставками подоходного налога, но уже в горизонтальной форме (E1:J3). Формула в ячейке B3 выглядит следующим образом:

=ГПР (B2 ; E1 : J3 ; 3)

	A	B	C	D	E	F	G	H	I	J	K
1				Income is Greater Than or Equal To...	\$0	\$2 651	\$27 301	\$58 501	\$131 801	\$284 701	
2	Enter Income:	\$21 566		Out Less Than...	\$2 650	\$27 300	\$58 500	\$131 800	\$284 700		
3	The Tax Rate is:	28,00%		Tax Rate	15,00%	28,00%	31,00%	36,00%	39,60%	45,25%	
4											
5											
6											

Рис. 8.3. В этом случае для поиска ставки подоходного налога можно воспользоваться функцией ГПР

Функция ПРОСМОТР

Функция ПРОСМОТР имеет следующий синтаксис:

ПРОСМОТР (искомое_значение; вектор_просмотра; вектор_результата)

Данная функция использует следующие аргументы:

- ◆ **искомое_значение** — значение, которое функция ищет в первом векторе.
- ◆ **вектор_просмотра** — диапазон, состоящий только из одной строки или одного столбца, содержащий значения, которые необходимо найти. Значения должны располагаться в порядке возрастания.
- ◆ **вектор_результата** — диапазон, состоящий только из одной строки или одного столбца и содержащий значения, которые будут возвращены. Размер этого диапазона должен равняться размеру диапазона вектор_просмотра.

Функция ПРОСМОТР просматривает диапазон, состоящий из одной строки или одного столбца (вектор_просмотра), находит необходимое значение (искомое_значение) и возвращает значение, содержащееся в соответствующей ячейке второго диапазона, который также состоит из одной строки или одного столбца (вектор_результата).



Имейте в виду, что значения диапазона вектор_просмотра должны располагаться в порядке возрастания. В том случае, когда значение **искомое_значение** меньше наименьшего значения, содержащегося в диапазоне вектор_просмотра, функция возвращает ошибку #Н/Д.



Справочная система Excel для функции ПРОСМОТР предлагает выбрать один из двух вариантов синтаксиса данной функции. Второй вариант синтаксиса функции ПРОСМОТР содержит аргумент **массив**. Данный синтаксис обеспечивает совместимость Excel с другими программными продуктами, поддерживающими электронные таблицы. Однако, в действительности, вместо этого синтаксиса предпочтительней использовать функции ВПР и ГПР.

Обратите внимание, на рис. 8.4 показана все та же таблица ставок подоходного налога, только на этот раз формула, введенная в ячейку В3, использует функцию ПРОСМОТР, чтобы вернуть соответствующую налоговую ставку:

=ПРОСМОТР (В2 ; D2 : D7 ; F2 : F7)



Имейте в виду, что, если значения, содержащиеся в первом столбце, не расположены в порядке возрастания, значение, возвращенное функцией ПРОСМОТР, может быть неправильным.

	A	B	C	D	E	F	G
1							
2	Enter Income:	\$123 409		Income is Greater Than or Equal To...	\$0	\$2 650	15.00%
3	The Tax Rate is:	36.00%			\$2 651	\$27 300	28.00%
4					\$27 301	\$58 500	31.00%
5					\$58 501	\$131 800	36.00%
6					\$131 801	\$284 700	39.60%
7					\$284 701		45.25%
8							
9							

Рис. 8.4. Чтобы найти соответствующую налоговую ставку, воспользуйтесь функцией ПРОСМОТР

В отличие от функции ВПР функция ПРОСМОТР использует два диапазона: первый диапазон содержит искомое значение, второй — результирующее значение, возвращенное функцией. В свою очередь, функция ВПР использует только диапазон таблицы поиска, а третий аргумент этой функции определяет номер столбца в таблице, в котором располагается результирующее значение. Конечно, этот аргумент может состоять из ссылки на ячейку.

Комбинирование функций ПОИСКПОЗ и ИНДЕКС

Достаточно часто, при выполнении задач поиска данных, функции ПОИСКПОЗ и ИНДЕКС используются вместе. Первая функция — функция ПОИСКПОЗ, возвращает относительную позицию ячейки в диапазоне, соответствующей заданному значению. Синтаксис функции следующий:

ПОИСКПОЗ (искомое_значение; массив_поиска; тип_сопоставления)

Функция ПОИСКПОЗ использует следующие аргументы:

- ◆ **искомое_значение** — значение, искомое в массиве массив_поиска. Если третий аргумент этой функции — тип_сопоставления, равен 0, а аргумент **искомое_значение** представляет собой текст, то он может содержать групповые символы * и ?
- ◆ **массив_поиска** — диапазон, который содержит искомое значение.
- ◆ **тип_сопоставления** — целое число в интервале от -1, 0, 1, которое устанавливает параметры соответствия



В том случае, если аргумент **тип_сопоставления** равен 1, функция ПОИСКПОЗ ищет самое большое значение, которое меньше, либо равно аргументу **искомое_значение** (значения диапазона массив_поиска должны располагаться в порядке возрастания). Если аргумент **тип_сопоставления** равен 0, функция ищет первое значение, в точности соответствующее аргументу **искомое_значение**. Если же аргумент **тип_сопоставления** равен -1, функция ищет наименьшее значение, которое больше или равно аргументу **искомое_значение** (в этом случае значения диапазона массив_поиска должны располагаться в порядке убывания). Если аргумент **тип_сопоставления** опускается, то по умолчанию он принимается равным 1.

Функция ИНДЕКС возвращает ячейку диапазона. Синтаксис функции следующий :

ИНДЕКС (массив; номер_строки; номер_столбца)

Функция ИНДЕКС использует следующие аргументы:

- ◆ **массив** — диапазон, являющийся массивом
- ◆ **номер_строки** — строка в массиве массив, из которого возвращается значение
- ◆ **номер_столбца** — столбец в массиве массив, из которого возвращается значение



В том случае, если массив содержит только одну строку или столбец, соответствующий аргумент номер_строки или номер_столбца использовать необязательно.

На рис. 8.5 показан рабочий лист, в котором в столбцах D, E и F содержатся даты, дни недели и суммы. При вводе даты в ячейку B1 формула, приведенная ниже (ячейка B2), просматривает значения в столбце D в поисках даты, а затем возвращает из столбца F соответствующую этой дате сумму:

=ИНДЕКС (F2:F21; ПОИСКПОЗ (B1;D2:D21; 0))

Date	Day	Amount
01.01.2001	понедельник	23
02.01.2001	Tuesday	179
03.01.2001	Wednesday	149
04.01.2001	Thursday	196
05.01.2001	Friday	131
06.01.2001	Monday	179
07.01.2001	Tuesday	134
08.01.2001	Wednesday	179
09.01.2001	Thursday	193
10.01.2001	Friday	191
11.01.2001	Monday	176
12.01.2001	Tuesday	189
13.01.2001	Wednesday	163
14.01.2001	Thursday	121
15.01.2001	Friday	100
16.01.2001	Monday	109
17.01.2001	Tuesday	151
18.01.2001	Wednesday	138
19.01.2001	Thursday	114
20.01.2001	Friday	156

Рис. 8.5. Функции ИНДЕКС и ПОИСКПОЗ используются для поиска данных в таблице

Когда пустая ячейка не равна нулю

Достаточно часто может возникнуть ситуация, когда ячейки результирующего диапазона не содержат никаких значений. Функции поиска Excel обрабатывают такие ячейки как нулевые значения. Например, рабочий лист в приведенном ниже рисунке, содержит таблицу поиска, состоящую из двух столбцов. Следующая формула просматривает первый столбец, отыскивая в нем имя, введенное в ячейку B1 и возвращает соответствующую этому имени сумму:

=ВПР (B1; D2: E8; 2)

Обратите внимание, что ячейка *Amount*, соответствующая имени *Charlie*, пуста, однако формула возвращает нулевое значение:

Name	Amount
Bob	45
Charlie	0
David	0
Frank	32
George	9
Harry	0
Mike	1

Если вы хотите, чтобы нулевые значения отличались от пустых ячеек, измените формулу просмотра, добавив в нее функцию ЕСЛИ, чтобы определить длину возвращаемого значения. В том случае, если возвращаемая ячейка пуста, длина значения воспринимается равной нулю, во всех других случаях — как ненулевое значение. Следующая формула отображает пустую строку (пробел) всякий раз, когда длина возвращаемого значения нулевая, и фактическое значение, когда длина значения не равна нулю:

```
=ЕСЛИ (ДЛСТР (ВПР (В1; D2 : E8 ; 2) ) = 0; " "; (ВПР (В1; D2 : E8 ; 2) ) )
```

Чтобы понять, как работает эта формула, начните с функции ПОИСКПОЗ. Данная функция ищет в диапазоне D2:D21 дату, содержащуюся в ячейке В1 и относительный номер строки, в которой эта дата найдена. Полученное значение используется в качестве второго аргумента функции ИНДЕКС. В результате формула возвращает соответствующее значение, содержащееся в диапазоне F2:F21.

Специальные формулы поиска данных

В дополнение к базовым формулам поиска данных приложение Excel содержит некоторые дополнительные типы формул, позволяющие осуществить специализированный поиск данных. К примеру, используя эти формулы, вы можете осуществлять поиск точного значения, искать значения в столбце, расположенном около первого столбца таблицы, выполнять поиск с учетом регистра символа, возвращать значение из множества таблиц поиска, а также проводить другие виды сложного поиска.

Поиск точного значения

Как было показано в предыдущих примерах, функции ВПР и ГПР не требуют точного соответствия искомого значения значению, содержащемуся в таблице поиска. Однако это допустимо, например, при поиске ставки подоходного налога. В других ситуациях может потребоваться полное соответствие значений. К примеру, при поиске табельного номера служащего необходимо полное соответствие номера.

Для того чтобы найти только точно соответствующее значение, используются те же функции ВПР или ГПР, но с необязательным четвертым аргументом со значением ЛОЖЬ.

На рис. 8.6 показан рабочий лист с таблицей, которая содержит табельные номера (столбец С), а также имена и фамилии служащих (столбец D). Таблица поиска носит название *EmpList*. Следующая формула, введенная в ячейку В2, ищет табельный номер служащего (ячейка В1) и возвращает имя служащего, имеющего данный табельный номер:

```
=ВПР (В1; EmpList; 2; ЛОЖЬ)
```

Поскольку последний аргумент функции ВПР имеет значение ЛОЖЬ, функция возвращает значение только в том случае, если найдено точное соответствие. В противном случае, если значение не найдено, формула возвращает ошибку #Н/Д. Безусловно, поиск приближенного соответствия значения табельного номера служащего не имеет, в данном случае, никакого смысла. Кроме того, обратите внимание, что табельные номера служащих в столбце С расположены не в порядке возрастания, поскольку при использовании в качестве четвертого аргумента логического значения ЛОЖЬ, упорядочения по возрастанию не требуется.

A	B	C	D
Employee No.	999	Employee Number	Employee Name
Employee Name	#N/A	873	Charles K. Barkley
		1109	Francis Jenkins
		1549	James Brackman
		1334	Linda Harper
		1643	Louise Victor
		1101	Melinda Hindquest
		1673	Michael Orenthal
		963	Peter Yates
		972	Sally Rice
		1398	Walter Franklin

Рис. 8.6. Поиск в этой таблице требует точного соответствия значений



Если вы хотите, чтобы вместо ошибки #Н/Д на экране отображалась другая информация (в случае, если табельный номер служащего не найден), используйте функцию ЕСЛИ, предотвращая появление ошибки #Н/Д (используя функцию ЕНД) и заменяя ее на любую другую строку. Следующая формула вместо ошибки #Н/Д отображает текст "Не найдено":

=ЕСЛИ(ЕНД(ВПР(В1;EmpList;2;ЛОЖЬ));
"Не найдено";ВПР(В1;EmpList;2;ЛОЖЬ))

Поиск значения слева

Функция ВПР всегда ищет значение в первом столбце заданного диапазона. Но как быть, если пользователю необходимо найти значение не в первом, а в каком-либо другом столбце? Конечно, очень удобно назначать третьему аргументу функции ВПР отрицательное значение. Но это невозможно.

Давайте рассмотрим эту проблему на простом примере (рис. 8.7). Предположим, что нам необходимо найти средний темп (столбец В, диапазон *Averages*) одного из игроков из столбца С (диапазон *Players*). Имя игрока, данные о котором необходимо найти, введено в ячейку *LookupValue*. В данном случае функция ВПР работать не будет, поскольку данные размещены не корректно. Конечно, данные можно перестроить с помощью одного действия, однако иногда и это невозможно.

A	B	C	D	E	F	G	H	I
At Bats	Average	Player		Player to lookup:	Darr			
12	0.333	Arias						
41	0.390	Darr		Average	0.390	<--	LOOKUP	
24	0.333	Davis		At Bats:	41	<--	LOOKUP	
25	0.160	Gomez						
23	0.217	Gonzalez		Average	0.390	<--	INDEX and MATCH	
30	0.300	Gwynn		At Bats:	41	<--	INDEX and MATCH	
0	0.000	Henderson						
51	0.333	Jackson						
43	0.186	Klesko						
36	0.139	Kotsay						
9	0.333	Magadan						
16	0.313	Mendez						
44	0.341	Nevin						
14	0.286	Perez						
28	0.321	Trammell						

Рис. 8.7. Функция ВПР не позволяет осуществить поиск значения в столбце В на основе данных столбца С

Один из способов решения этой проблемы — использовать функцию ПРОСМОТР, которая требует в качестве аргументов два диапазона. Следующая формула (в ячейке F3) возвращает средний темп игрока (из столбца B), имя которого введено в ячейку LookupValue:

=ПРОСМОТР (LookupValue; Players; Averages)

Для того чтобы использовать функцию ВПР, диапазон поиска (в данном случае диапазон Players) необходимо упорядочить в порядке возрастания. В дополнение к этому ограничению, формула имеет еще одну небольшую проблему: при вводе имени несуществующего игрока (другими словами, если ячейка LookupValue содержит значение, которое отсутствует в диапазоне Players), результат, возвращаемый формулой, будет ошибочным.

Более эффективный способ — совместное использование функций ИНДЕКС и ПОИСКПОЗ. Следующая формула работает точно так же, как предыдущая, за исключением того, что она возвращает ошибку #Н/Д, если искомый игрок не найден. Другое преимущество этой формулы состоит в том, что имена игроков не требуют сортировки:

=ИНДЕКС (Averages; ПОИСКПОЗ (LookupValue; Players; 0))

Поиск с учетом регистра

Давайте рассмотрим задачу поиска текста в таблице данных. Базовые функции поиска данных в Excel (ПРОСМОТР, ВПР и ГПР) не учитывают регистр символов. Поэтому, если вы используете эти функции, чтобы найти в таблице текст *бюджет*, формула будет возвращать любую найденную, соответствующую этому тексту строку, например: *БЮДЖЕТ*, *Бюджет* или *БюдЖет*.

Обратите внимание, рабочий лист, отображенный на рис. 8.8, содержит два диапазона: первый диапазон включает ячейки D2:D7 (*Range1*), второй — ячейки E2:E7 (*Range2*). Текст, который необходимо найти, содержится в ячейке B1 (*Value*).

Следующая формула массива, введенная в ячейку B2, осуществляет поиск заданного значения в диапазоне *Range1*, учитывая при этом регистр символов, и возвращает соответствующее значение из *Range2*.

Рис. 8.8. Чтобы найти текст, учитывая регистр символов, воспользуйтесь формулой массива

{=ИНДЕКС (Range2; ПОИСКПОЗ (ИСТИНА; СОВПАД (Value; Range1); 0))}

Данная формула находит слово DOG (все символы верхнего регистра) и возвращает соответствующее значение 300. В свою очередь, стандартная формула поиска, использующая функцию ПРОСМОТР, возвращает значение 400:

=ПРОСМОТР (Value; Range1; Range2)



Помните, что при вводе формулы массива необходимо использовать клавиши <Ctrl+Alt+Enter>.

Поиск значения в множестве таблиц

Действительно, рабочий лист может содержать любое количество таблиц поиска данных. В таких случаях в форму необходимо ввести информацию о том, в какой именно таблице

должен осуществляться поиск. Обратите внимание, на рис. 8.9 приведен пример использования множества таблиц на одном рабочем листе.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Sales Rep	Years	Sales	Comm. Rate	Commission		<3 Years Tenure		3+ Years Tenure			
2	Benson	2	120 000	7,00%	8 400		Amt Sold	Rate	Amt Sold	Rate		
3	Davidson	1	210 921	7,00%	14 764		0	1,50%	0	2,00%		
4	Ellison	1	100 000	7,00%	7 000		5 000	3,25%	50 000	6,25%		
5	Gomez	2	87 401	6,00%	5 244		10 000	3,50%	100 000	7,25%		
6	Hernandez	6	310 983	9,25%	28 766		20 000	5,00%	200 000	8,25%		
7	Kelly	3	43 902	2,00%	878		50 000	6,00%	300 000	9,25%		
8	Martin	2	121 021	7,00%	8 471		100 000	7,00%	500 000	10,00%		
9	Oswald	3	908	2,00%	18		250 000	8,00%				
10	Reginald	1	0	1,50%	0							
11	Veras	4	359 832	9,25%	33 284							
12	Wilmington	4	502 983	10,00%	50 298							
13												
14												
15												
16												

Рис. 8.9. Данный рабочий лист содержит множество таблиц поиска данных

Данная рабочая книга содержит два диапазона: диапазон G3:H9 (Table1) и диапазон J3:K8 (Table2) и вычисляет комиссионные вознаграждения продавцов фирмы. Ставка комиссионного вознаграждения каждого отдельного продавца зависит от двух факторов: срока работы продавца в данной фирме (столбец B) и объема сделанным им продаж (столбец C). Столбец D содержит формулы поиска ставки комиссионного вознаграждения в соответствующей таблице. Например, ячейка D2 содержит следующую формулу:

=ВПР(C2;ЕСЛИ(B2<3;Table1;Table2);2)

Второй аргумент функции ВПР содержит функцию ЕСЛИ, которая использует значение ячейки B2 для того, чтобы определить, какую из таблиц следует просматривать.

Формула в столбце E просто умножает объем продаж каждого продавца в столбце C на ставку комиссионного вознаграждения из столбца D. Ячейка E2 содержит следующую формулу:

=C2*D2

Определение оценки по количеству набранных баллов

Еще один типичный пример использования таблиц поиска данных — это выставление экзаменационных оценок по результатам тестирования студентов. На рис. 8.10 показан рабочий лист, который содержит таблицу с количеством набранных баллов и соответствующих категории (диапазон E2:F6 с именем *GradeList*), которая используется для присвоения студенту одной из оценок.

	A	B	C	D	E	F	G
1	Student	Score	Grade		Score	Grade	
2	Adams	36	F		0	F	
3	Baker	68	D		40	D	
4	Camden	50	D		70	C	
5	Dailey	77	C		80	B	
6	Gomez	92	A		90	A	
7	Hernandez	100	A				
8	Jackson	74	C				
9	Maplethorpe	45	D				
10	Paulson	60	D				
11	Ramirez	89	B				
12	Sosa	99	A				
13	Thompson	91	A				
14	Wilson	59	D				
15							

Рис. 8.10. Назначение категории оценки по результатам тестирования

Столбец C содержит формулы, которые используют функцию ВПР и таблицу поиска, чтобы на основании полученного каждым студентом количества баллов (столбец B) присвоить студенту соответствующую оценку. Например, ячейка C2 содержит следующую формулу:

=ВПР (B2;GradeList;2)

В том случае, когда таблица поиска содержит небольшое количество значений (как, например, на рис. 8.10), вместо нее можно использовать массивы символов. Например, приведенная ниже формула возвращает категории оценок, не используя при этом таблицу поиска. Информация таблицы поиска введена непосредственно в формулу массива. Для получения дополнительной информации о массивах обратитесь к главе 14.

=ВПР (B2; {0, "F"; 40, "D"; 70, "C"; 80, "B"; 80, "A"}; 2)

Другая, более удобная формула назначения категорий оценок, заключается в использовании функции ПРОСМОТР с двумя аргументами, представляющими собой массивы:

=ПРОСМОТР (B2; {0; 40; 70; 80; 90}; {"F"; "D"; "C"; "B"; "A"})

Вычисление среднего уровня оценок

Средний уровень оценок студентов в данном примере представляет собой числовое представление средней оценки, полученной студентами. Для вычисления этого показателя каждому буквенному символу оценки назначается определенное числовое значение (A=4, B=3, C=2, D=1 и F=0). Средний уровень оценки рассчитывается как средняя величина числовых значений оценок, взвешенная на количество часов, пройденных по данному курсу. К примеру, одночасовой курс встречается реже, чем трехчасовой. Градация показателя среднего уровня оценок начинается с 0 (F) и заканчивается 4,00 (A).

На рис. 8.11 показан рабочий лист, содержащий информацию о студенте. Этот студент прошел пять курсов, общее количество прослушанных им часов — 13 (диапазон B2:B6 — *CreditHours*). Оценки, полученные за каждый курс, содержатся в столбце C (диапазон C2:C6 — *Grades*). Столбец D содержит формулу поиска, которая вычисляет числовое значение оценки для каждого курса. Например, следующая формула (в ячейке D2) использует таблицу поиска диапазона G2:H6 (*GradeTable*).

=ВПР (C2;GradeTable;2;ЛОЖЬ)

1	Course	Credit Hrs	Grade	Grade Val	Weighted Val	Grade Table
2	Psych 101	3	A	4	12	A 4
3	PhysEd	2	C	2	4	B 3
4	PolSci 101	4	B	3	12	C 2
5	IndepStudy	1	A	4	4	D 1
6	IntroMath	3	A	4	12	F 0
7	GPA: 3.38 <- Requires multiple formulas and lookup table					
8						
9						
10						

Рис. 8.11. Для вычисления показателя среднего уровня оценки вы можете использовать множество формул

Формулы, введенные в столбец Е, вычисляют взвешенные значения. Например, формула в ячейке Е2 выглядит следующим образом:

=D2 * B2

И, наконец, следующая формула в ячейке В8 вычисляет показатель среднего уровня оценки: =СУММ(Е2 : Е6) / СУММ(В2 : В6)

Несмотря на то, что предыдущие формулы работают достаточно хорошо, вы можете несколько упростить вычисление среднего показателя. Чтобы выполнить это вычисление, можно использовать единственную формулу массива, не прибегая к использованию таблицы поиска и формул в столбцах D и E. Эта формула выглядит следующим образом:

{=СУММ((ПОИСКПОЗ(Grades; {"F"; "D"; "C"; "B"; "A"}; 0) - 1) * CreditHours) / СУММ(CreditHours) }

Поиск на пересечении строк и столбцов

На рис. 8.12 показан рабочий лист, который содержит данные о ежемесячных объемах продаж товаров компаний. Для того чтобы найти информацию об объеме продаж определенного товара в течение одного заданного месяца, в ячейку В1 необходимо ввести искомый месяц, а в ячейку В2 название товара.

1	Month:	July		Widgets	Sprockets	Snapshythes	Combined
2	Product:	Sprockets	January	2 892	1 771	4 718	9 381
3			February	3 360	4 711	2 615	10 706
4	Month Offset:	8	March	3 744	3 223	5 312	12 279
5	Product Offset:	3	April	3 221	2 438	1 108	6 767
6	Sales:	3 337	May	4 839	1 999	1 994	8 832
7			June	3 767	5 140	3 830	12 737
8			July	5 467	3 337	3 232	12 036
9	Single-formula ->	3 337	August	3 154	4 895	1 607	9 656
10			September	1 718	2 040	1 563	5 321
11			October	1 548	1 061	2 590	5 199
12			November	5 083	3 558	3 960	12 601
13			December	5 753	2 839	3 013	11 605
14			Total	44 566	37 012	35 542	117 120
15							
16							
17							

Рис. 8.12. Таблица поиска информации на пересечении строк и столбцов

Для упрощения работы с формулами на данном рабочем листе используются следующие диапазоны ячеек:

Имя диапазона	Диапазон
Month	B1
Product	B2
Table	D1:H14
MonthList	D1:D14
ProductList	D1:H1

Следующая формула (ячейка B4) использует функцию ПОИСКПОЗ и возвращает позицию заданного месяца в диапазоне ячеек MonthList. К примеру, если вы ищете месяц *January*, то данная формула возвращает значение 2, поскольку январь — это второй элемент диапазона MonthList (первый элемент — незаполненная ячейка D1).

=ПОИСКПОЗ (Month;MonthList;0)

Формула, введенная в ячейку B5, работает тем же образом, но применяется для диапазона ProductList.

=ПОИСКПОЗ (Product;ProductList;0)

И, наконец, формула в ячейке B6 возвращает соответствующий заданным условиям объем продаж. Формула использует функцию ИНДЕКС и, в качестве аргументов, результаты, полученные в ячейках B4 и B5.

=ИНДЕКС (Table;B4;B5)

Безусловно, все приведенные формулы можно объединить в одну, например:

=ИНДЕКС (Table;ПОИСКПОЗ (Month;MonthList;0);ПОИСКПОЗ (Product;ProductList;0))



Если у вас установлена Excel 97 или более поздняя версия программы, то можете использовать надстройку Мастер подстановок. Мастер подстановок помогает создать формулу для поиска значения, находящегося на пересечении строки и столбца (рис. 8.13).



Еще один способ быстрого создания таблицы для поиска значения на пересечении строк и столбцов предусматривает присвоение имен каждой строке и каждому столбцу таблицы. Чтобы сделать это, выделите таблицу с данными и выберите команду Вставка⇒Name⇒Create. После создания имен приведенные формулы выглядят гораздо проще, например:

=Колеса Июль

Данная формула использует пересечение двух диапазонов и возвращает данные о продажах колес в июле. Подробнее об этом см. в главе 3.

Поиск информации в двух столбцах

В отдельных случаях таблица поиска может содержать два столбца с искомыми данными. На рис. 8.14 приведен пример такой таблицы.

Таблица поиска содержит список автомобилей, для каждого из которых указаны марка, модель и номер. Диапазоны рабочего листа названы следующим образом:

F2:F12	Code
B1	Make
B2	Model
D2:D12	Range1
E2:E12	Range2

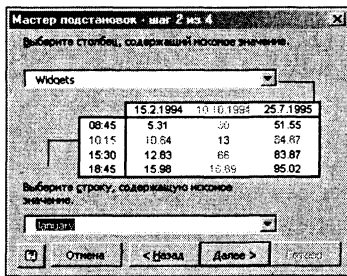


Рис. 8.13. Мастер подстановок создает формулы для поиска значения, находящегося на пересечении строк и столбцов

	A	B	C	D	E	F
1	Make:	Jeep	Make	Model	Code	
2	Model:	Grand Cherokee	Chev	Blazer	C-084	
3	Code:	J-701	Chev	Tahoe	C-823	
4			Ford	Explorer	F-772	
5			Ford	Expedition	F-229	
6			Isuzu	Rodeo	I-887	
7			Isuzu	Trooper	I-900	
8			Jeep	Cherokee	J-983	
9			Jeep	Grand Cherokee	J-701	
10			Nissan	Pathfinder	N-231	
11			Toyota	4Runner	T-871	
12			Toyota	Land Cruiser	T-981	
13						

Рис. 8.14. В данной рабочей книге выполняется поиск информации, на основе данных двух столбцов таблицы (D и E)

Следующая формула массива выводит на экран номер, соответствующий автомобилю определенной марки и модели:

```
{=ИНДЕКС (Code; ПОИСКПОЗ (Make&Model; Range1&Range2"; 0) ) }
```

Данная формула связывает текстовые строки, содержащиеся в ячейках Make и Model, а затем отыскивает текст в соответствующем объединении двух диапазонов: Range1 и Range2.

Определение расположения значения в диапазоне

В большинстве случаев, формулы просмотра данных таблицы используются для поиска определенных значений. Однако, иногда, пользователю необходимо определить расположение определенного значения или ячейки, содержащей это значение. К примеру, на рис. 8.15 показан рабочий лист, содержащий диапазон данных, состоящий всего лишь из одного столбца (Data). Ячейка B1 содержит значение, которое необходимо найти (Target).

Следующая формула, введенная в ячейку B2, возвращает адрес ячейки из диапазона Data, которая содержит значение Target:

```
=АДРЕС (СТРОКА (Data) +ПОИСКПОЗ (target; Data; 0) -1; СТОЛБЕЦ (Data) )
```

Если диапазон Data содержит только одну строку, для поиска адреса значения воспользуйтесь следующей формулой:

```
=АДРЕС (СТРОКА (Data) ; СТОЛБЕЦ (Data) +ПОИСКПОЗ (Target; Data; 0) -1)
```

Если диапазон Data содержит более одного экземпляра искомого значения, формулы возвратят первую найденную ячейку с этим значением. Если искомое значение не найдено, формулы возвратят ошибку #N/D.

	A	B	C	D
1	Target:	24	Data	
2	Address:	\$C\$9	74	
3			62	
4			60	
5			44	
6			50	
7			41	
8			77	
9			24	
10			55	
11			30	
12			12	
13			21	
14			7	
15			1	
16			22	
17			53	
18			36	
19			18	
20			68	
21				

Рис. 8.15. Формула в ячейке B2 возвращает адрес ячейки в диапазоне Data, в которой содержится значение, введенное в ячейке B1

Поиск значения, максимально приближенного к заданному

Формулы ПРОСМОТР и ГПР могут быть весьма полезны в следующих ситуациях:

- ◆ Если необходимо найти значение, в точности соответствующее контрольному значению. Для этого в качестве четвертого аргумента функций используется логическое значение ЛОЖЬ.
- ◆ Если необходимо найти значение максимально, приближенное к искомому. Для этого в качестве четвертого аргумента функций используется логическое значение ИСТИНА или аргумент опускается, а точное соответствие не ищется вообще; формула возвращает ближайшее к искомому значение, но в меньшую сторону.

Что делать, если пользователю необходимо найти значение, максимально соответствующее искомому, независимо от направления отклонения? Ни одна из предложенных нами базовых формул (ПРОСМОТР и ГПР) не сможет выполнить эту задачу.

На рис. 8.16 показан рабочий лист, содержащий имена студентов (столбец A) и некоторые значения (столбец B). Диапазон B2 : B20 носит название *Data*, ячейка *Target* содержит значение, которое необходимо найти в диапазоне *Data*. Ячейка E3 (*ColOffset*) содержит значение, которое определяет, на сколько столбцов и в какую сторону от диапазона *Data* происходит смещение при поиске возвращаемого значения.

Приведенная ниже формула массива определяет самое ближайшее значение диапазона *Data*, соответствующее значению *Target*, и возвращает имя студента в столбце A (т.е. столбец, смещенный на -1 от столбца с диапазоном *Data*). Формула возвращает имя Leslie, которому соответствует значение 8 000 — самое ближайшее значение к *Target*=8 025.

Student	Data	Target Value ->	Column Offset ->
Ann	9,101	8025	-1
Betsy	8,873		
Chuck	6,000		
David	9,820		
George	10,500		
Hilda	3,500		
James	12,873		
John	5,867		
Keith	8,989		
Leslie	8,000		
Michelle	1,124		
Nora	9,099		
Paul	6,800		
Peter	5,509		
Rasmusen	5,460		
Sally	8,400		
Theresa	7,777		
Violet	3,600		
Wendy	5,400		

Target Value -> 8025
Column Offset -> -1
Student: Leslie

Рис. 8.16. Данная рабочая книга демонстрирует выполнение поиска значения, ближайшего к заданному значению

```
{=ДВССЫЛ(АДРЕС(СТРОКА(Data)+ПОИСКПОЗ(МИН(ABS(Target-Data));ABS(Target-Data);0)-1;СТОЛБЕЦ(Data)+ColOffset))}
```

Если в диапазоне *Data* формула находит два значения, которые одинаково близки к значению *Target*, формула возвращает первое значение, найденное в списке.

Значение, содержащееся в ячейке ColOffset, может быть отрицательным (если столбец расположен слева от диапазона Data), положительным (если столбец расположен справа от диапазона Data) или равным 0 (если ближайшее значение расположено в этом же диапазоне Data).

Чтобы понять, как работает эта формула, вы должны уяснить себе принцип работы функции ДВССЫЛ. В качестве первого аргумента этой функции используется текстовая строка в виде ссылки на ячейку (или ссылки на ячейку, содержащую текстовую строку). В нашем примере текстовая строка создается с помощью функции АДРЕС, которая принимает номера строки и столбца, а возвращает адрес ячейки.

Поиск значения с использованием линейной интерполяции

Интерполяция представляет собой процесс создания недостающих значений на основе существующего ряда значений. Давайте рассмотрим пример интерполяции данных. На рис. 8.17 показан рабочий лист, в котором столбец D содержит список значений x , а столбец E — список значений y , соответствующих каждому значению x .

Кроме того, рабочий лист содержит график, который описывает связь между диапазонами x и y визуально. Расположение точек на диаграмме говорит о том, что зависимость между значениями x и y линейная: рост значения x вызывает пропорциональный рост значения y . Обратите внимание, что значения в диапазоне x введены не в строгой последовательности. Например, отсутствуют значения 3, 6, 7, 14, 17, 18 и 19.

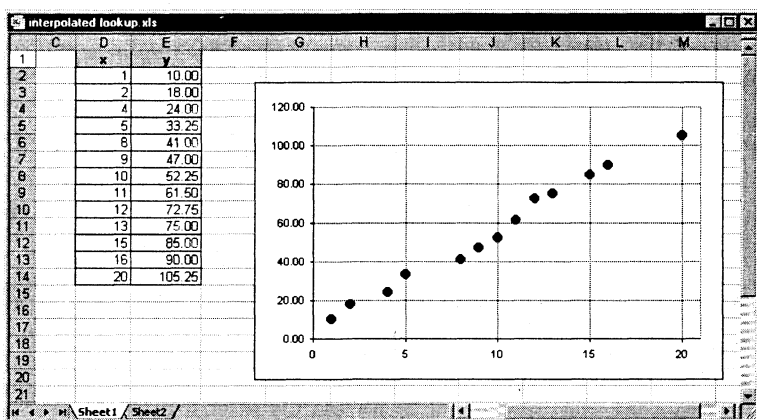


Рис. 8.17. Данная рабочая книга демонстрирует выполнение поиска значения с помощью интерполяции

Вы уже умеете создавать формулы, которые, просматривая значения диапазона x , будут возвращать значения диапазона y . Но как быть, если необходимо рассчитать значение y для отсутствующего значения x ? Обычная формула таблицы поиска возвращает неправильный результат, поскольку оперирует только существующими значениями y (вместо расчетного). Например, следующая формула ищет значение 3, но возвращает значение 18,00 (как значение, соответствующее значению 2 в диапазоне x).

=ПРОСМОТР(3; x; y)

Именно в таких случаях используется интерполяция значений. Другими словами, поскольку искомое значение (3) находится посередине существующих значений x (2 и 4), воз-

вращаемое значение у должно равняться 21,00, то есть также располагаться посередине между значениями 18,00 и 24,00.

Формулы линейной интерполяции

На рис. 8.18 показан рабочий лист с формулами, введенными в столбец В. Искомое значение введено в ячейку В1, а итоговая формула в ячейке В16 возвращает результат подсчетов. В том случае, если значение, содержащееся в ячейке В3, найдено в диапазоне x , формула возвращает соответствующее значение y . В противном случае, если значение в В3 не найдено, формула в ячейке В16 возвращает расчетное значение y , полученное путем линейной интерполяции.

	A	B	C	D	E
1	X-value to look up:	3		k	y
2				1	10.00
3	Matching row:	2		2	16.00
4	Exact match?	FALSE		4	24.00
5				5	33.25
6	1st row:	2		8	41.00
7	2nd row:	3		9	47.00
8				10	52.25
9	1st x value:	2		11	61.50
10	2nd x value:	4		12	72.75
11				13	75.00
12	1st looked up y value:	16		15	65.00
13	2nd looked up y value:	24		16	90.00
14				20	105.25
15	Adjustment factor:	0.5			
16	Interpolated lookup:	21.00			

Рис. 8.18. Столбец В содержит формулы, которые осуществляют поиск данных с помощью линейной интерполяции

Это может быть весьма неудобно, если значения в диапазоне x расположены в порядке возрастания, поскольку, если ячейка В1 содержит значение меньше, чем самое низкое значение x , или большее, чем самое большое значение, формула возвращает значение ошибки. Формулы, используемые для получения значений методом интерполяции приведены в табл. 8.2.

Таблица 8.2. Формулы поиска значений методом интерполяции

Ячейка	Формула	Описание
В3	=ПРОСМОТР (В1 ; х ; у)	Выполняет обычный поиск и возвращает значение диапазона x .
В4	=В1=В3	Возвращает значение ИСТИНА, если найденное значение соответствует искомому.
В6	=ПОИСКПОЗ (В3 ; х ; 0)	Возвращает номер строки диапазона x , которая содержит совпадающее значение.
В7	=ЕСЛИ (В4 ; в6 ; В6+1)	Возвращает ту же строку, что и формула в ячейке В6, если найдено значение, в точности соответствующее заданному. В противном случае, к результату ячейки В6 добавляется 1.
В9	=ИНДЕКС (х ; В6)	Возвращает соответствующее значение x для строки, полученной в ячейке В6.
В10	=ИНДЕКС (х ; В7)	Возвращает соответствующее значение x для строки, полученной в ячейке В7.

Ячейка	Формула	Описание
B12	=ПРОСМОТР (B9 ; x ; y)	Возвращает значение y, соответствующее значению x, полученному в ячейке B9.
B13	=ПРОСМОТР (B10 ; x ; y)	Возвращает значение y, соответствующее значению x, полученному в ячейке B10.
B15	=ЕСЛИ (B4 ; 0 ; (B1-B3) / (B10-B9))	Вычисляет корректирующий фактор, основанный на разнице между значениями x.
B16	=B12+ ((B13-B12) *B15)	Вычисляет расчетное значение y с помощью корректирующего фактора, полученного в ячейке B15.

Комбинирование функций ПРОСМОТР и ТЕНДЕНЦИЯ

Существует еще один, более предпочтительный способ поиска данных в таблице с помощью линейной интерполяции. Этот способ основан на использовании функций ПРОСМОТР и ТЕНДЕНЦИЯ и имеет одно большое преимущество — он требует использования только одной формулы (рис. 8.19).

	A	B	C	D	E
1	X-value to look up:	3		1	10.00
2				2	18.00
3	y value:	20.27913834		4	24.00
4				5	33.25
5				8	41.00
6				9	47.00
7				10	52.25
8				11	61.50
9				12	72.75
10				13	75.00
11				15	85.00
12				16	90.00
13				20	105.25
14					
15					
16					

Рис. 8.19. Данный рабочий лист содержит формулу, которая использует функции ПРОСМОТР и ТЕНДЕНЦИЯ

Формула, приведенная ниже (в ячейке B3), для принятия решения использует функцию ЕСЛИ. В том случае, если в диапазоне x найдено значение, в точности соответствующее искомому, формула возвращает соответствующее значение из диапазона y (использование функции ПРОСМОТР). Если точное соответствие не найдено, формула использует функцию ТЕНДЕНЦИЯ и возвращает значение y, рассчитанное методом максимально близкой величины (не прибегая к линейной интерполяции).

=ЕСЛИ (B1=ПРОСМОТР (B1 ; x ; x) ; ПРОСМОТР (ИНДЕСК (x ; ПОИСКПОЗ (ПРОСМОТР (B1 ; x ; x) ; x ; 0)) ; x ; y) ; ТЕНДЕНЦИЯ (y ; x ; B1))

Резюме

В этой главе мы вкратце обсудили функции, которые позволяют осуществить поиск информации в таблицах Excel. В данной главе рассматривалось множество примеров использования как основных формул поиска данных, так и различных дополнительных формул.

В следующей главе вашему вниманию будут представлены формулы, очень полезные для сведения итоговой информации, содержащейся в базе данных.

Базы данных и списки

В этой главе...

- ◆ Знакомство со списками и базами данных электронной таблицы
- ◆ Использование средства Автофильтр
- ◆ Использование расширенного фильтра
- ◆ Определение условий расширенного фильтра
- ◆ Использование функций базы данных
- ◆ Суммирование списка с помощью таблицы подстановки
- ◆ Создание промежуточных итогов
- ◆ Резюме

База данных электронной таблицы (известная также как *список*) — представляет собой определенным образом организованную совокупность информации. Выражаясь более конкретным языком, она состоит из строки заголовков (описательного текста) и следующих по ней строк, содержащих числовые или текстовые значения. Эта глава предлагает краткий обзор возможностей базы данных электронной таблицы Excel, а также некоторые весьма полезные формулы, способные управлять даже самыми громоздкими базами данных приложения.



Имейте в виду, что термин *база данных* используется, в данном случае, не по назначению. База данных электронной таблицы Excel — это скорее одна из таблиц в стандартной базе данных. В отличие от обычной базы данных, Excel не позволяет устанавливать взаимоотношения между таблицами.

Знакомство со списками и базами данных электронной таблицы

На рис. 9.1 отображен пример использования списка (или базы данных) на рабочем листе. Каждый столбец списка содержит заголовок (строка 1) и соответствующую информацию (следующие 20 строк). Обратите внимание, что содержимое ячеек представляет собой различные типы данных: текст, числовые значения, значения даты и логические значения. Столбец C содержит формулу, которая, в зависимости от годового дохода в столбце B, вычисляет ежемесячный доход служащего.

Столбцы списка рабочего листа, как правило, называются *полями*, а строки — *записями*. Используя эту терминологию, можно сказать, что данный список (см. рис. 9.1) содержит шесть полей (Name, Annual Salary, Monthly Salary, Location, Date Hired, Exempt) и двадцать записей.

	A	B	C	D	E	F
1	Name	Annual Salary	Monthly Salary	Location	Date Hired	Exempt
2	Anthony Taylor	\$55,500	\$4,625	Los Angeles	5/2/1998	FALSE
3	Charles S. Billings	\$39,000	\$3,250	Los Angeles	11/1/2000	FALSE
4	Chris Poundsworth	\$29,850	\$2,488	Seattle	6/12/1997	FALSE
5	Clark Bickerson	\$120,000	\$10,000	Los Angeles	4/19/2000	TRUE
6	Douglas Williams	\$89,687	\$7,474	Portland	1/3/1998	TRUE
7	Ivan Silberstein	\$95,000	\$7,917	Los Angeles	12/1/1998	TRUE
8	James Millen	\$27,690	\$2,308	Los Angeles	3/16/1997	FALSE
9	Jeffrey P. Jones	\$42,000	\$3,500	Seattle	4/9/2000	FALSE
10	Joe Morrison	\$24,000	\$2,000	Seattle	3/13/1999	FALSE
11	John T. Foster	\$39,500	\$3,292	Seattle	10/10/1999	FALSE
12	Kurt Karmichoff	\$48,000	\$4,000	Los Angeles	11/5/2000	FALSE
13	Michael Hayden	\$78,230	\$6,519	Seattle	6/1/1999	TRUE
14	Phillip A. Todd	\$29,500	\$2,458	Portland	1/3/1998	FALSE
15	Richard E. Card	\$43,000	\$3,583	Seattle	7/2/1997	FALSE
16	Rick Fogerty	\$89,873	\$7,489	Portland	6/28/1998	TRUE
17	Robert H. Miller	\$149,000	\$12,417	Portland	5/1/1999	TRUE
18	Stephen C. Carter	\$44,123	\$3,677	Los Angeles	9/29/1998	FALSE
19	Steven H. Katz	\$32,900	\$2,742	Los Angeles	3/16/2000	FALSE
20	Thomas E. Abbott	\$60,000	\$5,000	Los Angeles	4/12/1997	FALSE
21	Tom Brown	\$65,000	\$5,417	Seattle	5/27/2000	TRUE
22						

Рис. 9.1. Типичный пример использования списка на рабочем листе

Максимальный размер списка, создаваемого в приложении Excel, ограничен размером одного рабочего листа. Он может содержать не более 256 полей и не более 65 535 записей (одна строка предназначена для имен полей). Список такого размера требует огромного количества памяти и, иногда, его даже невозможно отобразить. Возможна и другая крайность — список, состоящий из одной ячейки, несмотря на то, что он практически бесполезен, тем не менее, может рассматриваться.



В версиях, предшествующих Excel 97, рабочий лист содержит только 16 384 строки.

Для чего же нужны списки? Цели использования списков самые разнообразные. Одни пользователи используют списки просто для отслеживания информации (например, информации о заказах), другие — для хранения данных, которые, в конечном счете, превращаются в отчеты. Однако, как правило, работа с любыми списками состоит из следующих операций:

- ◆ Ввод данных
- ◆ Фильтрация списка с целью отображения отдельных строк, отвечающих определенным условиям
- ◆ Сортировка списка
- ◆ Вставка формул для вычисления промежуточных итогов
- ◆ Создание формул для вычисления результатов, полученных на основе списка, который отфильтрован по определенным условиям
- ◆ Создание итоговой таблицы данных (как правило, для этого используется сводная таблица)

Все эти операции позволяют пользователю эффективно управлять организацией информации в списке. Ниже приведено несколько рекомендаций, которые помогут вам грамотно создать собственный проект списка.

Проектирование списка

Несмотря на то, что Excel весьма удобен в управлении информацией, хранящейся в списке, планирование организации данных в списке имеет важное значение и позволяет сделать список максимально простым в использовании. Поэтому при создании списка воспользуйтесь следующими рекомендациями:

- ◆ **Первая строка списка (строка заголовка) должна содержать понятные описательные подписи (одна для каждого столбца).** Если в качестве подписей столбцов вы используете длинные текстовые строки, используйте формат ячеек Переносить по словам, чтобы избежать нецелесообразного расширения столбцов.
- ◆ **Каждый столбец должен содержать только один тип информации.** К примеру, не стоит смешивать в одном столбце текстовые значения и значения в формате даты.
- ◆ **Формулы, используемые для вычислений данных в других полях списка, должны располагаться в той же строке, что и исходные данные.** В том случае, если вычисляющие формулы обращаются к ячейкам, расположенным за пределами списка, результаты, полученные при сортировке списка, могут быть самыми неожиданными.
- ◆ **Список не должен содержать пустых строк.** Работая со списком, Excel автоматически определяет его границы, поэтому пустая строка рассматривается как конец списка.
- ◆ **Для достижения правильности вычислений, список на рабочем листе должен храниться отдельно от другой информации.** В том случае, если, кроме самого списка, на рабочем листе содержится другая информация, она должна размещаться выше или ниже списка. Другими словами, не используйте ячейки, расположенные справа или слева от списка.
- ◆ **Первая строка списка должна быть закреплена.** Для этого выберите команду меню Окно⇒ Закрепить области, после чего вы сможете свободно перемещаться по списку, не теряя из виду подписи полей.
- ◆ **Прежде чем приступить к созданию данных, отформатируйте столбец, в который эти данные будут вводиться.** Например, если столбец содержит значения даты, записи в этом столбце должны отображаться в формате даты.

Использование средства Автофильтр

Фильтрация списка, наиболее часто встречающийся и очень полезный способ работы со списками, представляет собой процесс скрытия всех строк за исключением тех, которые отвечают условиям, определенным пользователем. Например, если рабочий лист содержит список заказчиков, фильтрация списка может быть использована для того, чтобы отобразить только тех заказчиков, которые проживают в штате Орегон.



Приложение Excel предоставляет в распоряжение пользователя два средства фильтрации списков. Первый — Автофильтр, как правило, использует простые условия фильтрации. Другой — Расширенный фильтр, предназначен для создания списка данных, отвечающих комплексным условиям. Более подробно этот тип фильтрации рассмотрен далее в этой главе.

Принципы работы средства Автофильтр

Для того чтобы использовать функциональные возможности средства Автофильтр, поместите курсор мыши в одну из ячеек списка и выберите команду Данные⇒Фильтр⇒Автофильтр. Excel самостоятельно определяет диапазон ячеек, в котором находится список, и добавляет к ячейкам, содержащим подписи полей, стрелки раскрывающегося списка (рис. 9.2).

	A	B	C	D	E	F
1	Name	Annual Salary	Monthly Salary	Location	Date Hired	Exempt
2	Anthony Taylor	\$55,500	\$4,625	Los Angeles	5/2/1998	FALSE
3	Charles S. Billings	\$39,000	\$3,250	Los Angeles	11/1/2000	FALSE
4	Chris Poundsworth	\$29,850	\$2,488	Seattle	6/12/1997	FALSE
5	Clark Bickerson	\$120,000	\$10,000	Los Angeles	4/19/2000	TRUE
6	Douglas Williams	\$89,687	\$7,474	Portland	1/3/1998	TRUE
7	Ivan Silberstein	\$95,000	\$7,917	Los Angeles	12/1/1998	TRUE
8	James Millen	\$27,690	\$2,308	Los Angeles	3/16/1997	FALSE
9	Jeffrey P. Jones	\$42,000	\$3,500	Seattle	4/6/2000	FALSE
10	Joe Morrison	\$24,000	\$2,000	Seattle	3/13/1999	FALSE
11	John T. Foster	\$39,500	\$3,292	Seattle	10/10/1999	FALSE
12	Kurt Karmichoff	\$48,000	\$4,000	Los Angeles	11/5/2000	FALSE
13	Michael Hayden	\$78,230	\$6,519	Seattle	6/1/1999	FALSE
14	Phillip A. Todd	\$29,500	\$2,458	Portland	1/3/1998	TRUE
15	Richard E. Card	\$43,000	\$3,583	Seattle	7/2/1997	FALSE
16	Rick Fogerty	\$89,873	\$7,489	Portland	6/28/1998	TRUE
17	Robert H. Miller	\$149,000	\$12,417	Portland	5/1/1999	TRUE
18	Stephen C. Carter	\$44,123	\$3,677	Los Angeles	9/29/1998	FALSE
19	Steven H. Katz	\$32,900	\$2,742	Los Angeles	3/16/2000	FALSE
20	Thomas E. Abbott	\$60,000	\$5,000	Los Angeles	4/12/1997	FALSE
21	Tom Brown	\$65,000	\$5,417	Seattle	5/27/2000	TRUE
22						

Рис. 9.2. После выбора команды **Данные** ⇒ **Фильтр** ⇒ **Автофильтр** Excel добавляет стрелки раскрывающегося списка к подписям полей в строке заголовка

Щелкните на одной из стрелок раскрывающегося списка — на экране отобразится список всех уникальных элементов, содержащихся в данном столбце. Выберите необходимый элемент, все строки списка, за исключением тех, которые содержат этот элемент, будут скрыты. Фильтрация списка выполняется по одному или по нескольким полям списка. После фильтрации поля цвет стрелки раскрывающегося в этом поле списка изменяется, тем самым сообщая пользователю, что к данным этого столбца уже применен фильтр.



Имейте в виду, что область действия средства Автофильтра имеет ограничения. В раскрывающемся списке отображаются только первые 1 000 элементов. Если ваш список превышает это количество, вы можете использовать Расширенный фильтр (это средство описано ниже).

Помимо уникальных элементов, содержащихся в поле списка на рабочем листе, раскрывающийся список предлагает еще пять условий фильтрации:

- ◆ (Все). Отображает все содержащиеся в столбце элементы. Используйте это условие для удаления фильтра из столбца.
- ◆ (Первые 10). Отображает первые 10 строк с максимальными или минимальными значениями ячеек текущего столбца. В действительности этот термин употребляется неправильно, поскольку это условие позволяет отобразить “первые *n*” элементов (пользователю остается выбрать необходимое число).
- ◆ (Условие). Отображает элементы, удовлетворяющие одному или двум условиям (рис. 9.3).
- ◆ (Пустые). Отображает строки, содержащие в этом столбце пустые ячейки. Эта опция доступна только в том случае, если в столбце содержатся пустые ячейки.
- ◆ (Непустые). Отображает строки, содержащие в этом столбце непустые ячейки. Эта опция доступна только в том случае, если в столбце содержатся пустые ячейки.



После выполнения фильтрации Excel автоматически назначает диапазону ячеек, содержащему отфильтрованный список, скрытое имя (`_FilterDatabase`). Обратите внимание, что имя начинается с символа подчеркивания. В дальнейшем пользователь может использовать это имя в макросах VBA или формулах. Чтобы выбрать отфильтрованный диапазон данных, нажмите комбинацию клавиш `<Ctrl+G>`, — на экране отобразится диалоговое окно Перейти к. Имейте в виду, что скрытое имя не отображается в списке имен, поэтому его необходимо вводить вручную. Просто введите в поле Ссылка значение `_FilterDatabase` и щелкните на кнопке ОК.

Выбор элемента (Условие) при использовании автофильтра весьма полезен, но все же имеет некоторые ограничения. Например, если вы хотите выполнить фильтрацию списка, чтобы отобразить только три значения поля (например, Нью-Йорк, Нью-Джерси или Коннектикут), использование автофильтра в данном случае неприемлемо. Такие задачи фильтрации требуют применения расширенного фильтра, который будет обсуждаться позже в этой главе.

Чтобы отменить фильтрацию и отобразить список полностью, щелкните на стрелке раскрывающегося списка и выберите первый элемент списка (Все). Кроме того, вы можете выбрать команду меню Данные⇒Фильтр⇒Отобразить все. Чтобы выйти из режима автофильтра и удалить стрелки раскрывающегося списка из ячеек с именами полей, выберите команду Данные⇒Фильтр⇒Автофильтр повторно.

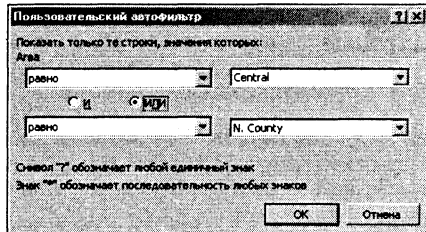


Рис. 9.3. Диалоговое окно Пользовательский автофильтр позволяет увеличить количество условий фильтрации

Подсчет и суммирование отфильтрованных данных

Для того чтобы отобразить определенное количество отфильтрованных записей, вы можете создать специальную формулу, используя функцию ПРОМЕЖУТОЧНЫЕ.ИТОГИ и значение 3 в качестве ее первого аргумента. Например:

```
=ПРОМЕЖУТОЧНЫЕ.ИТОГИ(3;A5:A400)
```

Первый аргумент функции ПРОМЕЖУТОЧНЫЕ.ИТОГИ определяет тип выполняемого "подведения итогов". Значение 3 в качестве первого аргумента определяет, что, в данном случае, подведение итогов аналогично использованию функции Excel СЧЕТЗ.

Имейте в виду, что аргумент, определяющий диапазон действия функции ПРОМЕЖУТОЧНЫЕ.ИТОГИ, должен начинаться с первой строки и заканчиваться последней строкой списка.



Данная формула должна вводиться в строку, расположенную выше или ниже самого списка. В противном случае, при фильтрации списка строка, содержащая формулу, будет скрыта. Кроме того, имейте в виду, что в итог, возвращаемый функцией ПРОМЕЖУТОЧНЫЕ.ИТОГИ, пустые ячейки не включаются.

Для того чтобы отобразить сумму значений, содержащихся в отфильтрованных записях, в качестве первого аргумента функции ПРОМЕЖУТОЧНЫЕ.ИТОГИ используйте значение 9. Например, следующая формула возвращает сумму значений, содержащихся в отфильтрованных строках столбца С:

```
=ПРОМЕЖУТОЧНЫЕ.ИТОГИ(9;C5:C400)
```

На рис. 9.4 отображены результаты, возвращенные формулами, использующими отфильтрованный список.



Функция **ПРОМЕЖУТОЧНЫЕ.ИТОГИ** — единственная функция, которая распознает данные, скрытые с помощью автофильтра. Другие формулы, использующие в качестве аргументов данные отфильтрованного списка, обращаются ко всем ячейкам списка. Например, если ячейка содержит формулу, которая суммирует значения, содержащиеся в столбце С, сумма значений будет включать не только значения в отображенных строках, а все значения столбца С.



Функция **ПРОМЕЖУТОЧНЫЕ.ИТОГИ** используется для нумерации строк отфильтрованного списка. Нумерация строк корректируется каждый раз, когда вы используете фильтр, скрывая или отображая необходимые строки списка. Например, если подписи полей списка содержатся в строке 1, введите следующую формулу в ячейку А2, а затем скопируйте ее в каждую строку столбца А:
=ПРОМЕЖУТОЧНЫЕ.ИТОГИ(3;В\$2:В2)

Для получения более подробной информации об использовании функции **ПРОМЕЖУТОЧНЫЕ.ИТОГИ** см. раздел “Создание промежуточных итогов” далее в этой главе.

Копирование и удаление отфильтрованных данных

Некоторые стандартные операции, которые выполняются при работе с обычными электронными таблицами, с отфильтрованным списком работают несколько по-другому.

Month	Region	Sales
Feb	North	5,584
Mar	North	3,531
Apr	North	5,955
May	North	3,286
Jun	North	4,705
Jul	North	3,436
Aug	North	3,292
Sep	North	3,779
Oct	North	3,193

Рис. 9.4. Формулы в ячейках С1 и С2 используют функцию **ПРОМЕЖУТОЧНЫЕ.ИТОГИ**

Например, обычно для того чтобы скрыть строки, вы используете команду меню **Формат**⇒**Строка**⇒**Скрыть**. В дальнейшем, при копировании диапазона ячеек, включающего скрытые строки, Excel копирует все данные, включая данные, содержащиеся в скрытых ячейках. Но при копировании данных списка, к которому был применен автофильтр, Excel будет копировать только видимые строки выбранного диапазона.

То же самое происходит при удалении строк отфильтрованного списка. Excel удаляет только видимые строки выбранного диапазона. Строки, скрытые с помощью автофильтра, в данном случае, игнорируются.

Функция **ПРОМЕЖУТОЧНЫЕ.ИТОГИ**

Функция **ПРОМЕЖУТОЧНЫЕ.ИТОГИ** достаточно универсальна. Ее уникальность состоит в том, что это единственная функция приложения Excel, которая игнорирует ячейки, находящиеся в скрытых строках. Тем не менее, и она имеет одно “но”: строки должны быть скрыты с помощью средства Автофильтр или Схема списка. Обычное скрытие строк вручную никоим образом не отразится на результатах, возвращаемых функцией.

Первый аргумент функции **ПРОМЕЖУТОЧНЫЕ.ИТОГИ** определяет, какая из функций фактически будет использована. Например, если в качестве первого аргумента используется значение 1 — действие функции **ПРОМЕЖУТОЧНЫЕ.ИТОГИ** будет аналогично функции **СРЗНАЧ**. Ниже приведена таблица функций, соответствующих первому аргументу функции **ПРОМЕЖУТОЧНЫЕ.ИТОГИ**:

Значение
1
2
3
4
5
6
7
8
9
10
11

Функция
СРЗНАЧ
СЧЕТ
СЧЕТЗ
МАКС
МИН
ПРОИЗВЕД
СТАНДОТКЛОН
СТАНДОТКЛОНП
СУММ
ДИСП
ДИСПР

Использование расширенного фильтра

Во многих случаях возможности автофильтра вполне удовлетворяют потребности пользователя. Однако, если вы хотите увеличить количество условий фильтрации, вам необходимо использовать возможности расширенного фильтра. Расширенный фильтр — гораздо более гибкий инструмент работы с данными списка, чем автофильтр. Однако, перед тем как использовать его, необходимо выполнить ряд подготовительных действий. Расширенный фильтр предоставляет пользователю следующие возможности:

- ◆ Использование комплексных условий фильтрации.
- ◆ Использование вычисляемых условий фильтрации.
- ◆ Копирование строк, отвечающих определенным условиям, в другой диапазон ячеек, указанный пользователем.

Заполнение промежутков

Импортируя данные, вы можете получить рабочий лист, подобный отображенному на рисунке, приведенном ниже. В этом примере каждая из ячеек столбца А одновременно связана с несколькими строками, содержащими данные. Сортировка такого списка может привести к беспорядочному отображению информации, и, в конечном счете, вы можете не понять, кем из продавцов было сделано такое количество продаж.

Хорошо, если список невелик. В этом случае отсутствующие данные в пустые ячейки можно ввести вручную. Однако, что делать, если размер базы данных огромен? Тогда используйте следующий, более эффективный способ заполнения пустующих ячеек:

1. Выберите диапазон ячеек (например, А3 : А14).
2. Нажмите комбинацию клавиш <Ctrl+G>, чтобы отобразить диалоговое окно Перейти к.
3. В диалоговом окне Перейти к щелкните на кнопке Выделить.

4. Выберите опцию Пустые ячейки.
5. В строке формул введите знак равенства (=), а затем адрес первой ячейки столбца, содержащей запись (например, =A3), и нажмите комбинацию клавиш <Ctrl+Enter>.
6. Снова выберите диапазон ячеек, а затем выполните команду Правка⇒ Копировать.
7. Выберите команду Правка⇒ Специальная вставка, опцию Значения, а затем щелкните на кнопке Ок.

Sales Rep	Month	Units Sold	Amount	New Clients
Bob	Jan	324	\$22,356	5
	Feb	331	\$22,839	6
	Mar	290	\$20,010	3
Karen	Jan	189	\$13,041	12
	Feb	234	\$16,146	11
	Mar	398	\$27,462	6
Elizabeth	Jan	541	\$37,329	16
	Feb	212	\$14,628	21
	Mar	681	\$46,989	7
Stan	Jan	771	\$53,199	14
	Feb	322	\$22,218	3
	Mar	821	\$56,649	11

Создание диапазона условий

Прежде чем использовать расширенный фильтр списка, необходимо создать *диапазон условий*, соответствующих определенным потребностям пользователя. Диапазон условий содержит информацию, которую приложение Excel использует для фильтрации списка. Данный диапазон должен отвечать следующим требованиям:

- ◆ Диапазон должен содержать, по крайней мере, две строки. Первая строка должна содержать все или хотя бы отдельные подписи полей списка.
- ◆ Другие строки диапазона условий состоят из условий, определенных пользователем.

Диапазон условий может помещаться в любом месте рабочего листа или даже на отдельном рабочем листе. Однако не следует помещать диапазон условий в строки, которые входят в список, поскольку при фильтрации списка Excel может скрыть некоторые из этих строк, после чего отдельные строки диапазона не будут отображаться. Поэтому диапазон условий, как правило, должен помещаться выше или ниже самого списка.

На рис. 9.5 показан диапазон условий, расположенный выше использующего его списка, в диапазоне ячеек A1 : B2. Обратите внимание, что данный диапазон условий не включает все подписи полей списка, поскольку для диапазона условий достаточно включения только тех подписей, которые используются для создания условий фильтрации.

В этом примере диапазон условий содержит только одну строку. Поля в каждой строке диапазона (за исключением строки заголовка) объединены опе-

Month	Region	Sales
Jan	North	
Jan	South	2,838
Jan	West	5,072
Feb	North	5,584
Feb	South	3,504
Feb	West	4,118
Mar	North	3,531
Mar	South	5,321
Mar	West	2,584
Apr	North	5,955
Apr	South	4,801
Apr	West	5,555
May	North	3,286
May	South	5,899
May	West	1,458
Jun	North	4,705

Рис. 9.5. Диапазон условий для фильтрации списка

ратором И. Поэтому после применения расширенного фильтра в списке отображаются только те строки, в которых ячейки в столбце Month содержат текст Jan, И ячейки в столбце Region содержат текст North. Возможно, условия, определенные в диапазоне условий, могут показаться несколько ненадежными. Более подробно эта тема будет рассмотрена в этой главе позже в разделе “Определение условий расширенного фильтра”.

Выделение уникальных записей списка

Многих пользователей приложения Excel волнует один и тот же вопрос: “Как избавиться от дублирования записей в списке?”

Самое простое решение этой задачи — это использование расширенного фильтра. Активируйте с помощью указателя мыши любую из ячеек списка и выберите команду меню Данные⇒Фильтр⇒Расширенный фильтр. В диалоговом окне Расширенный фильтр выберите опцию Скопировать результат в другое место и введите в поле Поместить результат в диапазон новый диапазон для расположения записей (новый диапазон должен располагаться на том же самом рабочем листе). После этого установите флажок опции Только уникальные записи и щелкните на кнопке ОК. Вы получите полную копию вашего списка, за исключением двойных записей. Имейте в виду, что это единственная операция, выполняемая расширенным фильтром, которая не требует создания диапазона условий.

Фильтрация списка

Чтобы выполнить фильтрацию списка, выберите любую из ячеек списка, а затем выберите команду меню Данные⇒Фильтр⇒Расширенный фильтр. После этого на экране отобразится диалоговое окно, показанное на рис. 9.6. По умолчанию в поле Исходный диапазон Excel предлагает весь диапазон списка, который вы можете изменить по собственному усмотрению. Однако, вы должны определить диапазон условий. Чтобы отфильтровать список на месте (то есть, просто скрыть строки, не удовлетворяющие заданным условиям), выберите опцию Фильтровать список на месте. При выборе другой опции — Скопировать результат в другое место, в поле Поместить результат в диапазон необходимо задать диапазон ячеек. Щелкните на кнопке ОК и Excel отфильтрует список в соответствии с заданными условиями.

В случае копирования отфильтрованных записей в другое место (при выборе опции Скопировать результат в другое место), пользователь может указать, какие из столбцов необходимо скопировать. Для этого, перед тем как отобразить диалоговое окно Расширенный фильтр, скопируйте подписи необходимых полей в первую строку области, в которую будут вставлены отфильтрованные строки. В диалоговом окне Расширенный фильтр в поле Поместить результат в диапазон введите ссылку на столбцы, содержащие скопированные подписи. После этого скопированные строки будут включать только те столбцы, для которых скопированы подписи.

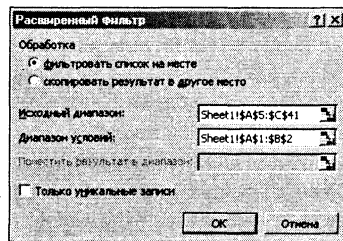


Рис. 9.6. Диалоговое окно Расширенный фильтр

Работа с данными списка

Выберите команду меню Данные⇒Форма — на экране отобразится диалоговое окно Excel, позволяющее пользователю значительно облегчить управление списками. Это диалоговое окно дает возможность вводить новые данные, удалять строки, искать строки, отвечающие определенным условиям.

Диалоговое окно управления формами очень удобно в использовании, но оно не идеально. Если же сама идея использования диалоговых окон для работы с данными списка вам по душе, попробуйте надстройку автора этой книги Enhanced Data Form, которая имеет множество преимуществ по сравнению с обычной формой Excel.

Data		Criteria
Name	James Millen	Next
AnnualSalary	27690	Insert
MonthlySalary	2307.5	Delete
Location	Los Angeles	Previous
Date Hired	3/16/1997	First
Exempt	FALSE	

Record 7 of 20

Edit

После того как надстройка установлена, иницируйте любую ячейку списка, а затем выберите команду меню Данные⇒JWalk Enhanced Data Form. Данные, содержащиеся в текущих записях, будут отображены в диалоговом окне. Для просмотра базы данных используйте горизонтальную полосу прокрутки (или кнопки Previous (Назад)/Next (Далее)). Все выполненные в этой форме изменения данных будут внесены в базу данных, также, как и их отмена. Данная форма позволяет обработать неограниченное количество полей, а поддерживающее групповые символы окно поиска дает возможность осуществить быстрый поиск необходимых записей, на основе данных любого поля.

Определение условий расширенного фильтра

Все старания компании Microsoft, связанные с возможностями управления списками в Excel, сфокусированы исключительно на автофильтрации. Принцип использования отдельного диапазона условий для расширенного фильтра возник вместе с появлением первой версии Lotus 1-2-3. Позднее этот метод был адаптирован к приложению Excel. Но он никогда не изменялся, даже несмотря на тот факт, что определение условий расширенного фильтра — один из самых щекотливых при работе с приложением Excel моментов. В этом разделе приведено множество примеров, которые помогут вам понять принципы создания диапазона условий, предназначенного для получения интересующей пользователя информации.

Все примеры в этом разделе используют список, показанный на рис. 9.7. Данный список содержит 125 записей и восемь полей и имеет достаточно разнообразный набор типов данных: значения, текстовые строки, логические значения и даты. Сам список размещен в диапазоне ячеек A8:H133 (строки, расположенные выше списка, используются для размещения диапазона условий).

	A	B	C	D	E	F	G	H
6								
7								
8	ListPrice	Date Listed	Area	Bedrooms	Baths	SqFt	Type	Pool
9	\$350 000	03.11.2000	N. County	3	2,5	1 991	Condo	ЛОЖЬ
10	\$215 000	05.11.2000	Central	3	1,75	2 157	Single Family	ИСТИНА
11	\$315 000	07.11.2000	S. County	2	2	1 552	Condo	ЛОЖЬ
12	\$379 000	11.11.2000	N. County	4	3	3 000	Single Family	ЛОЖЬ
13	\$248 500	21.11.2000	?	4	2,5	2 101	Single Family	ИСТИНА
14	\$297 500	23.11.2000	S. County	4	3,5	2 170	Single Family	ЛОЖЬ
15	\$259 900	27.11.2000	N. County	4	3	1 734	Condo	ЛОЖЬ
16	\$325 000	01.12.2000	S. County	4	3	2 800	Condo	ИСТИНА
17	\$208 750	03.12.2000	S. County	4	3	2 207	Single Family	ИСТИНА
18	\$227 500	03.12.2000		4	3	1 905	Condo	ЛОЖЬ
19	\$259 900	03.12.2000	N. County	3	2,5	2 122	Condo	ЛОЖЬ
20	\$405 000	06.12.2000	N. County	2	3	2 444	Single Family	ИСТИНА
21	\$236 900	07.12.2000	S. County	2	2	1 483	Condo	ЛОЖЬ
22	\$240 000	07.12.2000	S. County	3	2,5	1 595	Condo	ЛОЖЬ
23	\$304 900	09.12.2000	S. County	4	3	2 350	Single Family	ЛОЖЬ
24	\$349 900	13.12.2000	N. County	4	3	2 290	Single Family	ИСТИНА
25	\$249 000	21.12.2000	Central	4	3	1 940	Single Family	ИСТИНА
26	\$270 000	24.12.2000	Central	4	3	2 041	Single Family	ЛОЖЬ

Рис. 9.7. Данный список содержит информацию, представляющую собой перечень недвижимого имущества

Определение одного условия

Примеры, приведенные в этом разделе, характерны одним условием отбора записей. Другими словами, выбор необходимой записи определяет содержимое только одного поля.

Чтобы выбрать только те записи, которые в определенных полях содержатся определенные значения, введите имя поля в первую строку диапазона условий, а соответствующее значение во вторую строку. К примеру, диапазон условий (A1:A2), показанный на рис. 9.8, позволяет выбрать только те записи, которые в поле Bedrooms содержат значение 4.

	A	B	C	D	E	F	G	H
1	Bedrooms							
2	4							
3								
4								
5								
6								
7								
8	ListPrice	Date Listed	Area	Bedrooms	Baths	SqFt	Type	Pool
9	\$350 000	03.11.2000	N. County	3	2,5	1 991	Condo	ЛОЖЬ
10	\$215 000	05.11.2000	Central	3	1,75	2 157	Single Family	ИСТИНА
11	\$315 000	07.11.2000	S. County	2	2	1 552	Condo	ЛОЖЬ
12	\$379 000	11.11.2000	N. County	4	3	3 000	Single Family	ЛОЖЬ
13	\$248 500	21.11.2000	?	4	2,5	2 101	Single Family	ИСТИНА
14	\$297 500	23.11.2000	S. County	4	3,5	2 170	Single Family	ЛОЖЬ
15	\$259 900	27.11.2000	N. County	4	3	1 734	Condo	ЛОЖЬ
16	\$325 000	01.12.2000	S. County	4	3	2 800	Condo	ИСТИНА
17	\$208 750	03.12.2000	S. County	4	3	2 207	Single Family	ИСТИНА
18	\$227 500	03.12.2000		4	3	1 905	Condo	ЛОЖЬ
19	\$259 900	03.12.2000	N. County	3	2,5	2 122	Condo	ЛОЖЬ
20	\$405 000	06.12.2000	N. County	2	3	2 444	Single Family	ИСТИНА
21	\$236 900	07.12.2000	S. County	2	2	1 483	Condo	ЛОЖЬ

Рис. 9.8. Диапазон условий A1:A2 позволяет выбрать записи, которые содержат характеристики домов, имеющих четыре спальни



Чтобы выполнить подобную фильтрацию, можно использовать автофильтр.

Обратите внимание, что диапазон условий не обязательно должен включать подписи всех полей списка. Но если периодически вы используете разные наборы условий, гораздо удобней будет перечислить все подписи полей в первой строке диапазона условий.

Использование операторов сравнения

Чтобы сделать выбор записи более правильным, используйте операторы сравнения. Например, вы можете выбрать те записи, которые отвечают любому из следующих условий:

- ◆ Дома, которые имеют не менее четырех спален
- ◆ Дома, имеющие площадь менее 2 000 квадратных футов
- ◆ Дома по цене не более \$200 000 долларов

Чтобы выбрать записи, относящиеся к домам, имеющим четыре спальни, в диапазон условий введите следующие записи:

A1: Bedrooms
A2: >=4

Обратите внимание, в табл. 9.1 перечислены операторы сравнения, которые используются для сравнения текстовых или числовых значений. В том случае, если ни один из этих операторов сравнения не используется, по умолчанию в качестве оператора сравнения Excel использует знак равенства (=).

Таблица 9.1. Операторы сравнения

Оператор	Тип сравнения
=	Равно
>	Больше, чем
>=	Больше, чем или равно
<	Меньше, чем
<=	Меньше, чем или равно
<>	Не равно

В табл. 9.2 приведены примеры использования некоторых операторов сравнения.

Таблица 9.2. Примеры использования операторов сравнения

Условие	Выбор значений
>100	Записи, содержащие значения больше, чем 100
<>0	Записи, содержащие значения, не равные 0
=500	Записи, содержащие значения 500 (отсутствие знака равенства приводит к аналогичному результату)
<5000	Записи, содержащие значения меньшие, чем 5000
>=5000	Записи, содержащие значения меньшие или равные 5000

Использование групповых символов

Условия, представленные в виде текста, могут также использовать два групповых символа: звездочка (*) соответствует любому количеству символов; вопросительный знак (?) соот-

ветствует любому, но только одному, символу. В табл. 9.3 приведены примеры условий, содержащих текстовые значения. Конечно, некоторые из них редко встречаются в реальных условиях. Например, чтобы выбрать записи, содержащие один единственный символ, в качестве условия необходимо ввести формулу (см. последнюю запись в табл. 9.3).

Таблица 9.3. Примеры использования условий, содержащих текстовые значения

Условие	Выбор значения
= "Январь"	Записи, содержащие только текст <i>Январь</i> . Условие должно быть введено точно так же, как показано в этом примере: формула с предшествующим ей знаком равенства.
Январь	Записи, содержащие текст, начинающийся со слова <i>Январь</i> .
С	Записи, содержащие текст, начинающийся с символа <i>С</i> .
<>С	Записи, содержащие текст, начинающийся с любого символа, за исключением символа <i>С</i> .
>К	Записи, содержащие текст, начинающийся с символов от <i>Л</i> до <i>Я</i> .
Округ	Записи, содержащие текст, который включает слово <i>Округ</i> .
См*	Записи, содержащие текст, начинающийся с символов <i>См</i> .
с*с	Записи, содержащие текст, начинающийся с символа <i>С</i> и далее содержащий символ <i>С</i> .
с?с	Записи, содержащие текст, начинающийся с символа <i>С</i> , третий символ которого также является символом <i>С</i> . Обратите внимание, что это не означает выбор слов, содержащих только три символа.
= "с*с"	Записи, содержащие текст, начинающийся и заканчивающийся символом <i>С</i> . Условие должно быть введено точно так же, как показано в этом примере: формула с предшествующим ей знаком равенства.
<>*с	Записи, содержащие текст, который не заканчивается символом <i>с</i> .
<>?????	Все записи, содержащие более или менее пяти символов, но не в точности пять.
<>*с*	Записи, не содержащие символ <i>с</i> .
~?	Записи, содержащие единственный символ вопроса.
=	Пустые записи.
<>	Непустые записи.
= "с"	Записи, содержащие единственный символ <i>С</i> . Условие должно быть введено точно так же, как показано в этом примере: формула с предшествующим ей знаком равенства.



Имейте в виду, что при сравнении текстовых значений регистр символов не учитывается. Например, символы *се** могут соответствовать как слову *Селигман*, так и слову *СЕКР*.

Определение множества условий

Достаточно часто у пользователя возникает необходимость выбора записи, учитывающей сразу несколько условий, которые основаны на соответствии более, чем одному полю списка или на соответствии множества значений одному и тому же полю. Такие условия используют логические операторы **И** или **ИЛИ**. Ниже приведено несколько примеров использования множества условий при обращении к базе данных по недвижимости:

- ◆ Цена дома ниже, чем \$ 250 000, минимальная площадь 2 000 квадратных футов.

- ◆ Дом с бассейном только для одной семьи.
- ◆ Дом имеет не менее четырех спален, не менее трех ванн, площадь дома менее 3 000 квадратных футов.
- ◆ Дом, находящийся в списке не более одного месяца, с ценой выше \$300 000.
- ◆ Совладение, имеющее площадь между 1 000 и 1 500 квадратных футов.
- ◆ Дом для одной семьи, находящийся в списке в марте.

Чтобы объединить условия с помощью оператора И, в диапазоне условий должно использоваться множество столбцов. На рис. 9.9 показан диапазон условий, с помощью которого отбираются записи, содержащие характеристики домов по цене менее, чем \$250 000, и площадью не менее 2 000 квадратных футов.

	A	B	C	D	E	F	G	H
1	ListPrice	SqFt						
2	<250000	>=2000						
3								
4								
5								
6								
7								
8	ListPrice	Date Listed	Area	Bedrooms	Baths	SqFt	Type	Pool
9	\$350 000	03.11.2000	N. County	3	2,5	1 991	Condo	ЛОЖЬ
10	\$215 000	05.11.2000	Central	3	1,75	2 157	Single Family	ИСТИНА
11	\$315 000	07.11.2000	S. County	2	2	1 552	Condo	ЛОЖЬ
12	\$379 000	11.11.2000	N. County	4	3	3 000	Single Family	ЛОЖЬ
13	\$248 500	21.11.2000	?	4	2,5	2 101	Single Family	ИСТИНА
14	\$297 500	23.11.2000	S. County	4	3,5	2 170	Single Family	ЛОЖЬ
15	\$259 900	27.11.2000	N. County	4	3	1 734	Condo	ЛОЖЬ
16	\$325 000	01.12.2000	S. County	4	3	2 800	Condo	ИСТИНА
17	\$208 750	03.12.2000	S. County	4	3	2 207	Single Family	ИСТИНА
18	\$227 500	03.12.2000		4	3	1 905	Condo	ЛОЖЬ
19	\$259 900	03.12.2000	N. County	3	2,5	2 122	Condo	ЛОЖЬ
20	\$405 000	06.12.2000	N. County	2	3	2 444	Single Family	ИСТИНА
21	\$260 000	07.12.2000	S. County	2	3	1 400	Condo	ЛОЖЬ

Рис. 9.9. Данный диапазон условий для отбора записей использует множество столбцов и логический оператор И

На рис. 9.10 отображен другой пример диапазона условий, в соответствии с которым из списка выбираются все записи, внесенные в список в марте месяце. Обратите внимание, что одно из имен полей Date Listed (Дата занесения в список) появляется в диапазоне дважды, поскольку, в данном случае, используется следующее условие: Дата занесения в список должна быть больше, либо равна 1 марта, И дата занесения в список должна быть меньше, либо равна 31 марта.



Примите к сведению, что условия, показанные на рис. 9.9, с американскими форматами даты могут работать неправильно. Чтобы гарантировать совместимость заданных условий с различными системами дат, используйте функцию ДАТА, как, например, в следующих формулах:

= ">=" & ДАТА (2001 ; 3 ; 1)
 = "<=" & ДАТА (2001 ; 3 ; 31)

	A	B	C	D	E	F	G	H
1	Date Listed	Date Listed						
2	>=3.1.2001	<=3.31.2001						
3								
4								
5								
6								
7								
8	ListPrice	Date Listed	Area	Bedrooms	Baths	SqFt	Type	Pool
9	\$350 000	03.11.2000	N. County	3	2,5	1 991	Condo	ЛОЖЬ
10	\$215 000	05.11.2000	Central	3	1,75	2 157	Single Family	ИСТИНА
11	\$315 000	07.11.2000	S. County	2	2	1 552	Condo	ЛОЖЬ
12	\$379 000	11.11.2000	N. County	4	3	3 000	Single Family	ЛОЖЬ
13	\$248 500	21.11.2000	?	4	2,5	2 101	Single Family	ИСТИНА
14	\$297 500	23.11.2000	S. County	4	3,5	2 170	Single Family	ЛОЖЬ
15	\$259 900	27.11.2000	N. County	4	3	1 734	Condo	ЛОЖЬ
16	\$325 000	01.12.2000	S. County	4	3	2 800	Condo	ИСТИНА
17	\$208 750	03.12.2000	S. County	4	3	2 207	Single Family	ИСТИНА
18	\$227 500	03.12.2000		4	3	1 905	Condo	ЛОЖЬ
19	\$259 900	03.12.2000	N. County	3	2,5	2 122	Condo	ЛОЖЬ
20	\$405 000	06.12.2000	N. County	2	3	2 444	Single Family	ИСТИНА

Рис. 9.10. Данный диапазон условий позволяет выбрать из списка записи, содержащие характеристики домов, внесенных в список в марте месяце

Чтобы объединить условия с помощью оператора ИЛИ, диапазон условий должен содержать более, чем одну строку с условием, что каждая из них объединяется с другой посредством оператора ИЛИ. Количество строк в диапазоне условий не ограничено. Обратите внимание, на рис. 9.11 показан диапазон условий (A1 : C3), содержащий две строки.

	A	B	C	D	E	F	G	H
1	Type	SqFt	ListPrice					
2	Condo	>=1800						
3	Single Family		<210000					
4								
5								
6								
7								
8	ListPrice	Date Listed	Area	Bedrooms	Baths	SqFt	Type	Pool
9	\$350 000	03.11.2000	N. County	3	2,5	1 991	Condo	ЛОЖЬ
10	\$215 000	05.11.2000	Central	3	1,75	2 157	Single Family	ИСТИНА
11	\$315 000	07.11.2000	S. County	2	2	1 552	Condo	ЛОЖЬ
12	\$379 000	11.11.2000	N. County	4	3	3 000	Single Family	ЛОЖЬ
13	\$248 500	21.11.2000	?	4	2,5	2 101	Single Family	ИСТИНА
14	\$297 500	23.11.2000	S. County	4	3,5	2 170	Single Family	ЛОЖЬ
15	\$259 900	27.11.2000	N. County	4	3	1 734	Condo	ЛОЖЬ
16	\$325 000	01.12.2000	S. County	4	3	2 800	Condo	ИСТИНА
17	\$208 750	03.12.2000	S. County	4	3	2 207	Single Family	ИСТИНА
18	\$227 500	03.12.2000		4	3	1 905	Condo	ЛОЖЬ
19	\$259 900	03.12.2000	N. County	3	2,5	2 122	Condo	ЛОЖЬ
20	\$405 000	06.12.2000	N. County	2	3	2 444	Single Family	ИСТИНА

Рис. 9.11. Данный диапазон содержит два набора условий, каждый из которых расположен в отдельной строке диапазона

В этом примере после фильтрации списка на экран выводятся строки, отвечающие любому из следующих условий:

- ◆ Совладение площадью не менее 1 800 квадратных футов ИЛИ
- ◆ Дом для одной семьи по цене ниже \$ 210 000



Имейте в виду, что с помощью автофильтра подобный вид отбора выполнить невозможно.

Определение вычисляемых условий

Использование вычисляемых условий представляет собой довольно мощный инструмент при фильтрации списка. Благодаря вычисляемым условиям вы можете фильтровать список на основании одного или большего количества вычислений. На рис. 9.12 показан диапазон условий, на основании которого Excel отбирает записи, содержащие характеристики домов, цена которых ниже средней по всему прайс-листу. Ячейка B2 содержит следующую формулу:

=ListPrice>СРЗНАЧ (А:А)



Имейте в виду, что вы должны подключить опцию Допускать названия диапазонов. В противном случае, данная формула будет возвращать ошибку #ИМЯ?. Чтобы установить эту опцию, выберите команду меню Сервис⇒Параметры и перейдите на вкладку Вычисления. После подключения опции ошибка #ИМЯ? перестанет создавать проблемы.

	A	B	C	D	E	F	G	H
1		Above Avg						
2		ИСТИНА						
3								
4								
5								
6								
7								
8	ListPrice	Date Listed	Area	Bedrooms	Baths	SqFt	Type	Pool
9	\$350 000	03.11.2000	N. County	3	2,5	1 991	Condo	ЛОЖЬ
10	\$215 000	05.11.2000	Central	3	1,75	2 157	Single Family	ИСТИНА
11	\$315 000	07.11.2000	S. County	2	2	1 552	Condo	ЛОЖЬ
12	\$379 000	11.11.2000	N. County	4	3	3 000	Single Family	ЛОЖЬ
13	\$248 500	21.11.2000	?	4	2,5	2 101	Single Family	ИСТИНА
14	\$297 500	23.11.2000	S. County	4	3,5	2 170	Single Family	ЛОЖЬ
15	\$259 900	27.11.2000	N. County	4	3	1 734	Condo	ЛОЖЬ
16	\$325 000	01.12.2000	S. County	4	3	2 900	Condo	ИСТИНА
17	\$208 750	03.12.2000	S. County	4	3	2 207	Single Family	ИСТИНА
18	\$227 500	03.12.2000		4	3	1 905	Condo	ЛОЖЬ
19	\$259 900	03.12.2000	N. County	3	2,5	2 122	Condo	ЛОЖЬ
20	\$405 000	06.12.2000	N. County	2	3	2 444	Single Family	ИСТИНА
21		07.12.2000	S. County	3	2	1 483	Condo	ЛОЖЬ

Рис. 9.12. Данный диапазон условий использует вычисляемое условие отбора

Запомните следующие правила использования вычисляемых условий:

- ◆ Формула, вычисляющая условия, — это всегда логическая формула, возвращающая значения ИСТИНА или ЛОЖЬ.
- ◆ Используйте в формулах подписи полей. Например, в предыдущем примере имя ListPrice — это не имя диапазона, а подпись одного из полей базы данных. В качестве альтернативы можно использовать ссылку на ячейку первой строки с данными, расположенную в представляющем интерес поле (это не ссылка на ячейку, содержащую имя поля). В данном примере ячейка в первой строке данных поля ListPrice — это A9. Следующая формула возвращает тот же самый результат, что и в предыдущем примере:
=A9>СРЗНАЧ (А:А)
- ◆ Игнорируйте значения, возвращаемые формулами в диапазоне условий, которые ссылаются на первую строку списка. Иногда использование в формуле подписей полей приводит к ошибке, такой как, например, #ИМЯ? или #ЗНАЧ!. Эту ошибку можно игнорировать, поскольку она не влияет на фильтрацию списка.

- ◆ При использовании вычисляемых условий не используйте уже существующие в списке подписи полей. Обратите внимание, на рис. 9.12 ячейка B1 содержит текст Above Avg, который не является именем одного из полей списка. По существу, вычисляемые условия просто создают новое поле в списке, поэтому пользователь должен указать новое имя в первой строке диапазона условий или просто оставить пробел в ячейке, предназначенной для имени поля.
- ◆ Используйте в формуле, вычисляющей условия, ссылку на целый столбец списка. В предыдущем примере функция СРЗНАЧ использовала в качестве аргумента диапазон A:A. Однако имейте в виду, чтобы подобная формула, вычисляющая условия, работала, ее необходимо поместить в другой столбец, но никак не в тот, который упомянут в самой формуле. В противном случае это может привести к циклической зависимости. С другой стороны, вы можете просто использовать истинный адрес ячеек столбца в пределах списка.
- ◆ Количество условий, используемых для фильтрации списка, не ограничено. Кроме того, условия могут смешиваться, а также сочетаться с другими не вычисляемыми условиями.
- ◆ В том случае, если вычисляющая формула обращается к значению, находящемуся за пределами рабочего листа, который содержит список, вместо относительных ссылок используйте абсолютные. Например, \$C\$1 вместо C1.

Примеры вычисляемых условий

Обратите внимание, на рис. 9.13 показан пример вычисляемого условия, в соответствии с которым из списка отбираются записи, содержащие характеристики домов, для которых сумма спален и ванных комнат превышает значение 8. Имя столбца в ячейке A1 носит только описательный характер и никоим образом не отражается на фильтрации.

Формула, вычисляющая условия, возвращает значение ошибки, поскольку формула обращается к именам полей. Однако, несмотря на ошибку, при фильтрации списка формула работает правильно.

=Bedrooms+Baths>8

В качестве альтернативы можно использовать следующую формулу, которая обращается к первой строке списка, содержащей данные:

=D9+E9>8

Данная формула не возвращает значение ошибки, и понять ее не так просто.

	A	B	C	D	E	F	G	H
1	B* Rooms							
2	#ЗНАЧ!							
3								
4								
5								
6								
7								
8	ListPrice	Date Listed	Area	Bedrooms	Baths	SqFt	Type	Pool
9	\$350 000	03.11.2000	N. County	3	2,5	1 991	Condo	ЛОЖЬ
10	\$215 000	05.11.2000	Central	3	1,75	2 157	Single Family	ИСТИНА
11	\$315 000	07.11.2000	S. County	2	2	1 552	Condo	ЛОЖЬ
12	\$379 000	11.11.2000	N. County	4	3	3 000	Single Family	ЛОЖЬ
13	\$248 500	21.11.2000	?	4	2,5	2 101	Single Family	ИСТИНА
14	\$297 500	23.11.2000	S. County	4	3,5	2 170	Single Family	ЛОЖЬ
15	\$259 900	27.11.2000	N. County	4	3	1 734	Condo	ЛОЖЬ
16	\$325 000	01.12.2000	S. County	4	3	2 800	Condo	ИСТИНА
17	\$208 750	03.12.2000	S. County	4	3	2 207	Single Family	ИСТИНА
18	\$227 500	03.12.2000		4	3	1 905	Condo	ЛОЖЬ
19	\$259 900	03.12.2000	N. County	3	2,5	2 122	Condo	ЛОЖЬ
20	\$405 000	06.12.2000	N. County	2	3	2 444	Single Family	ИСТИНА
21	\$236 000	07.12.2000	S. County	3	2	1 483	Condo	ЛОЖЬ

Рис. 9.13. Данный диапазон условий использует вычисляемые выражения

Ниже приведен еще один пример формулы, вычисляющей условие. Данная формула выбирает только те записи, которые находились в списке в пределах последних 60 дней.

=B9>СЕГОДНЯ()-60

=Date Listed>СЕГОДНЯ()-60

Использование массивов с вычисляемыми условиями

Приложение Excel не просто позволяет использовать вычисляемые условия, наравне с этим Excel поддерживает использование массивов с вычисляемыми условиями. Чтобы показать на примере, насколько это полезно, давайте представим, что нам необходимо идентифицировать в списке дом, который имеет “половину ванной комнаты”. Для этого необходимо отфильтровать все записи, содержащие нецелые числа: 3,5, 4,5 или любое другое значение нецелого числа в поле Baths. Обратите внимание, на рис. 9.14 отображен один из таких примеров, где диапазон условий A1:A5, чтобы сделать отбор необходимых данных, использует четыре оператора ИЛИ.

	A	B	C	D	E	F	G	H
1	Baths							
2		2						
3		3						
4		4						
5		5						
6								
7								
8	ListPrice	Date Listed	Area	Bedrooms	Baths	SqFt	Type	Pool
9	\$350 000	03.11.2000	N. County	3	2,5	1 991	Condo	ЛОЖЬ
10	\$215 000	05.11.2000	Central	3	1,75	2 157	Single Family	ИСТИНА
11	\$315 000	07.11.2000	S. County	2	2	1 552	Condo	ЛОЖЬ
12	\$379 000	11.11.2000	N. County	4	3	3 000	Single Family	ЛОЖЬ
13	\$248 500	21.11.2000	?	4	2,5	2 101	Single Family	ИСТИНА
14	\$297 500	23.11.2000	S. County	4	3,5	2 170	Single Family	ЛОЖЬ
15	\$259 900	27.11.2000	N. County	4	3	1 734	Condo	ЛОЖЬ
16	\$325 000	01.12.2000	S. County	4	3	2 800	Condo	ИСТИНА
17	\$206 750	03.12.2000	S. County	4	3	2 207	Single Family	ИСТИНА
18	\$227 500	03.12.2000		4	3	1 905	Condo	ЛОЖЬ
19	\$259 900	03.12.2000	N. County	3	2,5	2 122	Condo	ЛОЖЬ
20	\$405 000	06.12.2000	N. County	2	3	2 444	Single Family	ИСТИНА
21	\$196 000	07.12.2000	S. County	2	2	1 483	Condo	ЛОЖЬ

Рис. 9.14. Чтобы выбрать записи, содержащие характеристики домов с нецелым количеством ванных комнат, здесь использовано четыре условия ИЛИ

Другая возможная операция вычисления условий задействует только одну формулу:

=ИЛИ (Baths {2 ; 3 ; 4 ; 5 ; 6 ; 7})

Данная формула возвращает значение ИСТИНА, если значение в поле Baths равно любому из значений массива.

Использование функций базы данных

Достаточно часто, для того чтобы создать формулы, результаты действия которых отвечают определенным условиям фильтрации, используются функции Excel, предназначенные для работы с базами данных электронной таблицы. Вы можете найти эти функции в диалоговом окне Мастер функций в категории Работа с базой данных.

Таблица 9.4. Функции Excel, предназначенные для работы с базами данных электронной таблицы

Функция	Описание функции
ДСРЗНАЧ	Возвращает среднее значение выбранных фрагментов базы данных.
БСЧЕТ	Подсчитывает количество числовых ячеек в выборке из заданной базы данных по заданному критерию.
БСЧЕТА	Подсчитывает количество непустых ячеек в выборке из заданной базы данных по заданному критерию.
БИЗВЛЕЧЬ	Извлекает из базы данных одну запись, удовлетворяющую заданному критерию.
ДМАКС	Возвращает максимальное значение среди выделенных фрагментов базы данных или записей базы данных.
ДМИН	Возвращает минимальное значение среди выделенных фрагментов базы данных или записей базы данных.
ВДПРОИЗВЕД	Перемножает значения определенных полей записей базы данных, удовлетворяющих критерию.
ДСТАНДОТКЛ	Оценивает стандартное отклонение по выборке (характеризующей генеральную совокупность) из выделенной части базы данных.
ДСТАНДОТКЛП	Вычисляет стандартное отклонение по генеральной совокупности из выделенной части базы данных.
БДСУММ	Суммирует числа в поле столбца записей базы данных, удовлетворяющих условию.
ВДДИСП	Оценивает дисперсию по выборке из выделенной части базы данных (характеризующей генеральную совокупность).
ВДДИСПП	Вычисляет дисперсию по генеральной совокупности из выделенной части базы данных.

Все функции базы данных требуют отдельного диапазона условий, который указывается в качестве последнего аргумента функции. Диапазон условий, используемый функциями базы данных, полностью аналогичен диапазону, который рассмотрен ранее в разделе “Определение условий расширенного фильтра”.

Давайте рассмотрим рис. 9.15. Приведенная ниже формула, которая содержится в ячейке С2, предназначена для вычисления суммы значений списка, удовлетворяющих определенным условиям, использует функцию БДСУММ. В частности, формула возвращает сумму записей столбца Sales, которые в столбце Month содержат значение Feb, а в столбце Region — значения North или South.

=БДСУММ(БазаДанных; 3; Условия)

В данном случае, БазаДанных — это название списка, 3 — номер поля или столбца, который будет суммироваться, и Условия — имя диапазона условий (A1 : B3).

Следующая, альтернативная версия этой формулы, вместо номера поля использует его имя. Эта формула достаточно проста для чтения и продолжает корректно работать даже в том случае, если перед столбцом 3 будет вставлено новое поле.

=БДСУММ(База данных; "Продажи"; Условия)



Безусловно, это может показаться весьма обременительным, создавать диапазон условий каждый раз при использовании функции базы данных. К счастью, Excel обеспечивает пользователя некоторыми альтернативными способами выполнения условного суммирования и подсчета. В главе 7 вы найдете множество таких примеров, которые используют функции СУММЕСЛИ, СЧЕТЕСЛИ и различные другие методы.

Если же вы страстный поклонник формул массивов, вместо диапазона условий можете использовать формулу массива. Теоретически, следующая формула массива *должна* быть работоспособна (и *могла бы* устранить необходимость использования отдельного диапазона условий). Но, к сожалению, функции базы данных не поддерживают массивы, поэтому данная формула возвращает ошибку #ЗНАЧ!.

```
{=БДСУММ(База данных;3;{"Month":"Regoin";"Feb":"North"})}
```



В первом выпуске версии приложения Excel 97 функции базы данных работают неправильно, если первый аргумент функции обращается к диапазону, который содержит более, чем 32 768 строк. Однако в версии Excel 97 SR-1 эта проблема полностью исправлена.

Работа с файлами Lotus 1-2-3?

Открывая в Excel файлы Lotus 1-2-3, имейте в виду, что, в отличие от Lotus, Excel оценивает диапазоны условий базы данных по-другому. Поэтому, это может сказаться на результатах, полученных при использовании расширенного фильтра и функций базы данных.

Например, в 1-2-3 условия типа "Джон" представляют только те строки, которые содержат ячейки с текстом "Джон". После того как файл 1-2-3 открывается в Excel, в работу вступает опция Производить вычисления по правилам Lotus 1-2-3. Если эта установка не изменена, диапазоны условий оцениваются, как если бы они находились в 1-2-3.

Для того чтобы Excel оценила диапазон условий, используя свои собственные правила (которые отличаются от Lotus 1-2-3), выберите команду меню Сервис⇒Параметры, а затем на вкладке Переход сбросьте флажок опции Производить вычисления по правилам Lotus 1-2-3. Например, условие "Джон" позволяет найти любые строки, которые содержат ячейки с текстом, начинающимся со слова "Джон"; включая ячейки, которые содержат текст "Джона", "Джон Смит" и "Джонсон".



Для получения более подробной информации о работе с файлами 1-2-3 обратитесь к приложению А.

Суммирование списка с помощью таблицы подстановки

В этом разделе мы рассмотрим одну из технологий суммирования информации базы данных, использующую команду Excel Данные⇒Таблица подстановки. Несмотря на тот факт, что достаточно часто при выполнении подобных действий мы привыкли использовать сводную таблицу, данная технология все же имеет одно большое преимущество. В отличие от сводной таблицы, которая требует обновления со стороны пользователя, таблица подстановки обновляется автоматически.

На рис 9.16 показан фрагмент самого обычного списка объемов продаж фирмы. Список содержит пять столбцов: общий ежемесячный объем продаж каждого торгового представите-

ля (столбец E), общее количество коммерческих контрактов представителя (столбец D), а также соответствующий регион работы данного представителя (либо *North*, либо *South* — столбец C). Например, в январе месяце торговый представитель фирмы в северном регионе по фамилии Bob заключил 58 контрактов на общую сумму продаж \$283 800.

1	Month	Region	
2	Feb	North	9,088 <- DSUM
3	Feb	South	
4			
5	Month	Region	Sales
6	Jan	North	2,838
7	Jan	South	5,488
8	Jan	West	5,072
9	Feb	North	5,584
10	Feb	South	3,504
11	Feb	West	4,118
12	Mar	North	3,531
13	Mar	South	5,321
14	Mar	West	2,584
15	Apr	North	5,955
16	Apr	South	4,801
17	Apr	West	5,555
18	May	North	3,288

Рис. 9.15. Функция БДСУМ используется для суммирования списка с помощью диапазона условий

	Month	Sales Rep	Region	Contacts	Sales
1	Jan	Bob	North	58	283,800
2	Jan	Frank	North	35	507,200
3	Jan	Paul	South	25	107,600
4	Jan	Randy	South	47	391,600
5	Jan	Mary	South	39	226,700
6	Feb	Bob	North	44	558,400
7	Feb	Jill	North	46	350,400
8	Feb	Frank	North	74	411,800
9	Feb	Paul	South	29	154,200
10	Feb	Randy	South	45	258,000
11	Feb	Mary	South	52	233,800
12	Mar	Bob	North	30	363,100
13	Mar	Jill	North	44	532,100
14	Mar	Frank	North	57	258,400
15	Mar	Paul	South	13	286,000
16	Mar	Randy	South	14	162,200
17	Mar	Mary	South	36	134,300
18	Apr	Bob	North	54	595,500
19	Apr	Jill	North	44	480,100
20	Apr	Frank	North	79	555,500
21	Apr	Paul	South	36	328,200
22	Apr	Randy	South	31	154,200
23	Apr	Mary	South	22	200,600
24	May	Bob	North	63	328,600
25	May	Jill	North	70	589,900

Рис. 9.16. Чтобы суммировать этот список, вы можете воспользоваться таблицей подстановки

Данный список содержит 76 записей, весь диапазон списка (A1:E77) носит название Database. Другой диапазон — G1:H2, под названием Criteria, представляет Набор условий для фильтрации списка. Теперь, используя эти данные и команду меню Данные⇒Таблица подстановки, в диапазоне ячеек G8:K23 мы можем создать итоговую таблицу, содержащую ключевую информацию по определенному месяцу, как это показано на рис. 9.17.

Для создания итоговой таблицы выполните следующие действия:

1. Введите названия месяцев в диапазон ячеек G10:G21.
2. В диапазон ячеек H8:K8 введите подписи полей, представляющих интерес для суммирования.
3. В ячейки строки 9 введите формулы, приведенные ниже в табл. 9.5.
4. Выберите диапазон ячеек G9:K21.
5. Выберите команду меню Данные⇒Таблица подстановки. На экране отобразится диалоговое окно Таблица подстановки (рис. 9.18).
6. В этом диалоговом окне в поле Подставлять значение по столбцам введите адрес ячейки G2. Другое поле — Подставлять значение по строкам в оставьте пустым.
7. Щелкните на кнопке ОК.

Month	Sales Rep	Region	Contacts	Sales
Jan	Bob	North	58	283,800
Jan	Frank	North	35	507,200
Jan	Paul	South	25	107,600
Jan	Randy	South	47	391,600
Jan	Mary	South	39	226,700
Feb	Bob	North	44	558,400
Feb	Jill	North	46	350,400
Feb	Frank	North	74	411,800
Feb	Paul	South	29	154,200
Feb	Randy	South	45	258,000
Feb	Mary	South	52	233,800
Mar	Bob	North	30	353,100
Mar	Jill	North	44	532,100
Mar	Frank	North	57	258,400
Mar	Paul	South	13	286,000
Mar	Randy	South	14	162,200
Mar	Mary	South	36	134,300
Apr	Bob	North	54	595,500
Apr	Jill	North	44	480,100
Apr	Frank	North	79	555,500
Apr	Paul	South	36	328,200
Apr	Randy	South	31	154,200
Apr	Mary	South	77	700,600

	Sales Reps	Contacts	Sales	Sales/Contact
Jan	2	93	791,000	8,505
Feb	3	164	1,320,600	8,052
Mar	3	131	1,143,600	8,730
Apr	3	177	1,631,100	9,215
May	3	173	1,064,300	6,152
Jun	3	132	1,001,200	7,585
Jul	3	186	872,300	5,255
Aug	3	127	1,082,100	8,520
Sep	3	148	1,239,300	8,374
Oct	3	147	962,100	6,545
Nov	4	162	1,004,522	6,201
Dec	4	205	1,219,183	5,947
TOTALS		1,825	13,331,305	7,423

Рис. 9.17. Чтобы создать подобную итоговую таблицу, воспользуйтесь командой меню Данные⇒Таблица подстановки

Таблица 9.5. Ввод формул

Ячейка	Формула
H9	=БСЧЁТА(Database;"Sales Rep";Criteria)
I9	=БДСУММ(Database;"Contacts";Criteria)
J9	=БДСУММ(Database;"Sales";Criteria)
K9	=J9/K9

В диапазон ячеек H10:K21 Excel вставляет только одну формулу массива:

{=ТАБЛИЦА(;G2)}

Чтобы выполнить необходимые вычисления, формула использует информацию, содержащуюся в ячейках, расположенных слева (G10:G21) и выше (H9:K9) данного диапазона. Принцип ее работы состоит в том, что формула иницирует пересчет формул, содержащихся в строке 9, подставляя при этом в них соответствующие значения месяца, введенные в ячейки столбца G. Другими словами, диапазон условий, состоящий только лишь из одной строки, обрабатывает диапазон, содержащий некоторое множество строк.

В том случае, если в ячейку H2 ввести название необходимого региона (например, *North* или *South*), таблица подстановки будет отображать информацию именно для этого региона. Если же ячейка H2 останется незаполненной, таблица подстановки отобразит информацию, касающуюся всех регионов.

Создание промежуточных итогов

Следующая команда приложения Excel — команда меню Данные⇒Итоги, представляет собой весьма удобный инструмент работы со списками, который позволяет пользователю вставлять формулы в список автоматически. Данные формулы используют функцию ПРОМЕЖУТОЧНЫЕ.ИТОГИ, спектр действия которой гораздо шире, чем простое суммирование.

ние данных. Для того чтобы использовать эту особенность, всякий раз, при изменении значений в определенных полях, вы должны сортировать список.

На рис. 9.19 приведен пример списка, для которого вполне можно использовать вычисление промежуточных сумм. Данный список сортируется, в первую очередь, по полю *Month*, а затем по полю *Region*.

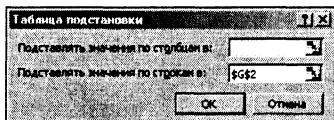


Рис. 9.18. Диалоговое окно Таблица подстановки используется для создания таблицы данных

	A	B	C	D	E
1	Month	State	Region	Contacts	Sales
2	Jan	California	West	58	263,800
3	Jan	Washington	West	35	507,200
4	Jan	Oregon	West	39	226,700
5	Jan	New York	East	25	107,600
6	Jan	New Jersey	East	47	391,600
7	Feb	California	West	44	558,400
8	Feb	Washington	West	74	411,800
9	Feb	Oregon	West	46	350,400
10	Feb	New York	East	52	233,800
11	Feb	New Jersey	East	29	154,200
12	Mar	California	West	30	353,100
13	Mar	Washington	West	57	258,400
14	Mar	Oregon	West	44	532,100
15	Mar	New York	East	36	134,300
16	Mar	New Jersey	East	14	162,200
17					

Рис. 9.19. Промежуточные итоги для данного списка вычисляются при каждом изменении месяца и региона

Чтобы вставить в список формулы автоматически подсчитывающие промежуточные итоги, поместите курсор мыши в любую ячейку списка и выберите команду меню Данные⇒Итоги. На экране отобразится диалоговое окно Промежуточные итоги, как это показано на рис. 9.20.

Диалоговое окно Промежуточные итоги предлагает пользователю выбрать следующие опции:

- ◆ При каждом изменении в. В этом раскрывающемся списке отображаются все поля, имеющиеся в списке базы данных. Весь список может быть отсортирован по выбранному полю.
- ◆ Операция. Раскрывающийся список предлагает выбрать одну из 11 доступных функций (по умолчанию Excel предлагает функцию суммирования).
- ◆ Добавить итоги по. В этом поле перечислены все имеющиеся в списке поля. Установите флажок рядом с полями, для которых должны вычисляться промежуточные итоги.
- ◆ Заменить текущие итоги. Если флажок для этой опции установлен, Excel будет удалять любые существующие формулы промежуточных итогов, заменяя их новыми формулами промежуточных итогов.
- ◆ Конец страницы между группами. Если флажок для этой опции установлен, после каждого промежуточного итога Excel вставляет разрыв страницы.
- ◆ Итоги под данными. Если флажок для этой опции установлен, Excel, по умолчанию, размещает промежуточные итоги под самими данными. В противном случае, формулы промежуточных итогов размещаются выше данных.
- ◆ Убрать все. Данная кнопка убирает из списка все формулы промежуточных итогов.

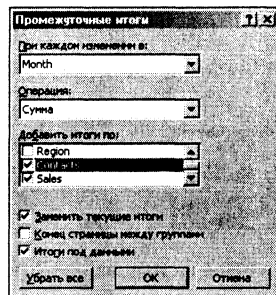


Рис. 9.20. Диалоговое окно Промежуточные итоги позволяет автоматически вставить формулы промежуточных итогов в сортируемый список

Щелкните на кнопке ОК, и Excel начнет анализировать список, вставляя определенные формулы и создавая уникальную схему промежуточных итогов списка. На рис. 9.21 показан рабочий лист после добавления к нему двух наборов промежуточных итогов, один из которых суммирует значения по месяцам, другой — по регионам. Безусловно, функция ПРОМЕЖУТОЧНЫЕ.ИТОГИ и использующие ее формулы вводятся вручную. Однако гораздо проще использовать команду меню Данные⇒Итоги.

1	Month	State	Region	Contacts	Sales
2	Jan	California	West	58	283,800
3	Jan	Washington	West	35	507,200
4	Jan	Oregon	West	39	226,700
5			West Total	132	1,017,700
6	Jan	New York	East	25	107,600
7	Jan	New Jersey	East	47	391,600
8			East Total	72	499,200
9	Jan Total			204	1,516,900
10	Feb	California	West	44	558,400
11	Feb	Washington	West	74	411,800
12	Feb	Oregon	West	46	350,400
13			West Total	164	1,320,600
14	Feb	New York	East	52	233,800
15	Feb	New Jersey	East	29	154,200
16			East Total	81	388,000
17	Feb Total			245	1,708,600
18	Mar	California	West	30	353,100
19	Mar	Washington	West	57	258,400
20	Mar	Oregon	West	44	532,100
21			West Total	131	1,143,600
22	Mar	New York	East	36	134,300
23	Mar	New Jersey	East	14	162,200
24			East Total	50	296,500
25	Mar Total			181	1,440,100
26					
27	Grand Total			630	4,665,600
28			Grand Total	630	4,665,600
29					

Рис. 9.21. Excel автоматически добавляет в список формулы промежуточных итогов и даже создает схему итогов списка



Имейте в виду, что при добавлении промежуточных итогов к отфильтрованному списку, после удаления фильтра итоги перестают быть точными.

Все формулы на этом рабочем листе используют функцию ПРОМЕЖУТОЧНЫЕ.ИТОГИ. Например, формула подсчета продаж в январе выглядит следующим образом:

=ПРОМЕЖУТОЧНЫЕ.ИТОГИ(9;E2:E7)

Несмотря на то, что данная формула обращается к двум ячейкам, которые также содержат формулу ПРОМЕЖУТОЧНЫЕ.ИТОГИ (ячейки E5 и E8), во избежание двойного счета в расчет итоговой суммы эти ячейки не включаются.

Чтобы установить уровень детализации отображаемых значений, воспользуйтесь элементами управления схемой списка. Например, на рис. 9.22 из всего списка показаны только строки с суммами, содержащие функцию ПРОМЕЖУТОЧНЫЕ.ИТОГИ.

Month	State	Region	Contacts	Sales
		West Total	132	1,017,700
		East Total	72	499,200
Jan Total			204	1,516,900
		West Total	164	1,320,600
		East Total	81	368,000
Feb Total			245	1,708,600
		West Total	131	1,143,600
		East Total	50	296,500
Mar Total			181	1,440,100
		Grand Total	630	4,665,600
		Grand Total	630	4,665,600

Рис. 9.22. Чтобы скрыть значения списка низших уровней и отобразить только строки с суммами, используйте элементы управления схемой списка

Резюме

В этой главе мы рассмотрели различные технологии создания и использования формул, предназначенных для работы со списком. Список (известный также как база данных электронной таблицы) представляет собой определенным образом организованную совокупность информации. Первая строка списка содержит имена полей, а последующие строки — данные (или записи). Автофильтр представляет собой весьма полезный способ фильтрации списка в соответствии с простыми условиями. А для фильтрации списка с помощью более сложных условий необходимо использовать расширенный фильтр, который требует определения диапазона условий. В этой главе мы также обсудили функции базы данных Excel (также требующие использования диапазона условий) и функцию ПРОМЕЖУТОЧНЫЕ.ИТОГИ.

Обратившись к главе 10, вы увидите насколько разнообразными могут быть вычисления в Excel.

Дополнительные вычисления

В этой главе...

- ◆ Преобразование единиц измерений
- ◆ Решение задач для прямоугольных треугольников
- ◆ Вычисление периметра, длины окружности, площади плоских фигур, поверхности и объема тел
- ◆ Решение систем линейных уравнений
- ◆ Округления
- ◆ Резюме

Эта глава содержит некоторую справочную информацию о других видах вычислений в приложении Excel. Возможно, вы не будете использовать их слишком часто, поэтому рассматривайте ее как шутку, и это поможет вам надолго запомнить полезный материал.

Преобразование единиц измерений

Представьте себе, что ваш офис находится в Нью-Йорке, и вы абсолютно точно знаете расстояние от Нью-Йорка до Лондона, но в милях. Приезжая в Европу, вы уже оперируете не милями, а километрами. Как перейти от расстояния в милях к расстоянию в километрах? Какой в данном случае коэффициент преобразования должен использоваться? В этом разделе вы найдете множество коэффициентов, которые станут полезны в различных формулах преобразования.



Надстройка Excel “Пакет анализа” предоставляет в распоряжение пользователя функцию ПРИБОР, которая позволяет выполнять множество различных преобразований единиц измерения (более детально возможности данной функции описаны в справочной системе). Однако, в некоторых случаях, гораздо эффективней создать собственные формулы преобразования. Для того чтобы выполнить это, необходимо использовать специальный коэффициент преобразования единиц измерения.

Таблицы преобразования единиц измерения

Чтобы преобразовать значение из одной единицы измерения в другую, воспользуйтесь одной из приведенных ниже таблиц коэффициентов преобразования. Для этого достаточно найти необходимый коэффициент преобразования. Например, для того чтобы преобразовать метры в дюймы, воспользуйтесь таблицей коэффициентов преобразования мер длины (Коэффициенты преобразования единиц измерения длины). В строке 3 (Метры) найдите столбец под названием Дюйм. Как видите, коэффициент преобразования метров в дюймы равен 39,37007874.

В дальнейшем этот коэффициент может использоваться в формулах. Например, если значение, содержащееся в ячейке A1, представлено в метрах, следующая формула преобразует это значение в дюймы:

$$=A1 * 39,37007874$$

Использование таблиц преобразования единиц измерения

Чтобы преобразовать метрические единицы измерения из одной размерности в другую, воспользуйтесь дополнительными коэффициентами преобразования, приведенными в табл. 10.1. Для этого просто умножьте исходное метрическое значение на необходимый коэффициент преобразования. Возьмем, к примеру, измерение расстояния в метрах. Километр — это 1 метр, умноженный на $1\text{E}+03$, или 1 000 метров. И наоборот, миллиметр — это 1 метр, умноженный на $1\text{E}-03$, или 1/1 000 метра.



Имейте в виду, что некоторые значения, которые отображены в таблицах, приведенных в этой главе, округлены. В свою очередь, таблицы коэффициентов преобразований, содержащиеся в рабочих книгах, содержат абсолютно точные значения. Чтобы увеличить точность каких-либо расчетов, используйте коэффициенты преобразования, приведенные в рабочих книгах.

Таблица 10.1. Коэффициенты преобразования метрических единиц

Метрические префиксы	Коэффициенты
Exa (экса)	$1\text{E}+18$
Peta (пета)	$1\text{E}+15$
Tera (тера)	$1\text{E}+12$
Giga (гига)	$1\text{E}+09$
Mega (мега)	$1\text{E}+06$
Kilo (кило)	$1\text{E}+03$
Hecto (гекто)	$1\text{E}+02$
Deci (деци)	$1\text{E}-01$
Centi (санти)	$1\text{E}-02$
Milli (милли)	$1\text{E}-03$
Micro (микро)	$1\text{E}-06$
Nano (нано)	$1\text{E}-09$
Pico (пико)	$1\text{E}-12$
Femto (фемто)	$1\text{E}-15$
Atto (атто)	$1\text{E}-18$

Для преобразования значения из метрической единицы измерения в неметрическую *умножьте* коэффициент преобразования единиц измерения на коэффициент преобразования метрической системы. Наоборот, для преобразования значения из неметрической единицы измерения в метрическую, используемый коэффициент преобразования необходимо *разделить* на коэффициент преобразования метрической системы.

К примеру, предположим, что ячейка A1 содержит значение в миллиметрах, которое необходимо преобразовать в дюймы. Умножьте значение в ячейке A1 на коэффициент перевода метр-дюйм (39,37007874), а затем полученный результат — на коэффициент преобразования метрической системы ($1\text{E}-03$). В результате формула будет выглядеть следующим образом:

$$=A1 * 39,37007874 * 1\text{E}-03$$

Безусловно, данная формула может иметь более простой вид, однако, результат останется неизменным:

$$=A1*0,03937007874$$

Теперь предположим, что в ячейке A1 содержится значение в дюймах, которое необходимо преобразовать в миллиметры. В этом случае коэффициент преобразования дюйм-метр будет равен 0,0254, а коэффициент преобразования метров — 1E-03. Формула преобразования дюймов в миллиметры будет выглядеть следующим образом:

$$=A1*0,0254/1E-03$$

Или в сокращенном виде:

$$=A1*25,4$$

Преобразования единиц измерения длины

Обратите внимание, в табл. 10.2 приведены коэффициенты преобразования для шести наиболее часто используемых единиц измерения длины. Подробно о ее использовании рассказано в разделе “Использование таблиц преобразования единиц измерения” ранее в этой главе.

Преобразования единиц измерения веса

В табл. 10.3 приведены коэффициенты преобразования для трех типовых единиц измерения веса. Подробно о ее использовании рассказано в разделе “Использование таблиц преобразования единиц измерения” ранее в этой главе.

Преобразования единиц измерения жидкости

В табл. 10.4 приведены коэффициенты преобразования для восьми наиболее часто используемых единиц измерения жидкости. Подробно о ее использовании рассказано в разделе “Использование таблиц преобразования единиц измерения” ранее в этой главе.

Преобразования единиц измерения площади

В табл. 10.5 приведены коэффициенты преобразования для семи наиболее часто используемых единиц измерения площади. Подробно о ее использовании рассказано в разделе “Использование таблиц преобразования единиц измерения” ранее в этой главе.

Преобразования единиц измерения объема

В табл. 10.6 приведены коэффициенты преобразования для четырех наиболее часто используемых единиц измерения объема. Подробно о ее использовании рассказано в разделе “Использование таблиц преобразования единиц измерения” ранее в этой главе.

Преобразования единиц измерения силы

В табл. 10.7 приведены коэффициенты преобразования для трех наиболее часто используемых единиц измерения силы. Подробно о ее использовании рассказано в разделе “Использование таблиц преобразования единиц измерения” ранее в этой главе.

Преобразования единиц измерения энергии

В табл. 10.8 приведены коэффициенты преобразования для пяти наиболее часто используемых единиц измерения энергии. Подробно о ее использовании рассказано в разделе “Использование таблиц преобразования единиц измерения” ранее в этой главе.

Преобразования единиц измерения времени

В табл. 10.9 приведены коэффициенты преобразования для пяти наиболее часто используемых единиц измерения времени. Подробно о ее использовании рассказано в разделе “Использование таблиц преобразования единиц измерения” ранее в этой главе.

Таблица 10.2. Коэффициенты преобразования единиц измерения длины

	Фут	Дюйм	Метр	Морская миля	Сухопутная уставная миля	Ярд
Фут	1	12	0,3048	0,000164579	0,000189394	0,333333333
Дюйм	0,083333333	1	0,0254	1,37149E-05	1,57828E-05	0,027777778
Метр	3,280839895	39,37007874	1	0,000539957	0,000621371	1,093613298
Морская миля	6076,115486	72913,38583	1852	1	1,150779448	2025,371828
Сухопутная уставная миля	5280	63360	1609,344	0,868976242	1	1759,999999
Ярд	3	36	0,9144	0,000493737	0,000568182	1

Таблица 10.3. Коэффициенты преобразования единиц измерения веса

	Грамм	Унция	Фунт
Грамм	1	0,035274	0,002205
Унция	28,34952	1	0,0625
Фунт	453,5923	16	1

Таблица 10.4. Коэффициенты преобразования единиц измерения жидкости

	Чаша	Унция жидкая	Галлон	Литр	Пинта	Кварта	Столовая ложка	Чайная ложка
Чаша	1	8	0,0625	0,23664	0,5	0,25	16	48
Унция жидкая	0,125	1	0,007813	0,02958	0,0625	0,03125	2	6
Галлон	16	128	1	3,786235	8	4	256	768
Литр	4,225833	33,80667	0,264115	1	2,112917	1,056458	67,61333	202,84
Пинта	2	16	0,125	0,473279	1	0,5	32	96
Кварта	4	32	0,25	0,946559	2	1	64	192
Столовая ложка	0,0625	0,5	0,003906	0,01479	0,03125	0,015625	1	3
Чайная ложка	0,020833	0,166667	0,001302	0,00493	0,010417	0,005208	0,333333	1

Таблица 10.5. Коэффициенты преобразования единиц измерения площади

	Акр	Гектар	Квадратный фут	Квадратный дюйм	Квадратный метр	Квадратная миля	Квадратный ярд
Акр	1	0,404685642	43560	6272640	4046,856422	0,0015625	4839,999997
Гектар	2,471053815	1	107639,1042	15500031	10000	0,003861022	11959,90046
Квадратный фут	2,29568E-05	9,2903E-06	1	144	0,09290304	3,58701E-08	0,1111111111
Квадратный дюйм	1,59423E-07	6,4516E-08	0,006944444	1	0,00064516	2,49098E-10	0,000771605
Квадратный метр	0,000247105	1E-04	10,76391042	1550,0031	1	3,86102E-07	1,195990046
Квадратная миля	640	258,998811	27878400	4014489600	2589988,11	1	3097599,998
Квадратный ярд	0,000206612	8,36127E-05	9	1296	0,836127361	3,22831E-07	1

Таблица 10.6. Коэффициенты преобразования единиц измерения объема

	Кубический фут	Кубический дюйм	Кубический метр	Кубический ярд
Кубический фут)	1	1728	0,028316847	0,037037037
Кубический дюйм	0,000578704	1	1,63871E-05	2,14335E-05
Кубический метр	35,31466672	61023,74409	1	1,307950618
Кубический ярд	27	46656	0,764554859	1

Таблица 10.7. Коэффициенты преобразования единиц измерения силы

	Дина	Ньютон	Фунт силы
Дина	1	0,00001	2,25E-06
Ньютон	10000	1	0,224809
Фунт силы	444822,2	4,448222	1

Таблица 10.8. Коэффициенты преобразования единиц измерения энергии

	BTU (Британская тепловая единица)	Calorie (IT) (Большая калория)	Calorie (Th' mic) (Калория)	Электрон- вольт	Эрг	Футофунт	Лошадиная сила в час	Джоуль	Ватт час
BTU (Британская тепловая единица)	1	251,9966	252,1655	6,59E+21	1,06E+10	25036,98	0,000393	1055,058	0,293072
Calorie (IT) (Большая калория)	0,003968	1	1,00067	2,61E+19	41867928	99,35441	1,56E-06	4,186795	0,001163
Calorie (Th' mic) (Калория)	0,003966	0,99933	1	2,61E+19	41839890	99,28787	1,56E-06	4,183991	0,001162
Электрон-вольт	1,52E-22	3,83E-20	3,83E-20	1	1,6E-12	3,8E-18	5,97E-26	1,6E-19	4,45E-23
Эрг	9,48E-11	2,39E-08	2,39E-08	6,24E+11	1	2,37E-06	3,73E-14	1E-07	2,7E-11
Футофунт	3,99E-05	0,010065	0,010072	2,63E+17	421399,8	1	1,57E-08	0,04214	1,17E-05
Лошадиная сила в час	2544,426	641186,8	641616,4	1,68E+25	2,68E+13	63704732	1	2684517	745,6997
Джоуль	0,000948	0,238846	0,239006	6,24E+18	9999995	23,73042	3,73E-07	1	0,000278
Ватт час	3,412133	859,8459	860,4221	2,25E+22	3,6E+10	85429,48	0,001341	3599,998	1

Таблица 10.9. Коэффициенты преобразования единиц измерения времени

	День	Час	Минута	Секунда	Год
День	1	24	1440	86400	0,002738
Час	0,041667	1	60	3600	0,000114
Минута	0,000694	0,016667	1	60	1,9E-06
Секунда	1,16E-05	0,000278	0,016667	1	3,17E-08
Год	365,25	8766	525960	31557600	1

Преобразование единиц измерения температуры

В этом разделе приведены формулы, предназначенные для преобразований между тремя шкалами температур: шкалой Фаренгейта, шкалой Цельсия и шкалой Кельвина. Преобразования температур не похожи на преобразования величин, обсуждавшихся ранее в этой главе, поскольку в данном случае необходимо использовать не только коэффициент преобразования, но и специальные формулы для преобразования величин. Обратите внимание, в табл. 10.10 приведены формулы, которые преобразуют значение температуры, содержащееся в ячейке под названием *темп*.

Таблица 10.10. Формулы преобразования единиц измерения температуры

Тип преобразования	Формула
Фаренгейт — Цельсий	$= (\text{темп} - 32) * (5 / 9)$
Фаренгейт — Кельвин	$= (\text{темп} - 32) * (5 / 9) + 273$
Цельсий — Фаренгейт	$= (\text{темп} * 1,8) + 32$
Цельсий — Кельвин	$= \text{темп} + 273$
Кельвин — Цельсий	$= \text{темп} - 273$
Кельвин — Фаренгейт	$= ((\text{темп} - 273) * 1,8) + 32$

Решение задач для прямоугольных треугольников

Любой прямоугольный треугольник имеет шесть компонентов: три стороны и три угла. Обратите внимание, на рис. 10.1 показан прямоугольный треугольник, который имеет три угла: А, В и С; а также три стороны: гипотенузу, основание и высоту. Угол С всегда равен 90 градусов (или $\pi/2$ радиан), поэтому, зная любые два другие компонента этого треугольника (исключая угол С), используя определенные формулы, вы всегда сможете найти оставшиеся компоненты этого треугольника.

Вспомните, как выглядит теорема Пифагора:

$$\text{Height}^2 + \text{Base}^2 = \text{Hypotenuse}^2$$

Если известны две стороны прямоугольного треугольника, всегда можно вычислить третью. Например, следующая формула вычисляет высоту прямоугольного треугольника по данным длин гипотенузы и основания:

$$= \text{КОРЕНЬ}(\text{Hypotenuse}^2 - \text{Base}^2)$$

Другая формула, вычисляющая основание прямоугольного треугольника, использует гипотенузу и высоту:

$$= \text{КОРЕНЬ}((\text{Hypotenuse}^2) - (\text{Height}^2))$$

Формула расчета гипотенузы прямоугольного треугольника использует основание и высоту:

$$= \text{КОРЕНЬ}((\text{Height}^2) + (\text{Base}^2))$$

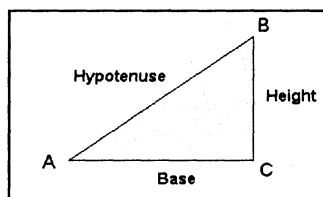
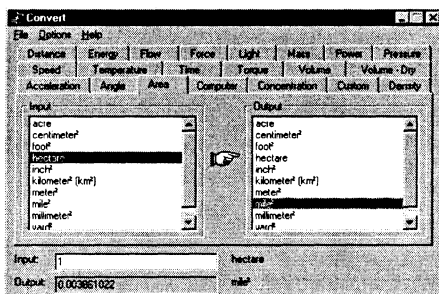


Рис. 10.1. Компоненты прямоугольного треугольника

Преобразование в другие единицы измерения

Безусловно, в одной только этой главе невозможно уместить все существующие в мире коэффициенты преобразования из одной единицы измерения в другую. Для многих систем измерения вы должны вычислить эти коэффициенты самостоятельно. Прекрасный источник для поиска такой информации — различные Web-узлы в Internet. Чем активнее вы используете поисковые сервера, тем выше вероятность того, что необходимая информация будет найдена.

Кроме того, существуют определенные виды программного обеспечения. Вы можете выгрузить с Web-узла <http://www/joshmadison.com/software/> весьма популярную (бесплатную) программу Джоша Мэдисона (Josh Madison) Convert, которая позволяет выполнять любые мыслимые и немыслимые преобразования единиц измерения.



Полезны также и другие тригонометрические тождества:

$$\text{SIN}(A) = \text{Height}/\text{Hypotenuse}$$

$$\text{SIN}(B) = \text{Base}/\text{Hypotenuse}$$

$$\text{COS}(A) = \text{Base}/\text{Hypotenuse}$$

$$\text{COS}(B) = \text{Height}/\text{Hypotenuse}$$

$$\text{TAN}(A) = \text{Height}/\text{Base}$$



Имейте в виду, что все тригонометрические функции Excel подразумевают, что угол, представляющий собой аргумент функции, представлен в радианах. Для преобразования градусов в радианы используйте функцию РАДИАНЫ. Для обратного преобразования радиан в градусы используйте функцию ГРАДУСЫ.

Если известны высота и основание, следующую формулу можно использовать для вычисления угла между гипотенузой и основанием (угол A):

$$=\text{ATAN}(\text{Height}/\text{Base})$$

Формула, приведенная выше, возвращает значение угла в радианах. Для преобразования значения в градусы используйте следующую формулу:

$$=\text{ГРАДУСЫ}(\text{ATAN}(\text{Height}/\text{Base}))$$

Если известны высота и основание, следующая формула может использоваться для вычисления угла между гипотенузой и высотой (угол B):

$$=\text{ПИ}()/2-\text{ATAN}(\text{Height}/\text{Base})$$

Данная формула возвращает значение в радианах. Для преобразования значения в градусы используйте следующую формулу:

$$=90-\text{ГРАДУСЫ}(\text{ATAN}(\text{Height}/\text{Base}))$$

На рис. 10.2 показана рабочая книга, которая содержит формулы для вычисления различных элементов прямоугольного треугольника.

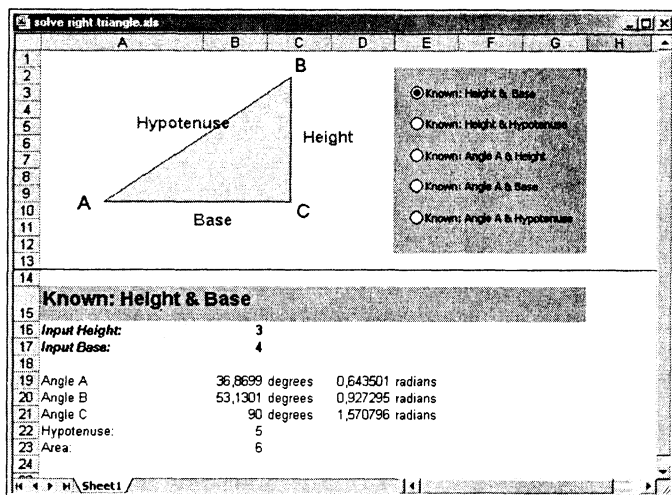


Рис. 10.2. Данная рабочая книга может быть очень полезна для вычислений компонентов прямоугольных треугольников

Вычисление периметра, длины окружности, площади плоских фигур, поверхности и объема тел

В этом разделе приведены формулы для расчета площади, поверхности, окружности и объема для простых двух- и трехмерных форм.

Вычисление площади и периметра квадрата

Площадь квадрата рассчитывается как квадрат длины одной из его сторон. Следующая формула вычисляет площадь квадрата, сторона которого находится в ячейке с именем сторона:

$$=\text{сторона}^2$$

Чтобы вычислить периметр квадрата, длину одной из его сторон умножьте на 4. Следующая формула вычисляет периметр квадрата, сторона которого находится в ячейке с именем сторона:

$$=\text{сторона} * 4$$

Вычисление площади и периметра прямоугольника

Площадь прямоугольника рассчитывается путем умножения его высоты на основание. Следующая формула возвращает площадь прямоугольника, используя ячейки с именами высота и основание:

$$= \text{высота} * \text{основание}$$

Периметр прямоугольника вычисляется путем сложения его удвоенной высоты и удвоенного основания. Следующая формула возвращает площадь прямоугольника, используя ячейки с именами высота и основание:

$$= (\text{высота} * 2) + (\text{основание} * 2)$$

Вычисление площади и периметра окружности

Площадь окружности рассчитывается как произведение квадрата радиуса на величину π . Следующая формула возвращает площадь окружности. Ячейка с именем радиус содержит радиус окружности.

$$= \text{ПИ}() * (\text{радиус}^2)$$

Радиус окружности равен половине ее диаметра.

Чтобы вычислить длину окружности, диаметр окружности необходимо умножить на величину π . Следующая формула вычисляет длину окружности, используя ячейку с именем диаметр.

$$= \text{диаметр} * \text{ПИ}()$$

Диаметр окружности равен радиусу окружности, умноженному на 2.

Вычисление площади трапеции

Для вычисления площади трапеции необходимо сложить две параллельные стороны, умножить их сумму на высоту и разделить на 2. Следующая формула вычисляет площадь трапеции, используя ячейки с именами сторона1, сторона2 и высота:

$$= ((\text{сторона1} + \text{сторона2}) * \text{высота}) / 2$$

Вычисление площади треугольника

Площадь треугольника представляет собой сумму основания и высоты треугольника, деленную на 2. Следующая формула вычисляет площадь треугольника, используя ячейки с именами основание и высота:

$$= (\text{основание} * \text{высота}) / 2$$

Вычисление площади поверхности и объема шара

Чтобы вычислить площадь поверхности шара, квадрат радиуса умножается на число π , после чего полученное произведение умножается на 4. Следующая формула возвращает площадь поверхности шара, радиус которого находится в ячейке с именем радиус:

$$= \text{ПИ}() * (\text{радиус}^2) * 4$$

Объем шара вычисляется как произведение радиуса, возведенного в третью степень, на 4π , а затем полученное произведение делится на 3. Следующая формула возвращает объем шара, радиус которого находится в ячейке с именем радиус:

$$= (\text{радиус}^3) * (4 * \text{ПИ}()) / 3$$

Вычисление площади поверхности и объема куба

Площадь поверхности куба вычисляется путем умножения квадрата его стороны на 6. Следующая формула вычисляет площадь поверхности куба, используя ячейку с именем сторона, которая содержит длину стороны куба:

$$= (\text{сторона}^2) * 6$$

Объем куба вычисляется возведением в третью степень длины его стороны. Следующая формула возвращает объем куба, используя ячейку с именем сторона:

$$= \text{сторона}^3$$

Вычисление площади поверхности и объема конуса

Следующая формула вычисляет площадь поверхности конуса (включая площадь основания). Формула использует ячейки по имени радиус и высота:

$$= \text{ПИ}() * \text{радиус} * (\text{КОРЕНЬ}(\text{высота}^2 + \text{радиус}^2) + \text{радиус})$$

Объем конуса вычисляется путем умножения квадрата радиуса основания на число π и высоту, затем произведение делится на 3. Следующая формула вычисляет объем конуса, используя ячейки с именами радиус и высота:

$$= (\text{ПИ}() * (\text{радиус}^2) * \text{высота}) / 3$$

Вычисление объема цилиндра

Чтобы вычислить объем цилиндра, квадрат радиуса основания цилиндра необходимо умножить на число π и высоту. Следующая формула рассчитывает объем цилиндра, используя ячейки с именами радиус и высота:

$$= (\text{ПИ}() * (\text{радиус}^2) * \text{высота})$$

Вычисление объема пирамиды

Чтобы вычислить объем пирамиды, необходимо вычислить площадь ее основания, а затем умножить ее на высоту и разделить на 3. Представленная ниже формула вычисляет объем пирамиды, используя для расчетов ячейки с именами ширина (ширина основания), длина (длина основания) и высота (высота пирамиды):

$$= (\text{ширина} * \text{длина} * \text{высота}) / 3$$

Решение систем линейных уравнений

В этом разделе мы рассмотрим формулы, которые используются для решения систем линейных уравнений. Ниже приведен пример системы линейных уравнений:

$$\begin{aligned}3x + 4y &= 8 \\4x + 8y &= 1\end{aligned}$$

Решение систем линейных уравнений предполагает нахождение значений x и y , удовлетворяющих обоим уравнениям. Система уравнений, приведенная выше, будет иметь следующее решение:

$$\begin{aligned}x &= 7,5 \\y &= -3,625\end{aligned}$$

Количество переменных в системе уравнений должно равняться числу уравнений. В предыдущем примере использованы два уравнения с двумя неизвестными. Для решения системы с тремя неизвестными (x , y и z) требуется три уравнения.

Обратите внимание, ниже приведены основные этапы решения системы уравнений, а на рис. 10.3 показано решение системы уравнений, представленной в начале этого раздела.

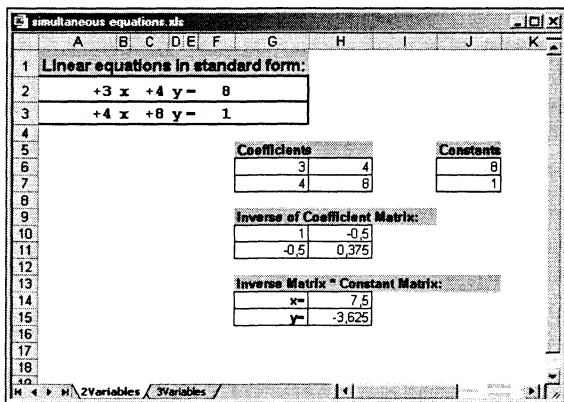


Рис. 10.3. Данная рабочая книга содержит формулы для расчета систем уравнений

1. Преобразуйте уравнения в стандартную форму: расположите все переменные слева от знака равенства. Для выполнения этой операции воспользуйтесь правилами элементарной алгебры. Например, два следующих уравнения идентичны, однако второе уравнение представлено в стандартной форме.

$$\begin{aligned}3x - 8 &= -4y \\3x + 4y &= 8\end{aligned}$$

2. Разместите коэффициенты уравнения в диапазоне ячеек $n \times n$, где n соответствует числу переменных. Обратите внимание, на рис. 10.3 коэффициенты расположены в диапазоне ячеек G6:H7.
3. Введите константы уравнения (значения, расположенные справа от знака равенства) в вертикальный диапазон ячеек. На рис. 10.3 это диапазон ячеек J6:J7.

4. Воспользуйтесь приведенной ниже формулой массива, чтобы сформировать обратную матрицу коэффициентов. На рис. 10.3 формула массива помещена в диапазон ячеек G10:H11.

{=МОБР(G6:H7)}

5. С помощью следующей формулы массива умножьте обратную матрицу коэффициентов на матрицу констант. На рис. 10.3 данная формула массива введена в диапазон ячеек H14:H15, который содержит решение системы уравнений.

{=МУМНОЖ(G10:H11, J6:J7)}



В главе 16 этой книги приведены примеры использования итерации для расчета ряда систем уравнений.

Округления

Excel предоставляет в распоряжение пользователя ряд функций, которые позволяют, используя различные методы, округлять введенные и полученные значения. Перечень этих функций приведен в табл. 10.11.

Таблица 10.11. Функции округления Excel

Функция	Описание
ОКРВВЕРХ	Округляет число до ближайшего целого или до ближайшего кратного указанному значению.
РУБЛЬ.ДЕС*	Преобразует цену в рублях, выраженную в виде дроби, в цену в рублях, выраженную десятичным числом.
РУБЛЬ.ДРОБЬ*	Преобразует цену в рублях, выраженную десятичным числом, в цену в рублях, выраженную в виде дроби.
ЧЕТН	Округляет число до ближайшего четного целого. Отрицательные числа округляются всегда в наименьшую сторону.
ОКРВНИЗ	Округляет число до ближайшего меньшего по модулю целого.
ЦЕЛОЕ	Округляет число до ближайшего меньшего целого.
НЕЧЕТ	Округляет число до ближайшего нечетного целого.
ОКРУГЛ	Округляет число до указанного количества десятичных разрядов.
ОКРУГЛВНИЗ	Округляет число до ближайшего меньшего по модулю целого.
ОКРУГЛВВЕРХ	Округляет число до ближайшего по модулю большего целого.
ОТВР	Отбрасывает дробную часть числа так, что остается целое число.

*Функции, доступные только после установки надстройки "Пакет анализа"



Примите к сведению, что существует разница между округлением значений и их форматированием. При форматировании значения для отображения конкретного положения десятичной точки, формулы, обращающиеся к этим значениям, используют истинное значение, а отличие состоит лишь во внешнем его представлении на экране. Когда же значение округляется, связанные с ним формулы используют неточные до определенного знака величины.



В главе 6 приведены примеры округления значений времени.

В следующем разделе мы рассмотрим примеры формул, использующих различные типы округлений.

Основные формулы округления

Функция ОКРУГЛ весьма полезна при округлении значений до указанного числа десятичных знаков. Число десятичных знаков указывается в качестве второго аргумента функции. Например, ниже приведена формула, которая возвращает значение 123,40 (значение округляется до одного десятичного знака):

=ОКРУГЛ(123,37;1)

В том случае, если второй аргумент функции ОКРУГЛ равен нулю, значение округляется до ближайшего целого. Например, следующая формула возвращает значение 123,00:

=ОКРУГЛ(123,37;0)

Второй аргумент функции ОКРУГЛ может быть и отрицательным. В таком случае число округляется слева от десятичной запятой. К примеру, следующая формула возвращает значение 120,00:

=ОКРУГЛ(123,37;-1)

Функция ОКРУГЛ округляет значения как вверх, так и вниз. Но как может измениться значение, такое как 12,5, при округлении его до целого числа? Оказывается, функция ОКРУГЛ округляет такие числа в сторону удаления от нуля. Например, следующая формула возвращает значение 13,0.

=ОКРУГЛ(12,5;0)

Другая формула, использующая в качестве первого аргумента отрицательное значение, возвращает значение -13,0 (округление в сторону удаления от нуля).

=ОКРУГЛ(-12,5;0)

В особых случаях вам может потребоваться несколько усложнить округление значений. Это можно выполнить с помощью функций ОКРУГЛВНИЗ и ОКРУГЛВВЕРХ. Например, следующая формула возвращает значение 12,0. Первоначальное значение округляется вниз.

=ОКРУГЛВНИЗ(12,5;0)

Следующая формула возвращает значение 13,0. Значение округлено вверх до ближайшего целого.

=ОКРУГЛВВЕРХ(12,43;0)

Округление значений денежных величин

Весьма часто у пользователя может возникнуть необходимость перевести денежную величину в рублях до ближайшего значения в копейках. Например, вычисленная цена может быть 45,78923р. Как в этом случае округлить вычисляемое значение цены до ближайшего значения в копейках? Это достаточно просто, ведь существует три метода округления таких значений:

- ◆ Округление вверх до ближайшего значения в копейках.
- ◆ Округление вниз до ближайшего значения в копейках.
- ◆ Округление до ближайшего значения в копейках (округление может быть как вверх, так и вниз).

Следующая формула использует значение в рублях и копейках, содержащееся в ячейке A1 и округляет его до ближайшего значения в копейках. К примеру, если ячейка A1 содержит значение 12,421р., то формула возвращает значение 12,42р.

=ОКРУГЛ(A1;2)

Чтобы округлить значение вверх до ближайшего значения в копейках, воспользуйтесь функцией ОКРВВЕРХ. Следующая формула округляет значение ячейки A1 вверх до ближайшего значения в копейках. К примеру, если ячейка A1 содержит значение 12,421р., то формула возвращает значение 12,43р.

=ОКРВВЕРХ(A1;0,01)

Другая функция — функция ОКРВНИЗ, предназначена для округления значения в рублях вниз. Например, следующая формула округляет значение ячейки A1 вниз до ближайшего значения в копейках. Если ячейка A1 содержит значение 12,421р., то формула возвращает значение 12,42р.

=ОКРВНИЗ(A1;0,01)

Чтобы округлить значение в рублях вверх до ближайших пяти копеек, воспользуйтесь следующей формулой:

=ОКРВВЕРХ(A1;0,05)

Работа с дробными значениями денежных единиц

Если по роду своей деятельности вы занимаетесь финансовыми расчетами, например, на фондовой бирже, то найдете для себя очень полезными функции приложения Excel РУБЛЬ.ДРОБЬ и РУБЛЬ.ДЕС, которые поставляются вместе с надстройкой “Пакет анализа”.

Возьмем, к примеру, значение 9,25р. Какую функцию необходимо использовать, чтобы точно выразить десятичную часть как простую дробь (9 1/4, 9 2/8, 9 4/16 и тому подобное). Функция РУБЛЬ.ДРОБЬ использует два аргумента: десятичное значение в рублях и целое число, которое будет использоваться в качестве знаменателя дроби. Например, следующая формула возвращает значение 9,1 (где, 1 — числитель дроби 1/4).

=РУБЛЬ.ДРОБЬ(9,25;4)



Имейте в виду, что значение, возвращаемое функцией РУБЛЬ.ДРОБЬ, нельзя использовать в других вычислениях. Например, значение, полученное в предыдущем примере, будет интерпретироваться как 9,1, но не как 9,25. Для выполнения расчетов с использованием этого значения, его необходимо преобразовать обратно в значение десятичной дроби. Для этого используется функция РУБЛЬ.ДЕС.

Другая функция — функция РУБЛЬ.ДЕС, преобразует значение в рублях, выраженное в виде дроби, в значение в рублях, выраженное десятичным числом. Данная функция также использует второй аргумент, который представляет собой знаменатель дроби. Например, следующая формула возвращает значение 9,25.

=РУБЛЬ.ДЕС(9,1;4)



Безусловно, функции РУБЛЬ.ДРОБЬ и РУБЛЬ.ДЕС предназначены не только для обработки значений в денежном формате. Вы можете использовать их, например, для работы с футами и дюймами. Предположим, что мы имеем значение, представленное в футах — 8,5. Чтобы преобразовать это значение в футы и дюймы, воспользуйтесь приведенной ниже формулой. Данная формула возвращает значение 8,06 (или 8 футов и 6 дюймов).

=РУБЛЬ.ДРОБЬ(8,5;12)

Другой пример использования этих функций — это ведение статистики в бейсболе. Подающий может выполнить 6 2/3 серии подач, что обычно отображается, как 6,2. Следующая формула возвращает значение 6,2:

=РУБЛЬ.ДРОБЬ(6+2/3;3)

Использование функций ЦЕЛОЕ и ОТБР

Внешне функции Excel ЦЕЛОЕ и ОТБР кажутся очень похожими, поскольку обе преобразуют исходное значение в целое. Функция ОТБР просто удаляет дробную часть числа. А функция ЦЕЛОЕ округляет число вниз до ближайшего целого, основываясь на дробной части числа.

На практике функции ЦЕЛОЕ и ОТБР возвращают различные результаты только тогда, когда в качестве аргумента функции используется отрицательное число. Например, следующая формула возвращает значение -14,0:

=ОТБР(-14,3;2)

Напротив, значение, которое возвращает следующая формула, будет равно -15,0, поскольку значение -14,3 округляется вниз до следующего меньшего целого.

=ЦЕЛОЕ(-14,3;2)

Функция ОТБР использует дополнительный (необязательный) параметр, который показывает количество десятичных знаков, которые должны присутствовать в возвращаемом значении. К примеру, формула, которая приведена ниже, возвращает значение 54,33 (после запятой остается два знака).

=ОТБР(54,3333333;2)

Округление до четного и нечетного целого

Иногда у пользователя может возникнуть необходимость округлить какое-либо значение вверх до ближайшего четного или нечетного целого. Для это и предназначены функции НЕЧЕТ и ЧЕТН. Данные функции используют только один аргумент и возвращают значение в виде целого числа. Функция ЧЕТН округляет аргумент вверх до ближайшего четного целого. Функция НЕЧЕТ округляет аргумент вверх до ближайшего нечетного целого. Обратите внимание, в табл. 10.12 показаны несколько примеров использования этих функций.

Округление значения до n-го разряда значащих цифр

Представьте себе, что вам необходимо округлить значение до определенного разряда значащих цифр. Например, округлить значение 1 432 187 до первых двух значащих цифр — 1 400 000. Или другой пример: значение 9 187 877 с тремя значащими цифрами будет равно 9 180 000.

Если значение представляет собой положительное число без десятичной части, вы можете воспользоваться следующей формулой, которая, в данном случае, будет работать. Формула округляет значение, содержащееся в ячейке A1 до двух значащих цифр. Для округления до другого числа значащих цифр необходимо заменить аргумент 2 этой формулы на необходимое значение.

=ОКРУГЛВНИЗ (A1 ; 2 -ДЛСТР (A1))

Таблица 10.12. Результат работы функций ЧЕТН и НЕЧЕТ

Число	Функция ЧЕТН	Функция НЕЧЕТ
-3,6	-4	-5
-3,0	-4	-3
-2,4	-4	-3
-1,8	-2	-3
-1,2	-2	-3
-0,6	-2	-1
0,0	0	1
0,6	2	1
1,2	2	3
1,8	2	3
2,4	4	3
3,0	4	3
3,6	4	5

Чтобы выполнить эту операцию для нецелых и отрицательных значений, необходимо прибегнуть к маленькой хитрости. Следующая формула представляет собой более распространенный метод решения задачи округления значения, содержащегося в ячейке A1. При этом число значащих знаков определяется в ячейке A2. Данная формула работает как для положительных, так и для отрицательных и нецелых чисел.

=ОКРУГЛ (A1 ; A2 -1 -ЦЕЛОЕ (LOG10 (ABS (A1))))

Например, если ячейка A1 содержит значение 1,27845, а ячейка A2 содержит значение 3, то формула возвращает значение 1,28000 (значение округлено до трех значащих цифр).

Резюме

В этой главе мы рассмотрели сразу несколько тем. Это и преобразование единиц измерения, и создание тригонометрических формул, и решение систем уравнений с двумя или тремя неизвестными, а также округление значений.

В следующей главе мы обсудим финансовые формулы.

Часть III

Финансовые формулы

ГЛАВА 11

“Знакомство
с финансовыми формулами”

ГЛАВА 12

“Финансовые функции
дисконтирования и обесценивания”

ГЛАВА 13

“Дополнительное использование
финансовых функций и формул”

Глава 11

Знакомство с финансовыми формулами

В этой главе...

- ◆ Основные финансовые функции Excel
- ◆ Анализ денежных потоков
- ◆ Функции накопления, дисконтирования и амортизации
- ◆ Преобразование процентных ставок
- ◆ Эффективная стоимость займов
- ◆ Вычисление процента и основных платежей
- ◆ Соответствие процентов и частоты выплаты
- ◆ Ограничения финансовых функций Excel
- ◆ Резюме

Безопасно спорить о том, что наиболее часто Excel используют для выполнения вычислений, в том числе и финансовых. Каждый день люди принимают сотни тысяч финансовых решений, основанных на числах, которые вычисляются в электронной таблице. Эти решения могут начинаться с простых (*Могу ли я купить новый автомобиль?*) и заканчиваться сложными (*Закончиться ли покупка Корпорации XYZ прибылью в течение последующих 18 месяцев?*) вопросами. Это первая из трех глав, в которых обсуждаются финансовые вычисления, которые можно выполнить с помощью Excel.

Основные финансовые функции Excel

В данной главе представлено много примеров использования пяти основных финансовых функций. Ниже приводится синтаксис этих функций (полуужирным шрифтом выделены обязательные аргументы):

- ◆ ПС (**ставка**; **кпер**; **плт**; бс; тип)
- ◆ ВС (**ставка**; **кпер**; **плт**; пс; тип)
- ◆ ПЛТ (**ставка**; **кпер**; **пс**; бс; тип)
- ◆ СТАВКА (кпер; ставка; пс; бс; тип; предположение)
- ◆ КПЕР (ставка; **плт**; **пс**; бс; тип)

Как вы видите, эти функции гибкие и крайне полезные при решении широкого ряда задач. Для эффективного использования этих функций необходимо различать три основных понятия:

- ◆ обозначение денежных потоков положительными или отрицательными числами;
- ◆ временное изменение количества денег;
- ◆ эквивалентность процентных ставок.

Эти понятия полностью раскрываются в этой главе и используются в последующих главах.

Основная терминология

- ◆ Текущее значение (ПС) — это приведенная стоимость. Если вы помещаете в банк \$5 000 на депозитный вклад, то эта сумма представляет начальную сумму или текущее количество денег, которые вы вложили. Если вы занимаете \$15 000 для приобретения автомобиля, эта сумма представляет приведенное или текущее значение займа. Текущее значение может быть положительным или отрицательным.
- ◆ Будущее значение (БС) — это начальная сумма плюс проценты. Если вы вкладываете \$5 000 на протяжении пяти лет при 6% годовых, то получите \$6 312,38 в конце пятилетнего срока. Это будущее значение вашего вклада с начальным значением \$5 000. Если вы берете трехгодовой заем \$15 000 для покупки авто и платите 7% годовых, то выплачиваете общую сумму \$16 673,16. Эта сумма представляет начальную сумму плюс выплаченный процент. Будущее значение может быть положительным и отрицательным.
- ◆ Платеж (ПЛТ) — это сумма одноразовой выплаты или одноразовой выплаты плюс проценты. Если вы вкладываете \$100 ежемесячно на сберегательный счет, то платеж составляет \$100. Если вы ежемесячно выплачиваете \$825, то платеж составлено из выплаты основной суммы и процентов.
- ◆ Процентная ставка — это процентное выражение начальной суммы, обычно имеющее ежегодную основу. Например, вы можете ежегодно получать 5,5 % с банковского депозитного вклада. Или вы можете взять заем со ставкой 7,75 %.
- ◆ Период — представляет время, на протяжении которого проводятся выплаты или получение процентов со счета. Например, банковский депозитный вклад с ежеквартальной выплатой процентов или заем для покупки авто, который требует ежемесячных платежей.
- ◆ Срок — это общее количество времени действия вклада или погашения займа. 12-месячный банковский депозитный счет рассчитан на один год. 30-летний заем рассчитан на 30-летний срок.

Анализ денежных потоков

Посмотрите на выписку из счета в банке, и денежные потоки станут очевидными! Имея дело с финансовыми функциями Excel, важно понимать то, как “представлять” денежные потоки, идущие в обоих направлениях. Другими словами: вы используете знаки “плюс” и “минус”?

Чтобы решить финансовые задачи, используя основные финансовые функции Excel, нужно выполнить два предварительных действия:

1. Определить владельца денежных средств. Например, в простой задаче накопления, вы смотрите на нее с точки зрения вкладчика или банка? В задаче займа, вы должник или кредитор? При вычислении стоимости ряда будущих выплат, являетесь вы покупателем (выплата за приобретенный товар) или вы продавец (получение выплат за проданный товар)?
2. Определить, поступают к вам денежные средства (знак “плюс”) или уходят от вас (знак “минус”).

Определившись с описанными выше категориями, вы сможете использовать финансовые функции Excel для проведения эффективных финансовых расчетов и правильно интерпретировать возвращаемые результаты.

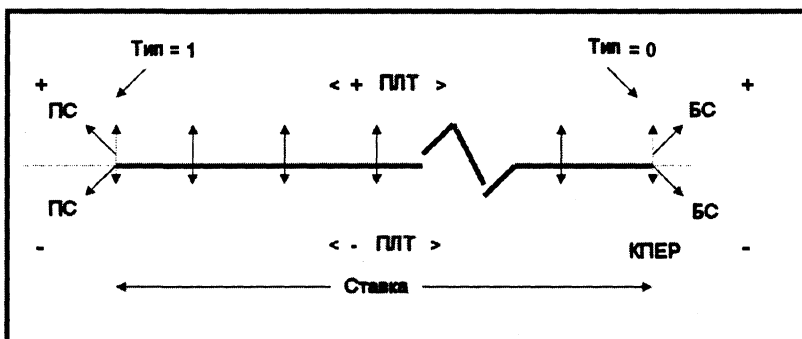
Связь между КПЕР, ПЛТ, и СТАВКА

Excel не “знает” ничего о различных периодах времени, например таких, как месяцы, недели или годы. Она рассчитывает периоды времени в определенных единицах измерения и предполагает, что параметры приведены соответствующим образом и не перепутаны.

Приведенная ниже диаграмма иллюстрирует временное перемещение денег, характеризуемых в Excel функциями ПС, БС, ПЛТ, КПЕР, и СТАВКА. Стрелки представляют потоки и их направление (оба варианта). Любая разрешимая задача состоит из четырех известных переменных и одной неизвестной переменной. Неизвестная переменная — название функции, а известные переменные представляют собой аргументы функции.

Диаграмма должна быть сбалансирована с точки зрения поступающих и исходящих денежных потоков. В данном случае используется единая процентная ставка, которая эффективна на протяжении периода времени, измеряемого КПЕР. Также предполагается, что используется единый уровень платежа, который справедлив на протяжении времени, измеряемого КПЕР. Аргумент Тип указывает на то, выплачиваются ли платежи авансом или в конце срока.

Если известны четыре из пяти переменных, то Excel может решить задачу. Существует всего одно исключение: в случае выплат, Excel нужно знать их срок (т.е. аргумент Тип).



Обычно поступающие к вам деньги отмечаются знаком “плюс”. Уходящие от вас деньги имеют знак “минус”. Например, если задача приведенной стоимости решается в пользу отрицательного значения, то это означает, что сумма предоставляется вами. Если значение положительное, то деньги вами одалживаются. Давайте рассмотрим пример вычисления платежей вклада. Если вы берете деньги в займы, то подсчитанные выплаты имеют знак “минус” (который указывает на то, что вы каждый период выплачиваете определенную сумму). Вычисляя процентную ставку займа, необходимо правильно определить направление выплат. Иначе Excel предположит, что все платежи происходят в одном направлении и породит ошибку. Например, формула может показать #ЧИСЛО!, которая указывает на бесконечно высокую норму прибыли (деньги приходят к вам, и ничего при этом вами не выплачивается).

Функции накопления, дисконтирования и амортизации

В этом разделе содержится ряд примеров, которые демонстрируют использование 5 основных функций Excel для решения задач накопления и амортизации денежных средств. Хотя мы рассматриваем амортизацию и накопление как отдельные задачи, это по сути одно и то же. Фактически, единственная разница в противоположности перемещения денежных средств.

Мы можем классифицировать рассматриваемые задачи по уровню сложности. В простых задачах мы имеем дело только с двумя из трех переменных (текущее значение, выплата и будущее значение). В сложных задачах мы имеем дело со всеми тремя параметрами. Хотя мы классифицируем их как простые и сложные задачи, формулы Excel требуют введения значений для всех трех переменных. Таким образом, вместо “отсутствующего” элемента в формуле вводится ноль.

Простые задачи накопления

В этом разделе содержится семь примеров, которые демонстрируют простые операции накопления.

ПРИМЕР 1

Сколько накопится на счету в \$1 000 по истечении трех лет при 7% годовых?

На рис. 11.1 показана эта задача на рабочем листе.

	A	B	C	D	E	F	G	H
1	Пример 1							
2	Задача							
3								
4	Какая сумма накопится на вкладе \$1000 спустя три года при годовой ставке 7%?							
5								
6	Функция							
7								
8	FV(ставка;кпер;плт;пс;тип)							
9	Выплат нет. ПЛТ равно 0, а тип -- 0							
10								
11								
12	Данные задачи							
13								
14	ставка			7%				
15	кпер			3				
16	плт			0				
17	пс			(\$1 000,00)				
18	тип			0				
19								
20	Ответ							
21								
22	FV			\$1 225,04				
23								

Рис. 11.1. Вычисление будущего значения

Используемая функция: FV(ставка;кпер;плт;пс;тип) Эта формула возвращает \$1 225,04:

=FV(7%;3;0;-1000;0)



Примеры формул в этой главе принимают строго заданные значения аргументов функции. В примерах на Web-сервере вместо аргументов функции используются ссылки на ячейки.

Обратите внимание, что эта задача рассматривается с точки зрения вкладчика. Таким образом, начальный вклад (пс) имеет отрицательное значение. Регулярные выплаты не производятся, так что аргумент плт равен 0. Без выплат тип аргумента несущественен.



Вводя числовые данные в качестве аргументов функции, убедитесь, что разделители разрядов не добавляются в ячейку. Например, вводите 1000, а не 1 000. В зависимости от региональных стандартов, разделители могут задаваться теми символами, что и разделители аргументов.

ПРИМЕР 2

Если вклад \$1 000 увеличился до \$2 000 за 8 лет, то какой была средняя годовая ставка?

Используемая функция: СТАВКА (кпер; плт; пс; бс; тип; предположение)

Эта формула возвращает 9,050773%:

=СТАВКА(8;0;-1000;2000;0)

Этот пример рассматривается с точки зрения вкладчика, таким образом, аргумент пс отрицательный, а аргумент (получаемые средства) бс положительный. Поскольку срок выражался в годах, полученное значение — эффективная годовая ставка.

ПРИМЕР 3

У меня на депозитном счету \$100 000, вложенные под 14% годовых. Сколько времени потребуется для того, чтобы я стал миллионером?

Используемая функция: КПЕР (ставка; плт; пс; бс; тип)

Эта формула возвращает 17,573:

=КПЕР(14%;0;-100000;1000000;0)

Этот пример рассмотрен с точки зрения вкладчика. Таким образом, аргумент пс отрицателен и аргумент (поступление \$1 миллиона) бс положителен. Поскольку ставка представляется в годовых процентах, то эффективный срок тоже выражается в годах.

ПРИМЕР 4

У меня на депозитном счету \$10 573,45, положенные под 1% ежемесячно. Счет открыт 12 месяцев назад. Каков начальный вклад?

Используемая функция: ПС (ставка; кпер; плт; бс; тип)

Эта формула возвращает -\$9 383,40:

=ПС(1%;12;0;10573,45;0)

Без регулярных выплат аргумент плт равен 0 и тип аргумента несущественен. Так как \$10 573,45 уже есть на счету, то аргумент бс получает знак “плюс”, а подсчитанное начальное значение отрицательное.

ПРИМЕР 5

Если я вношу \$300 ежемесячно (начиная с сегодня) на счет, положенный под 1% ежемесячно, сколько у меня будет на счету через 2 года?

Используемая функция: БС (ставка; кпер; плт; пс; тип)

=БС(1%;24;-300;0;1)

Эта формула возвращает \$8 172,96:

В этом примере срок указывается в годах, но процент и выплаты ежемесячные. Это требует предварительного преобразования единиц времени. Наиболее эффективный способ — преобразовать годы в месяцы. Другой вариант — преобразовать процентную ставку в ежегодную эффективную ставку, а потом преобразовать \$300 в эквивалентную сумму за год. Это должно привести к тому же результату, хотя расчеты в данном случае более громоздкие.

Обратите внимание, что начало выплат с “сегодня” характеризует авансовые платежи. Следовательно, тип аргумента равен 1. Никакой начальный взнос не указан, поэтому аргумент пс равен 0.

Во всех предыдущих примерах вопросы можно перефразировать таким образом, что отрицательное значение станет положительным, а положительное — отрицательным. Так, пример 1 может быть перефразирован следующим образом.

ПРИМЕР 6

Если я занимаю \$1 000 на три года под 7% годовых, то сколько всего я должен выплатить?

Используемая функция: БС (ставка; кпер; плт; пс; тип)

Это формула возвращает -\$1 225,04:

$$=БС(7\%; 3; 0; 1000; 0)$$

Здесь вопрос ставится с точки зрения заемщика, поэтому формула изменена так, что начальный вклад (аргумент пс) положительный. Регулярные выплаты не производятся, так что аргумент плт равен 0. Без выплат тип аргумента несущественен.

Примеры 2–5 можно также перефразировать, например, вкладчик становится заемщиком, а заемщик становится вкладчиком.

ПРИМЕР 7

Если с \$1 000 вклад увеличился до \$3 000 по истечении восьми лет, то какой была средняя годовая ставка?

Используемая функция: СТАВКА (кпер; плт; пс; бс; тип; предположение)

Это формула возвращает 14,720269%:

$$=СТАВКА(8; 0; -1000; 3000; 0)$$

Этот пример рассмотрен с точки зрения вкладчика. Таким образом, аргумент пс отрицателен, а аргумент бс положителен. Из-за того, что срок выражался в годах, то возвращаемое значение — эффективная годовая ставка. Без регулярных выплат аргумент плт равен 0, а тип аргумента несущественен.



Важная особенность финансовых вычислений состоит в том, что их можно обратно проверить для подтверждения результата. Это можно сделать без электронных таблиц, используя обычный калькулятор, основную формулу или другую функцию.

Следующие действия демонстрируют метод проверки результата 14,720269%, полученного в последнем примере.

1. Вычислите, сколько с суммы \$1 000 накапливается за 8 лет при подсчитанной ставке.

Формула возвратит \$3 000:

$$=БС(14,720269\%; 8; 0; -1000; 0)$$

2. Вычислите начальное значение для вклада \$3 000, открытого с указанной ставкой восемь лет назад. Следующая формула возвращает -1 000:

$$=ПС(14,720269\%; 8; 0; 3000; 0)$$

3. Вычислите, сколько времени потребуется, чтобы вклад с \$1 000 увеличился до \$3 000 при неизменной ставке. Следующая формула возвращает восемь:

$$=КПЕР(14,720269\%; 0; -1000; 3000; 0)$$

4. Вычислите будущую стоимость, используя следующую формулу, которая возвращает 14,720269%:

$$=(3000 / -1000) ^ (1/8) - 1$$

Методика перекрестной проверки заключается в таком сравнении вычисленного значения с исходными данными, при котором разница составляет нуль. Во всех предыдущих проверках вычитание исходных данных из вычисленных приводит к нулевому результату. Если в результате проверки всех рассчитываемых на рабочей книге результатов получены нули, то сумма проверок также будет равняться нулю. Хотя это будет неточный нуль, благодаря существованию погрешности округления.

Сложные задачи накопления

В этом разделе описаны четыре примера сложных задач накопления. Существует два типа сложных задач накопления:

- ◆ Задачи, которые имеют ненулевые значения любых двух ключевых параметров (начальное значение, платеж и будущее значение) и требуют поиска третьего параметра.
- ◆ Задачи, которые имеют ненулевые значения всех трех параметров (начальное значение, платеж и будущее значение) и требуют выполнения функций СТАВКА или КПЕР.

ПРИМЕР 8

С начальным балансом \$5 500 и ежемесячной выплатой \$500 (в конце каждого месяца), сколько я могу накопить за срок более трех лет при 0,75% в месяц?

На рис 11.2 показан этот пример, открытый на рабочем листе.

	A	B	C	D	E	F	G	H	I
1	Пример 8								
2	Задача								
3	При начальном балансе \$5 500 и ежемесячных выплатах \$500 (в конце месяца) сколько накопится на счете через три года при месячной ставке 0,75%?								
4	Функция								
5	BS(ставка,кпер,плт,пс,тип)								
6	Данные задачи								
7	ставка		0,750%						
8	кпер		36						
9	плт		-\$ 500,00						
10	пс		-\$ 5 500,00						
11	тип		0						
12	Ответ								
13	бс			\$27 773,91					

Рис. 11.2. Вычисление будущего значения

Используемая функция: БС (ставка; кпер; плт; пс; тип)

Эта формула возвращает \$27 773,91:

=БС(,75%;36;-500;-5500;0)

Отрицательный знак аргумента пс может поставить вас в тупик из-за того, что он представляет собой текущий баланс. Однако, поскольку рассматривается будущий баланс, текущее значение необходимо рассматривать как вклад. Выплаты и сроки указываются на ежемесячной основе; таким образом, трехлетний срок необходимо преобразовать в 36 месяцев. БС возвращает положительное значение, что указывает на увеличение вклада.

ПРИМЕР 9

Мой остаток на счету пять лет назад составлял \$25 000. Я добавлял \$4 500 в конце каждого года. Сейчас баланс равен \$70 000. Какой была моя среднегодовая ставка?

Используемая функция: СТАВКА (кпер; плт; пс; бс; тип; предположение)

Эта формула возвращает 10,9382%:

=СТАВКА(5;-4500;-25000;70000;0;0)



СТАВКА — это особенно мощная функция, так как решение поставленной задачи получается только в результате итерации. Только иногда приходится добавлять предполагаемую ставку в качестве шестого аргумента. Если его опустить, то Excel по умолчанию предложит 0.

ПРИМЕР 10

У меня на счету задолженность банку \$12 000 и я вношу \$1 000 в конце каждого месяца. Сколько времени потребуется на то, чтобы я стал миллионером, если я выплачиваю пеню 0,6% за месяц?

Используемая функция: КПЕР (ставка; плт; пс; бс; тип)

Следующая формула возвращает 337,78 месяцев:

=КПЕР(0,6%;-1000;12000;1000000;0)

Обратите внимание, что вопрос строится так, что задолженность на текущем счету является вкладом. Таким образом, он требует отрицательного знака для аргумента пс.

Если задолженность на текущем счету рассматривается как заем, будущее значение должно быть положительным. В таком случае требуются два независимых вычисления. Сначала надо вычислить время до получения нулевого баланса, а потом необходимо рассчитать время для получения \$1 миллиона на счету.

Используя курс 0,8% для кредита и 0,6% для вклада. Формула возвращает 337,96 месяцев:

=КПЕР(0,8%;-1000;12000;0;0)+КПЕР(0,6%;-1000;0;1000000;0)

ПРИМЕР 11

Я вношу \$1 000 в месяц (в конце периода) и собираюсь это делать на протяжении последующих десяти лет. Если предстоит накопить \$1 000 000, сколько я должен сейчас при месячной ставке 0,7%?

Используемая функция: ПС (ставка; кпер; плт; бс; тип)

Эта формула возвращает -\$351 972,24:

=ПС(0,7%;120;-1000;1000000;0)

Нужно преобразовать годы в месяцы для гарантии соответствия аргументов ставка, плт и кпер.

Проработав первые 11 примеров, можно получить общий алгоритм вычислений:

1. Определите требуемую функцию.
2. Определите направление денежных потоков и используемые для расчетов функции.
3. Убедитесь, что единицы измерения времени в функциях СТАВКА, КПЕР и ПЛТ одни и те же (или преобразуйте значения так, чтобы сделать их равными).
4. Вставьте аргументы в правильном порядке (предпочтительно с помощью ссылок на ячейки).
5. Проанализируйте полученный результат.
6. Определите, какая функция или вычисления требуются для проведения перекрестной проверки.
7. Убедитесь в отсутствии ошибок.

Простые задачи дисконтирования

Можно воспринимать заем как “обратное накопление”. Вместо определения будущего значения по начальному вкладу и ставке давайте вычислим значение начального вклада по его будущей величине.

Как и в случае с накоплениями, здесь задачи решаются с помощью двух или трех значений ПС, БС и ПЛТ. Если в вычисления вовлечены только два параметра, то этот вариант называется *простым дисконтированием*, а если все три, то — *сложным дисконтированием*.

ПРИМЕР 12

Какой начальный вклад предполагает получение \$25 000 после пяти лет при ставке 6,5% годовых?

На рис. 11.3 показан этот пример, открытый на рабочем листе.

Рис. 11.3. Вычисление начального значения

Используемая функция: ПС (ставка ; кпер ; плт ; бс ; тип)

Эта формула возвращает $-\$18\,247,02$:

$=\text{ПС}(6,5\%; 5; 0; 25000; 0)$

Обратите внимание на логику знаков. Если вы собираетесь получить на вкладе деньги, то БС — положительный аргумент. Чтобы получить в будущем положительное значение нам надо вначале положить в банк определенную сумму, т.е. использовать в формуле отрицательный аргумент. Без выплат тип аргумента несущественен.

Точность вычисления гарантируется перекрестной проверкой ответа другой функцией. В этом случае можно проверить, накопится ли \$25 000 после пяти лет на вкладе \$18 247,02 при ставке 6,5%, или нет. Следующая формула перекрестной проверки действительно возвращает \$25 000:

$=\text{БС}(6,5\%; 5; 0; -18247,02; 0)$

ПРИМЕР 13

Ежегодная арендная плата за недвижимость на следующие 25 лет составляет \$25 000. Если считать аренду займом с 8% годовых, то сколько надо заплатить сейчас, чтобы рассчитаться через 25 лет?

Используемая функция: ПС (ставка; кпер; плт; бс; тип)

Следующая формула возвращает $-\$266\,869,40$:

$=\text{ПС}(8\%; 25; 25000; 0; 0)$

Этот результат можно проверить, используя функцию СТАВКА. Она возвращает значение 8,00%:

$=\text{СТАВКА}(25; 25000; -266869,40; 0; 0)$

Как правило, выплаты за аренду недвижимого имущества производятся авансом. В таком случае пример 13 необходимо изменить, установив параметр тип равным 1.

ПРИМЕР 14

Предположите, что выплата в примере 13 суммы \$25 000 проводится вечно. При ставке 8%, сколько необходимо заплатить сначала?

Это пример задачи дисконтирования, которую Excel не может решить, используя стандартные функции. Проблема состоит в том, что нельзя использовать термин “вечность” в качестве аргумента кпер. Решение состоит в том, чтобы использовать очень длительный период, например 1 000 лет. Получаемый результат наиболее точен для большинства задач.

Используемая функция: ПС (ставка; кпер; плт; бс; тип)

Следующая формула возвращает $-\$312\,500,00$:

$=\text{ПС}(8\%; 1000; 25000; 0; 0)$

Другой вариант — использовать формулу для вычисления текущего значения:

$\text{ПС} = \text{ПЛТ} / \text{СТАВКА}$

В нашем примере, следующая формула возвращает $\$312\,500,00$:

$=25000 / 0,08$

Обратите внимание, что знак разный, так как в последней формуле знаки вообще не учитывались.

Если сумма выплачена авансом, то расчеты адаптируются в результате присвоения 1 аргументу тип. Следующая формула возвращает $-\$337\,500,00$:

$=\text{ПС}(8\%; 1000; 25000; 0; 1)$

Формула проверки полученного результата также изменяется:

$\text{ПС} = \text{ПЛТ} * (1 + \text{СТАВКА}) / \text{СТАВКА}$

В нашем примере следующая формула возвращает $\$337\,500,00$:

$=25000 * (1 + 0,08) / 0,08$

ПРИМЕР 15

Имущество с текущей стоимостью \$2 000 000 продается на правах аренды с номинальной арендной платой сроком на пять лет. Покупатель оплатил \$1 750 000. Не принимая во внимание рост стоимости, какой была учетная ставка?



Номинальная арендная плата — плата, которая взимается с клиента в учет покупки жилья.

Используемая функция: СТАВКА (кпер; плт; пс; бс; тип; предположение)

Эта формула возвращает 2,706609%:

$=\text{СТАВКА}(5; 0; -1750000; 2000000; 0)$

Выплата сегодня представляет собой отрицательное текущее значение. Стоимость после 5 лет — положительная.

Для проверки ответа используйте следующую формулу (которая возвращает \$2 000 000,03):
=БС(2,706609%;5;-1750000;0)

Ошибка округления вызывается строго заданной ставкой до шести десятичных разрядов. Обычно аргумент задается ссылкой на ячейку, содержащую переменное (непостоянное) значение.

ПРИМЕР 16

Недвижимость, проданная на правах аренды, оценена в \$230 000. Аренда длится четыре года и выплачивалась каждый месяц наперед в размере \$6 000. Если предположить, что доход должен равняться 0,75%, то какая прибыль была бы получена от аренды в результате транзакции? Прибыль от аренды — это значение расчетной арендной платы минус выплаченная сумма.

Используемая функция: ПЛТ (ставка; кпер; пс; бс; тип)

Следующая формула возвращает \$5 680,95:
=ПЛТ(0,75%;48;-230000;0;1)

Добавив выплаченную наперед арендную плату (\$6 000) получим значение \$11 680,95.

Сложные задачи дисконтирования

Сложные задачи дисконтирования включают использование всех трех параметров: начального значения, платежей и будущего значения. Примеры сложного дисконтирования в этом разделе по существу обратны сложным задачам накопления.

ПРИМЕР 17

Если использовать учетную ставку 0,75% в месяц, сколько необходимо выплатить в начале за имущество, которое по оценке будет стоить \$5 000 000, при ежемесячной выплате \$25 000 в течение пяти лет?

Используемая функция: ПС (ставка; кпер; плт; бс; тип)

Следующая формула возвращает -\$4 406 865,34:
=ПС(0,75%;60;25000;5000000;1)

В этом примере в качестве единицы времени используется месяц, а выплаты производятся ежемесячно. Таким образом, аргумент кпер преобразован в месяцы.

Можно проверить это вычисление с помощью функции СТАВКА. Следующая формула возвращает 0,75%:

=СТАВКА(60;25000;-4406865,34;5000000;1)

ПРИМЕР 18

Я оплатил \$1 200 000 за имущество с помощью расчетной арендной платы \$12 000 в месяц. Если я продам на протяжении пяти лет за \$1 500 000, какую суммарную прибыль я получу?

Требуемая функция: СТАВКА (кпер; плт; пс; бс; тип; прогноз)

Следующая формула возвращает 1,29136%:
=СТАВКА(60;12000;-1200000;1500000;1)

Этот результат можно проверить с помощью функции ПС. Следующая формула возвращает -\$1 200 000,00:

=ПС(1,29136%;60;12000;1500000;1)

Предполагается, что аренда выплачивается ежемесячно авансом в течение пяти лет. Это задача решается благодаря преобразованию годов в месяцы. Таким образом, формула возвращает ежемесячную процентную ставку.



Обратите внимание, ежемесячная арендная плата не преобразуется в ежегодную арендную плату. Это происходит, потому что сумма аренды \$12 000 в месяц отличается от суммы \$144 000 в год. Чтобы получить эквивалентную ежегодную сумму, надо знать ставку, являющуюся частью данных, которую мы пытаемся вычислить.

ПРИМЕР 19

Имущество приобретено за \$1 600 000. Расчетная арендная плата составляла \$10 000 и выплачивалась авансом каждый месяц. Если я хочу получить доход равный 1% в месяц, какая должна быть стоимость имущества через 5 лет, если я планирую его продать?

Используемая функция: БС (ставка; кпер; плт; пс; тип)

Эта формула возвращает \$2 081 851,05:

=БС(1%; 60; 10000; -1600000; 1)

Этот результат можно проверить, используя следующую формулу (которая возвращает – \$1 600 000):

=ПС(1%; 60; 10000; 2081851,05; 1)

Задачи амортизации

Амортизация — это термин, характеризующий выплату займа. В этой главе, фактически, уже описано большинство операций ее вычисления, но рассматривались они исключительно с точки зрения задачи накопления.

ПРИМЕР 20

Какие выплаты необходимо сделать по займу \$200 000, взятому на 10 лет под 0,5% месячных?

Этот пример показан на рис. 11.4.

	A	B	C	D	E	F	G	H	I
1	Пример 20								
2	Задача								
3									
4	Определите выплаты по займу \$200 000 сроком 10 лет и месячной ставкой 0,5%								
5									
6									
7	Функция								
8									
9	ПЛТ(ставка;кпер;пс;бс;тип)								
10									
11									
12	Данные задачи								
13									
14	ставка			0,5%					
15	кпер			120					
16	пс			\$200 000,00					
17	тип			0					
18									
19	Ответ								
20									
21	ПЛТ			(\$2 220,41)					
22									

Рис. 11.4. Вычисление выплаты займа

Используемая функция: ПЛТ (ставка; кпер; пс; бс; тип)

Следующая формула возвращает \$2 220,41:

=ПЛТ (0,5%;120;200000;0;0)

Можно проверить результат с помощью функции ПС для начальной суммы займа. Следующая формула возвращает \$200 000:

=ПС (0,5%;120;-2220,41;0;0)

В этом примере заем полностью выплачивается через 10 лет, и аргумент бс равен нулю. Также обратите внимание, что выплаты проводятся ежемесячно, поэтому указана ежемесячная ставка. Таким образом, 10-летний срок преобразуется в месяцы.

ПРИМЕР 21

Я могу позволить себе ежемесячные выплаты \$2 500 со ставкой 0,45% (в месяц) в течение 20 лет. Сколько я могу занять, чтобы полностью погасить заем?

Используемая функция: ПС (ставка; кпер; плт; бс; тип)

Эта формула возвращает \$366 433,74:

=ПС (0,45%;240;-2500;0;0)

Обратите внимание, при расчете займа всегда предполагается, что выплаты проводятся с задолженностью и тип аргумента равен 0. Также примите к сведению, что процентная ставка (и платежи) рассматриваются в расчете на месяц. Таким образом, 20-летний срок необходимо преобразовать в месяцы.

Можно проверить ответ, рассчитав ставку для займа \$366 433,74, предоставляемой на 240 месяцев. Следующая формула возвращает 0,45%:

=СТАВКА (240;-2500;366433,74;0;0)

ПРИМЕР 22

В настоящий момент я имею задолженность \$150 000 по займу, и выплачиваю \$1 900 в месяц. Текущая процентная ставка равна 0,45% в месяц. Сколько времени потребуется для выплаты всего долга?

Используемая функция: КПЕР (ставка; плт; пс; бс; тип)

Следующая формула возвращает 97,76:

=КПЕР (0,45%;-1900;150000;0;0)

Из-за того, что выплаты производятся ежемесячно и ставка определяется в расчете на один месяц, формула возвращает период амортизации в месяцах. Этот ответ, хотя математически и правилен, имеет редкое практическое применение. Выплаты действительно производятся точно раз в месяц. Это подразумевает, что заем будет выплачен к 98 месяцу. Фактически, перед вами стоит выбор: произвести дополнительную оплату в конце 97 месяца, или сделать уменьшенную оплату на 98 месяц. Каждый вариант подсчитан с помощью функции БС.

Чтобы вычислить дополнительную оплату в конце 97 месяца, подсчитайте сумму с помощью следующей формулы (которая возвращает -\$1 429,85):

=БС (0,45%;97;-1900;150000;0)

Таким образом, заключительная оплата после 97 месяца равна -\$3 329,85 (то есть, нормальный платеж -\$1 900 плюс -\$1 429,85).

Чтобы вычислить уменьшенную оплату в 98 месяце, используйте эту же формулу (которая возвращает \$463,72):

=БС (0,45%;98;-1900;150000;0)

Таким образом, конечная оплата через 98 месяцев равна -\$1 436,28 (то есть, нормальный платеж -\$1,900 плюс \$463,72).



Часто проблема возникает там, где платеж меньше суммы процента. В рассмотренном примере, неоплаченный долг равен \$150 000, а процент в первом месяце составляет \$675 ($\$150\,000 \cdot 0,45\%$). Если платеж меньше, чем процент, то задолженность будет возрастать, а заем увеличится до бесконечности (скорее будет казаться растущим до бесконечности). Если это произойдет, функция КПЕР возвратит сообщение об ошибке #ЧИСЛО!.

ПРИМЕР 23

Соглашение о потребительском займе предоставляет мне кредит \$1 000 с оплатой \$100 в месяц в течение 12 месяцев. Какая процентная ставка?

Используемая функция: СТАВКА (кпер; плт; пс; бс; тип; предположение)

Следующая формула возвращает 3,503153%:

$$=СТАВКА(12; -100; 1000; 0; 1)$$

Перед тем как вы оцените, насколько это соглашение великодушно, запомните, что выплаты производятся ежемесячно. Таким образом, результат равен ежемесячной процентной ставке! Эффективная ежегодная ставка равна 51,16%, подсчитанная так, как указано ниже:

$$=(1+0,03503153)^{12}-1$$

Ежегодная ставка, основанная на номинальной ежемесячной основе, возвращает 42,05%, вычисленной, как указано ниже:

$$=3,503153 \cdot 12$$



Существует большая разница между эффективной ежегодной ставкой и номинальной годичной ставкой, вычисленной на ежемесячной основе. Разница увеличивается с уровнем используемых ставок.

ПРИМЕР 24

Я беру заем со вздутыми выплатами размером \$300 000 на 15 лет, с выплатой только \$100 000 на ежемесячной основе. В конце срока выплачивается баланс \$200 000. Процентная ставка равна 0,4% в месяц, а платежи производятся ежемесячно. Какой размер ежемесячной выплаты?

Привычный тип займа (используемый для увеличения занимаемой суммы) это так называемый заем со вздутыми выплатами. Заем делится на две части: 1) выплачивается часть, которая полностью погашается до конца срока, 2) часть “вздутых выплат”. На протяжении срока займа для “вздутой” части займа выплачивает только процент. Начальная стоимость “вздутой” части займа оплачивается, как сумма в конце займа.

Для проведения расчетов займа со вздутыми выплатами в формулах ПС, ПЛТ, СТАВКА и КПЕР используется правильно определенный параметр бс.

Используемая функция: ПЛТ (ставка; кпер; пс; бс; тип)

Следующая формула возвращает -\$1 580,41:

$$=ПЛТ(0,4\%; 180; 300000; -200000; 0)$$

Обратите внимание, что сумма общего займа (\$300 000) определяется параметром пс.

Чтобы проверить полученное значение достаточно рассчитать значение начального займа (пс). Следующая формула возвращает значение \$299 999,43 (ошибка округления).

$$=ПС(0,4\%; 180, -1580,41; -200000; 0)$$

Выплаты при “вздутом” займе можно сравнить с выплатами при обычном займе. Следующая формула возвращает \$202 509,64 (обычный заем).

$$=ПС(0,4\%; 180; -1580,41; 0; 0)$$

Выплаты при обычном займе \$300 000 составляют -\$2 341,24. Это значение рассчитывается по формуле.

$$= \text{ПЛТ}(0, 4\%; 180; 300000; 0; 0)$$

Последний пример амортизации можно адаптировать под “вздутый” заем, добавив в функции ПС, ПЛТ, КПЕР и СТАВКА ненулевой аргумент БС.

“Вздутый” заем также рассчитывается и с помощью функции БС. Используя ее, внимательно анализируйте знаки подставляемых в нее аргументов. Если функция БС возвращает положительное значение, то это значит, что заем переплачен. Отрицательное значение выражает “вздутую часть” займа.

Обычно подобные расчеты разбиваются на два этапа. Сначала рассчитывается нормальная амортизация. Затем подсчитывается размер “вздутой части” займа. В следующем примере эта операция рассмотрена детально.

Пример 25

Если банк настаивает на амортизации \$200 000 по займу, то какую дополнительную сумму я могу одолжить при “вздутом” займе и ежемесячных выплатах \$3 000? Срок займа 10 лет, а ставка 0,4% в месяц.

Используемая функция: ПЛТ (ставка; кпер; пс; бс; тип)

Первый шаг — вычислить оплату обычного амортизационного займа \$200 000. Следующая формула возвращает -\$2 101,81:

$$= \text{ПЛТ}(0, 4\%; 120; 200000; 0; 0)$$

Если вы можете выплачивать \$3 000, то дополнительная сумма \$898,19 может быть оплачена как процент “вздутого” займа (то есть, \$3 000 – \$2 101,81). Элемент вздутых выплат можно подсчитать по известной сумме процентной ставки. Следующая формула, представляющая собой элемент “вздутых выплат”, возвращает \$224 546,88:

$$= 898,19 / 0,4\%$$

Вычисление можно проверить, вычислив выплаты для полного займа \$424 546,88 с элементом вздутых выплат \$224 546,88. Следующая формула возвращает -\$3 000:

$$= \text{ПЛТ}(0, 4\%; 120; 424546,88; -224546,88; 0)$$

Преобразование процентных ставок

Предыдущие примеры представлены для того, чтобы наглядно проиллюстрировать взаимосвязь между процентной ставкой, частотой выплат и общей суммой. Часто, однако, интерпретировать финансовую задачу не так просто. Существует две ситуации, в которых приходится преобразовывать процентную ставку:

- ◆ Когда надо произвести вычисления, в которых используются значения частоты платежей или количества периодов выплат, а обрабатываемая величина не соответствует частоте платежей или периоду времени.
- ◆ Когда результаты вычислений, в которых используются значения частоты платежей или количества периодов выплат, нужно выразить в других единицах времени.

Для создания точных формул важно понять принцип эквивалентности процентных ставок. Попросту говоря, любая текущая процентная ставка на протяжении одного периода времени эквивалентна другой процентной ставке на протяжении другого периода времени.

Методы указания процентных ставок

Существует три часто используемых метода указания процентных ставок:

- ◆ *Номинальная ставка.* Процент указывается на ежегодной основе, в зависимости от частоты выплат за год. Например, часто используемая годовая процентная ставка, равная 6%, при перерасчете равняется 0,5% в месяц (с регулярными выплатами).
- ◆ *Ежегодная эффективная ставка.* Процентная ставка, представляющая собой сумму, насчитываемую за один год. Например, при 10%-ой ежегодной эффективной ставке, с суммы \$1 000 насчитывается \$100 в конце каждого года.
- ◆ *Периодическая эффективная ставка.* Процентная ставка, представляющая собой сумму, насчитываемую на протяжении периода времени меньше года. Например, со ставкой 3% за полгода с суммы \$300 насчитывается \$9 каждые шесть месяцев.

Процентная ставка характерна тем, что использование одного из этих трех коэффициентов позволяет провести перерасчет для любого другого коэффициента. Например, рассмотрите процентную ставку 1% в месяц для вклада \$100. В первом месяце на вклад начисляется процент, равный \$1. Если процент кредита не изымается, то он добавляется к начальной сумме, и последующий процент основывается на новом балансе. 1% ежемесячная процентная ставка эквивалентна 12,6825% за год (эффективная годовая ставка). Она вычисляется с помощью следующей формулы:

$$= (1 + 0,01)^{12} - 1$$

Другой пример номинальной ставки — процентная ставка, указываемая как 6% за год, насчитываемая раз в квартал. Это означает, что 1,5% (то есть, 6%/4) выплачивается или начисляется каждые три месяца.

Большинство банков и финансовых учреждений указывают процент на ежемесячной основе. Однако, сообщая о возвратах с вкладов или при сравнении процентных ставок, часто указывают ежегодные эффективные проценты, делающие более легким сравнение курсов. Например, мы знаем, что 12% годовых, выплачиваемые ежемесячно, больше, чем 12% годовых, выплачиваемые ежеквартально, но мы не знаем (без промежуточного преобразования) насколько.

Преобразование процентных ставок с помощью встроенных финансовых функций

Как вы видите, существует 10 различных преобразований при перерасчете между номинальной, ежегодной эффективной и периодической эффективной системами начислений.

Используя надстройку финансовых функций, можно вводить функцию на рабочий лист вручную или использовать диалоговое окно вставки функций Excel (функции размещаются в категории Финансовые). Табл. 11.1 содержит список 10 функций преобразований процентных ставок, содержащихся в надстройке финансовых функций. В таблице также показан эквивалент формуле Excel.

Имена функции и аргументы сначала могут привести в тупик, но вы вскоре получите понятие о них. Имя каждой функции состоит из трех частей:

- ◆ Процентная ставка, которую вы имеете (Eff, AnnEff, или Nom). Обратите внимание, что частота исходных эффективных и номинальных ставок обозначается как x .
- ◆ Знак связи, который определяет направление преобразования, — это символ подчёркивания ($_$).

- ◆ Желаемая (конечная) процентная ставка (*Eff*, *AnnEff*, или *Nom*). Опять-таки, частота сложных процентов обозначается *x* (если она такая же, как и частота имеющейся ставки) или *y* (если другая).

Таблица 11.1. Пользовательские функции VBA преобразования обычной ставки

Функция надстройки	Описание	Эквивалентная формула Excel
<i>Effx_Nomx</i> (<i>Effx</i> , <i>Freqx</i>)	Преобразует эффективную ставку на протяжении периода меньше года в эквивалент номинальной ставки для заданной частоты.	(отсутствует)
<i>Effx_AnnEff</i> (<i>Effx</i> , <i>Freqx</i>)	Преобразует эффективную ставку для частоты меньше года в эквивалент ежегодной эффективной ставки.	ЭФФЕКТ(номинальная_ставка;кол_пер)
<i>Effx_Nomy</i> (<i>Effx</i> , <i>Freqx</i> , <i>Freqy</i>)	Преобразует эффективную ставку для частоты меньше года в эквивалент номинальной ставки для отличной от исходной частоты.	НОМИНАЛ(эффект_ставка;кол_пер)
<i>Effx_Effy</i> (<i>Effx</i> , <i>Freqx</i> , <i>Freqy</i>)	Преобразует эффективную ставку для частоты меньше года в эквивалент эффективной ставки для другой частоты, которая также меньше года.	НОМИНАЛ(ЭФФЕКТ(номинальная_ставка;кол_пер1) кол_пер2)/ кол_пер2
<i>Nomx_Effx</i> (<i>Nomx</i> , <i>Freqx</i>)	Преобразует номинальную ставку в эквивалент эффективной ставки для частоты номинальной ставки.	(отсутствует)
<i>Nomx_AnnEff</i> (<i>Nomx</i> , <i>Freqx</i>)	Преобразует номинальную ставку в эквивалент ежегодной эффективной ставки.	ЭФФЕКТ(номинальная_ставка;кол_пер)
<i>Nomx_Nomy</i> (<i>Nomx</i> , <i>Freqx</i> , <i>Freqy</i>)	Преобразует номинальную ставку для частоты в эквивалент номинальной ставки для другой частоты.	НОМИНАЛ(ЭФФЕКТ(номинальная_ставка;кол_пер1) кол_пер2)
<i>Nomx_Effy</i> (<i>Nomx</i> , <i>Freqx</i> , <i>Freqy</i>)	Преобразует номинальную ставку для частоты в эквивалент эффективной ставки, для частоты меньше года, который не является частотой данной номинальной ставки.	НОМИНАЛ(ЭФФЕКТ(номинальная_ставка;кол_пер1) кол_пер2)/ кол_пер2
<i>AnnEff_Effx</i> (<i>AnnEff</i> , <i>Freqx</i>)	Преобразует ежегодную эффективную ставку в эквивалент эффективной ставки для частоты меньше года.	НОМИНАЛ(номинальная_ставка;кол_пер)/кол_пер
<i>AnnEff_Nomx</i> (<i>AnnEff</i> , <i>Freqx</i>)	Преобразует ежегодную эффективную ставку в эквивалент номинальной ставки.	=НОМИНАЛ(номинальная_ставка; кол_пер)

Упорядочение аргументов выполняется также просто:

- ◆ Первый аргумент — всегда имеющаяся процентная ставка.
- ◆ Второй аргумент — всегда частота. Обратите внимание, что каждая функция преобразования использует аргумент *Freqx*, заданный вторым.
- ◆ Если имеется вторая известная частота, отличная от *x* или ежегодная, то используется третий аргумент, *Freqy*.

Эффективная стоимость займов

Кредитные учреждения обычно рекламируют “лучшие” показатели процентных ставок, чтобы убедить вас в том, что они настолько низки, насколько это вообще возможно. Сообразительный заемщик может преобразовать представленные процентные значения и определить

действительную стоимость займа. Единственное надежное и убедительное сравнение — проанализировать эффективную стоимость с точки зрения ежегодной эффективной ставки или любой другой ставки, например, ежегодной номинальной ставки со сложными процентами, начисляемыми ежемесячно.

В этом разделе представлено четыре примера, демонстрирующие вычисление эффективной стоимости займов.

Воздействие платежей на эффективную ставку

В дополнение к проценту по займу, банки часто требуют оплату за открытие и обслуживание счета. Эти выплаты увеличивают эффективную стоимость займа. Но насколько?

ПРИМЕР 26

Банк указывает номинальную ставку займа 7% со сложными процентами ежемесячно, и вы заинтересованы в займе \$150 000 больше, чем на 10 лет, с ежемесячными выплатами. Банк предоставляет заем со ставкой 2% за открытие счета и дополнительной оплатой \$25 на обслуживание счета. Какая ежегодная эффективная стоимость займа?

Рабочий лист, который предоставляет решение этой задачи, содержится на Web-узле. Известные сведения вводятся в раздел данных рабочего листа. Табл. 11.2 представляет список ключевых формул, которые выполняют вычисления. Для ясности, формулы содержат действительные значения, а не ссылки на ячейки.

Таблица 11.2. Формулы, используемые на рабочем листе, показанном на рис. 11.5

Ячейка	Вычисление	Формула (действительные значения)
B16	плата за открытие счета	= \$150.000 * 2%
B17	эффективный заем	= \$150.000 - \$3,000
B18	строк займа	=10 * 12
B19	период ставки	=Nomx_Effx(7%;12)
B20	выплаты займа	=ПЛТ(0,583333%;120;150000;0;0)
B21	выплаты займа + оплата	= -\$1 741,63 - \$25
B22	эффективная стоимость займа	=СТАВКА(120;-1766,63;147000;0;0)



В ячейке B19 используется пользовательская функция VBA.

Платежи основываются на сумме займа \$150 000, но эффективная стоимость основывается на том факте, что после вычитания платы за открытие заемщик получает только \$147 000. Точно также обстоят дела с фактическими платежами свыше суммы обслуживания счета.

Воздействие дополнительных издержек изменяется соответственно сроку: чем короче срок, тем сильнее воздействие.

Займы с единообразной ставкой

Много потребительских соглашений о кредитовании заключаются для займа, в котором процент займа добавляется к самому займу, а выплаты насчитываются как сумма займа плюс

процент, разделенный на количество платежей. Для расчета подобного займа можно использовать стандартную функцию СТАВКА Excel.

ПРИМЕР 27

*Потребитель финансирует покупку автомобиля с одинаковой ставкой займа \$15 000 на 18 месяцев. Процентная ставка займа 10% * 1,5 этой суммы добавляется к займу, и он ежемесячно выплачивает 1/18 полученной суммы в течение 18 месяцев. Какая эффективная стоимость займа?*

Наиболее легкий путь для решения этой задачи — использовать функцию Efx_AnnEff (пользовательская функция VBA). Следующая формула возвращает 3,62%:

$\text{=Efx_AnnEff(СТАВКА(18; -17250/18; 15000; 0; 1); 12)}$

Обратите внимание, что, если срок такого займа только 12 месяцев, процентная ставка немного больше удвоенной общей ставки. Большинство государств и стран издало законы, обязывающие подобные соглашения займа иметь номинальную ежегодную ставку рассчитанную на ежемесячной основе, которая четко определена в соглашении займа.

Беспроцентные займы

Другое интересное вычисление — получение эффективной стоимости так называемого “беспроцентного” займа. При выполнении вычислений нужно знать, на каких условиях можно получить аналогичную услугу в другом месте (беспроцентного соглашения).

ПРИМЕР 28

Потребитель покупает аудиосистему с высоким качеством воспроизведения по цене в трейскаунте \$3 000 на “беспроцентных” условиях кредита в 12 месяцев с платежами авансом. Он мог бы купить идентичную систему за \$2 500 наличными или при обычном кредитовании. Какая эффективная стоимость этого займа?

Опять-таки, пользовательская функция Efx_AnnEff VBA обеспечивает наиболее простое решение. Эта формула возвращает 51,16%:

$\text{=Efx_AnnEff(СТАВКА(12; -(3000/12); 2500; 0; 1); 12)}$

Такие вычисления часто более сложные, когда эквивалентная цена в наличных денежных средствах субъективна (например, рынок б/у автомобилей).

Можно выполнить подобные вычисления для других типов соглашения, например: “Заплатите сегодня 25% и ничего не надо платить на протяжении 12 месяцев”. Снова следует установить эквивалентную наличную цену, а потом сравнить вычисления с той ценой, которая скорее раздувается розничным торговцем, предлагающим “беспроцентный” кредит.

Большинство государств и стран имеют законодательство, регулирующее потребительский тип кредитования, которое управляет уровнем процентных ставок. Во многих регионах единственное правило регулирования беспроцентных соглашений заключается в том, что розничный продавец не может предлагать тот же товар по наличной цене, отличной от указанной в “беспроцентном” соглашении.

Заем с “ежегодными” выплатами

Практика, которая внедряется в настоящее время, — рассчитать платежи на основе “ежегодной ставки”, и обязать заемщика выплачивать 1/12 полученной суммы ежемесячно. Расчет облегчен предварительно созданными таблицами ежемесячных платежей с кредита

\$1 000. Эта практика возымела большой успех частично из-за того, что подобный кредит имеет низкую рекламируемую ставку, чем номинальная или эффективная.

ПРИМЕР 29

Банк предлагает кредит \$100 000 со ставкой 7% на 10 лет, где ежемесячные платежи рассчитываются как 1/12 суммы, выплачиваемой ежегодно. Какая ежегодная эффективная стоимость?

Следующая формула (которая использует функцию `Effx_AnnEff`) возвращает 0,7522% (ежегодная эффективная ставка):

`Effx_AnnEff(СТАВКА(10*12; ПЛТ(7%; 10; 100000; 0; 0) / 12; 100000; 0; 0) ; 12)`

Вычисление процента и основных платежей

В этом разделе обсуждаются четыре функции Excel, которые позволяют:

- ◆ Вычислить процент или основные платежи (функции `ОСПЛТ` и `ПРПЛТ`)
- ◆ Вычислить суммарный процент или основные платежи между любыми двумя периодами времени

Использование функций `ОСПЛТ` и `ПРПЛТ`

Иногда необходимо знать (или просто любопытствовать), сколько от конкретного платежа составляет процент, и сколько составляет начальная сумма. Эта информация требуется для определения, например, влияния налога на процентную ставку. Если вы изучили примеры амортизации займа, то знаете, что сумма процента не постоянна во время действия займа. Процент уменьшается в то время, как начальная сумма возрастает.



Если вы уже создали план амортизации, эти функции не станут вам особенно полезны, так как вы можете просто обратиться к плану. Функции `ОСПЛТ` и `ПРПЛТ` используются, когда надо определить соотношение процент /начальный вклад для конкретного платежа.

Синтаксис этих двух функций приведен ниже (полужирным шрифтом выделены обязательные параметры):

`ОСПЛТ` (**ставка** ; **период** ; **кпер** ; **пс** ; **бс** ; тип)

`ПРПЛТ` (**ставка** ; **период** ; **кпер** ; **пс** ; **бс** ; тип)

Как и в случае функций амортизации, аргументы `ставка`, `период` и `кпер` имеют одни и те же приведенные единицы измерения времени. Если срок займа выражен в месяцах, аргумент `ставка` равняется проценту за месяц, а аргумент `период` (то есть, срок начисления процента) указывает конкретный месяц.

ПРИМЕР 30

Потребитель получает заем на покупку автомобиля (ежемесячные платежи) \$20 000 при ежегодной ставке 8%. Какой процент и начальная сумма заключительного платежа займа?

На рис. 11.5 показано решение, открытое на рабочем листе.

	A	B	C	D	E	F	G
1	Использование функции ОСПЛТ и ППЛТ						
2							
3	Пользовательские данные						
4	Объем займа	\$20 000,00					
5	Срок выплат в годах	3					
6	Количество выплат в год	12					
7	Эффективная годовая ставка	8%					
8	Тип выплат	0					
9							
10							
11	Предварительные вычисления						
12	Количество платежей	36					
13	Ставка за период	0,6667%					
14	Платеж	(\$626,73)					
15	Срок выплаты займа в периодах	36					
16							
17	Процент	(\$4,15)					
18	Основной платеж	(\$622,58)					
19							
20	Всего	(\$626,73)					

Рис. 11.5. На этом рабочем листе вычисляется процент и начальные параметры на протяжении любых периодов выплаты займа

Используемая функция: ППЛТ (ставка; период; кпер; пс; бс; тип)

Эта формула вычисляет процент последнего платежа и возвращает -\$4,15:

=ППЛТ (8%/12; 36; 36; 20000; 0; 0)

Следующая формула вычисляет основную сумму заключительного платежа и возвращает -\$622,58:

=ОСПЛТ (8%/12; 36; 36; 20000; 0; 0)

К концу срока займа практически вся выплата равняется основному капиталу. Чтобы сравнить это с первым периодом займа, поменяйте аргумент срок на 1. После этого формула возвратит -\$133,33 (процент) и -\$493,39 (основная сумма).



Можно проверить вычисления с помощью функции ПЛТ (возвращает общую сумму: процент плюс основной капитал). Следующая формула возвращает -\$626,73, что составляет сумму выплаты займа (и сумму результатов двух предыдущих формул):

=ПЛТ (8%/12; 36; -20000; 0)

Использование функций ОБЩДОХОД и ОБЩПЛАТ

Функции ОСНПЛАТ и ППЛАТ могут быть весьма полезны. Однако часто надо знать процент или начальный капитал последовательных выплат. В этом случае обычно используются функции ОБЩДОХОД и ОБЩПЛАТ. Эти функции нужны для создания ежегодных планов амортизации и установки правильного процента для налоговой декларации. Синтаксис этих функций показан ниже (обязательны все параметры):

ОБЩДОХОД (ставка; кпер; пс; начальн_период; конечн_период; тип)

ОБЩПЛАТ (ставка; кпер; пс; начальн_период; конечн_период; тип)



Эти функции доступны только при установке надстройки "Пакет анализа".

ПРИМЕР 31

Потребитель занимает сумму \$250 000, подлежащую выплате свыше 10 лет при 5,6% номинальной ежемесячной ставке со сложными процентами. Какая сумма процента и начальный капитал на первом году займа?

Следующая формула предназначена для определения начального платежа и возвращает \$13 512,31:

=ОБЩПЛАТ (Номх_Еффх (5,6%;12);10*12;250000;1;12;0)

Следующая формула возвращает \$19 194,42 (выплата процента):

=ОБЩДОХОД (Номх_Еффх (5,6%;12);10*12;250000;1;12;0)

Можно проверить эти результаты, используя функцию ПЛТ для вычисления выплат. Следующая формула возвращает \$32 706,74, что равно сумме предыдущих результатов:

=ПЛТ (Номх_Еффх (5,6%;12);10*12;250000;0;0) *12



Эти все формулы используют пользовательскую функцию Номх_Еффх.

Соответствие процентов и частоты выплаты

Предыдущие примеры описывают расчеты, в которых принимает участие частота выплат номинальных процентов, которые представляются ежемесячными платежами. Так, например, можно получить номинальную ставку на основе ежемесячных платежей. Как обычно, реальный операции не всегда настолько идеальны, как того хотелось бы.

ПРИМЕР 32

Банк указывает номинальную ставку 6,3% в месяц, но позволяет проводить еженедельные платежи, эквивалентные процентной ставке. Если я занимаю \$300 000 на 10 лет, какие должны быть еженедельные платежи?

Простой путь решить подобную задачу — использовать обычную функцию преобразования Номх_Еффу. Эта формула возвращает \$777,51:

=ПЛТ (Номх_Еффу (6,3%;12,52);10*52;300000;0;0)

ПРИМЕР 33

Мы открыли многолетний счет, с которого нужно ежемесячно снимать \$12 500. Какой эквивалент ежегодной суммы при использовании депозитной ставки с номинальными процентами 7% в месяц за год? Ежемесячная выплата проводится наперед, а эквивалентная сумма подсчитывается в конце каждого года.

Сначала вычислите ежемесячную эффективную ставку (используя пользовательскую функцию VBA). Следующая формула возвращает 0,58333%:

=Номх_Еффх (7%;12)

Потом вычислите эквивалентную ежегодную сумму, используя функцию ЕС. Эта формула возвращает -\$154 907,29:

=-ЕС (0,58333%;12;-12500;0;0)



В этом примере знаки могут привести вас в замешательство. Естественно, надо обращаться с расходами, как имеющими отрицательное значение, а поступлениями, как имеющими положительное значение. Однако мы будем использовать результат в качестве расходов, так что знак его изменяется на противоположный. Этого можно достичь, используя снимаемую со счета ежемесячную сумму — \$12 500 в качестве расходов или в результате изменения знака в функции БС.

Ограничения финансовых функций Excel

Основные финансовые функции (ПС, БС, ПЛТ, СТАВКА, КПЕР, ОБЩДОХОД и ОБЩПЛАТ) Excel очень полезны, но у них имеется два общих ограничения:

- ◆ Они обрабатывают только один уровень процентной ставки.
- ◆ Они обрабатывают только один уровень платежа.

Например, функция КПЕР не может использоваться при изменении суммы платежа, что характерно при использовании кредитной карточки. В таких вычислениях ежемесячная выплата основывается на уменьшении остаточного баланса, и может также подчиняться правилу минимальной суммы.

Общее решение задачи изменяемых платежей — создать план платежей наличными и использовать другие финансовые функции, которые могут обрабатывать различные ставки и периодические выплаты. Примеры подобных расчетов приведены в следующих двух главах. Если вкратце, то для этого применяются следующие функции:

- ◆ БЗРАСПИС. Возвращает будущую стоимость первоначальной основной суммы после применения ряда (набора) ставок сложных процентов.
- ◆ ВСД. Возвращает внутреннюю ставку доходности для ряда потоков денежных средств, представленных численными значениями.
- ◆ ЧПС. Возвращает величину чистой приведенной стоимости инвестиции, используя ставку дисконтирования, а также стоимости будущих выплат (отрицательные значения) и поступлений (положительные значения).
- ◆ МВСД. Возвращает модифицированную внутреннюю ставку доходности для ряда периодических денежных потоков. МВСД учитывает как затраты на привлечение инвестиции, так и процент, получаемый от реинвестирования денежных средств.
- ◆ ЧИСТВНДОХ. Возвращает внутреннюю ставку доходности для графика денежных потоков, которые не обязательно носят периодический характер.
- ◆ ЧИСТНЗ. Возвращает чистую приведенную стоимость для денежных потоков, которые не обязательно являются периодическими.

Отсроченное начало ряда регулярных выплат

В некоторых случаях ряд выплат может иметь отсроченное начало. Можно вычислить ПС регулярного ряда потоков с отсроченным началом с помощью формулы, подобной приведенной ниже:

$$=ПС(ставка; кпер; плт; бс; тип) * (1+ставка)^{-ОТС_ПЕР}$$

Здесь ОТС_ПЕР представляет число периодов, для которых отсрочивается первый платеж наличными.

ПРИМЕР 34

Я хочу одолжить деньги на основе отсроченной выплаты. Период отсрочки — один год. С этого времени заем выплачивается на протяжении 10 лет с ежемесячными платежами. Эффективная процентная ставка 8% в год. Заем обеспечивается недвижимостью, которой я владею, и банк готов предоставить аренду с выплатами, не превышающим 75% оцененной прибыли \$9 500 за месяц. Сколько мне можно занять?

Следующая формула использует обычную функцию AnnEff_Efx и возвращает \$550 422,02:
$$= \text{ПС}(\text{AnnEff_Efx}(8\%; 12; 10 * 12; -9500 * 75\%; 0; 0)) * (1 + \text{AnnEff_Efx}(8\%; 12))^{-12}$$

Оценка серии регулярных платежей

Мы можем расширить основной принцип дисконтирования последовательных, но различных платежей в результате формирования цепочки функций ПС. Например, если ПС1, ПС2 и ПС3 представляют различные текущие значения ряда платежей на протяжении периодов времени КПЕР1, КПЕР2 и КПЕР3, приведенная стоимость всего ряда выплат может быть найдена с помощью функции:

$$\text{ПС1} + \text{ПС2} (1+1)^{-\text{КПЕР1}} + \text{ПС2} (1+1)^{-\text{КПЕР1} + \text{КПЕР2}}$$

ПРИМЕР 35

Какая текущая стоимость имущества, приносящего прибыль \$5 000 в месяц на протяжении четырех лет, \$6 500 в месяц на протяжении последующих трех лет и \$8 500 в месяц на протяжении последних трех лет? Через 10 лет имущество будет оцениваться в \$1 300 000. Ежегодная ставка 10%, а все выплаты производятся авансом.

Следующая формула возвращает -\$978 224,54:

$$= \text{ПС}(\text{AnnEff_Efx}(10\%; 12); 48; 5000; 0, 1) + \text{ПС}(\text{AnnEff_Efx}(10\%; 12); 36; 6500; 0; 1) * (1 + \text{AnnEff_Efx}(10\%; 12))^{-48} \\ + 48 + \text{ПС}(\text{AnnEff_Efx}(10\%; 12); 36; 8500; 1300000; 1) * (1 + \text{AnnEff_Efx}(10\%; 12))^{-48 + 36}$$

Обратите внимание на то, что будущее значение \$1 300 000 включено в последнюю функцию ПС. Тот же ответ можно получить в результате “встраивания” следующего текущего значения в предыдущую функцию, как будущего значения. Но, помня о том, что ПС при этом представляет будущую прибыль, знак необходимо изменить на противоположный. Следующая формула возвращает \$978 224,54:

$$= \text{ПС}(\text{AnnEff_Efx}(10\%; 12); 48; 5000; -\text{ПС}(\text{AnnEff_Efx}(10\%; 12); 36; 6500, - \\ \text{ПС}(\text{AnnEff_Efx}(10\%; 12); 36; 8500; 1300000; 1); 1); 1)$$

Из этих двух вариантов первая формула (использование суммы формул дисконтирования) выглядит проще последнего метода; она выглядит более приемлемой, хотя и громоздкой для построения. Для упрощения она разбивается на три части (три ячейки), которые впоследствии объединяются вместе.

Следующая формула возвращает \$200 344,00:

$$= \text{ПС}(\text{AnnEff_Efx}(10\%; 12); 48; 5000; 0; 1)$$

Следующая формула возвращает \$139 559,07:

$$= \text{ПС}(\text{AnnEff_Efx}(10\%; 12); 36; 6500; 0; 1) * (1 + \text{AnnEff_Efx}(10\%; 12))^{-48}$$

Следующая формула возвращает \$638 331,47:

$$= \text{ПС}(\text{AnnEff_Efx}(10\%; 12); 36; 8500; 1300000; 1) * (1 + \text{AnnEff_Efx}(10\%; 12))^{-48 + 36}$$

Сумма всех трех элементов равняется \$978 224,54.

Резюме

В этой главе описаны финансовые функции и основные понятия временных денежных потоков и эквивалентных процентных ставок. В главе представлен ряд примеров, в которых использовались основные финансовые функции для накопления, дисконтирования и амортизации займа.

В следующей главе представлены примеры, которые обеспечивают в Excel расчет обесценивания, и описаны методы определения чистых приведенных прибылей и внутренней нормы прибыли.

Финансовые функции дисконтирования и обесценивания

В этой главе...

- ◆ Использование функции ЧПС
- ◆ Использование функции ВСД
- ◆ Несколько ставок в функциях ВСД и МВСД
- ◆ Использование функции БЗРАСПИС
- ◆ Резюме

Функции ЧПС (чистая приведенная стоимость) и ВСД (внутренняя ставка доходности), возможно, наиболее часто используемые инструменты финансового анализа. В этой главе представлено много примеров использования этих функций для различных типов финансовых расчетов.

Использование функции ЧПС

Функция ЧПС возвращает сумму любого ряда регулярных платежей, рассчитанных на настоящий момент с помощью одной учетной ставки. Синтаксис функции ЧПС приведен ниже (обязательные параметры выделены полужирным шрифтом):

ЧПС (**ставка**; **значение1**; значение2; . . .)

Притоки капитала представлены в виде положительных значений, а оттоки платежей — в виде отрицательных. Функция ЧПС имеет те же ограничения, что и рассмотренные в предыдущей главе финансовые функции типа ПС, ПЛТ, БС, Кпер и СТАВКА. Единственное отличие заключается в том, что суммы платежей могут изменяться.

Если отрицательные потоки превышают положительные, то функция возвратит отрицательную сумму. Точно так же, если положительные потоки превышают отрицательные потоки, функция ЧПС возвратит положительную сумму.

Если значение, возвращаемое функцией ЧПС, положительное, то это указывает на то, что в нулевом периоде инвестор мог бы выплатить полученную сумму и все-таки достичь необходимой учетной ставки. Если ЧПС возвращает отрицательное значение, то инвестор не получает необходимой учетной ставки. Эта ставка часто называется *границкой ставкой*. Последствия отрицательного ЧПС те, что инвестор выплачивает слишком много. “Правильная цена” требует дефицита в начальном денежном потоке.

Используемая учетная ставка — это единая эффективная ставка на протяжении периода учета денежных потоков. Таким образом, если денежные потоки регистрируются ежемесячно, то надо использовать ежемесячную эффективную ставку.

Определение ЧПС

Функция ЧПС предполагает, что первый платеж проводится в конце первого периода. Обратите внимание, что этот случай отличается от определения, используемого большинством финансовых калькуляторов, и оно не совпадает с определением, используемым учреждениями типа Института оценок США. Например, последнее определение устанавливает ЧПС, как разницу между текущим значением положительных и отрицательных денежных потоков.

Если использовать функцию ЧПС без дополнительной корректировки, то результат не будет соответствовать этому определению.

Таким образом, используя функцию ЧПС, нужно принимать во внимание начальный момент времени, “нулевой”, и начальный денежный поток. По этой причине, процедура адаптивования вычислений с помощью ЧПС в Excel выполняется следующим образом:

- ◆ Расценивайте число периодов скорее за количество точек, чем диапазонов времени между точками.
- ◆ Всегда включайте начальный денежный взнос в расчет, даже если выплаты не производятся до конца 1 периода (точка 1).
- ◆ Используйте формулу, подобную показанной ниже, для включения начального денежного потока:

$\text{=ЧПС(ставка; диапазон) * (1+ставка)}$

При использовании этой процедуры вычисления соответствуют принятым определениям ЧПС, а результаты совпадают с полученными на надежном финансовом калькуляторе. Между прочим здесь нет ошибки Microsoft. Справочная система четко указывает, что первый платеж в диапазоне предположительно приходится на конец первого периода. Если использовать предыдущую формулу и начальный момент времени (точка 0) \$0, то вы всегда получите правильный ответ.

Примеры функции ЧПС

В этом разделе содержится ряд примеров, которые демонстрируют функцию ЧПС.

ПРИМЕР 1

На рис. 12.1 показан рабочий лист с вычислением чистой приведенной стоимости для ряда денежных потоков в диапазоне B6 : B13.

Вычисление ЧПС в ячейке B15 требует использования следующей формулы. Эта формула возвращает $-\$33\,629,14$:

$\text{=ЧПС(B3; B6 : B13) * (1+B3)}$

Рабочий лист на рис. 12.1 также демонстрирует метод перекрестного контроля результатов вычислений ЧПС. Столбец E содержит копию исходных денежных потоков за одним исключением. В точке 0 денежный поток теперь равен исходному потоку в точке 0 (B6) минус подсчитанное ЧПС. В этом примере значение равно $-\$166\,370,86$. Формула перекрестной проверки в ячейке E15, показанной здесь, возвращает \$0,00:

$\text{=ЧПС(B3; E6 : E13) * (1+B3)}$

Как работает перекрестная проверка? Учетная ставка 10% используется для вычисления излишка или недостатка, который возникает от желаемого 0% уровня. В данном случае излишек равен \$33 629,14. Другой излишек выражается в сроках текущей стоимости (точка 0). Если излишек вычитается из потока точки 0, то в результате повторного расчета не должно возникать излишка. Другими словами, если текущее значение ЧПС вычитается из потока в точке 0, то ЧПС при той же ставке должно равняться 0. Если оно равно 0, это означает, что был достигнут необходимый процент.

Время	Денежные потоки	Время	Денежные потоки
0	(\$200 000,00)	0	(\$166 370,86)
1	\$40 000,00	1	\$40 000,00
2	\$30 000,00	2	\$30 000,00
3	\$20 000,00	3	\$20 000,00
4	\$50 000,00	4	\$50 000,00
5	\$20 000,00	5	\$20 000,00
6	\$50 000,00	6	\$50 000,00
7	\$30 000,00	7	\$30 000,00

Рис. 12.1. Этот рабочий лист использует функцию ЧПС

ПРИМЕР 2

В примере, показанном на рис. 12.2, вычисляется чистая приведенная стоимость денежных потоков, которые начинаются в конце первого периода.

Время	Денежные потоки	Коэффициент	Текущее значение
0	\$0,00	1	\$0,00
1	\$40 000,00	0,909090909	\$36 363,64
2	\$30 000,00	0,826446281	\$24 793,39
3	\$20 000,00	0,751314801	\$15 026,30
4	\$50 000,00	0,683013455	\$34 150,67
5	\$20 000,00	0,620921323	\$12 418,43
6	\$50 000,00	0,56447393	\$28 223,70
7	\$30 000,00	0,513158118	\$15 394,74

Рис. 12.2. На этом рабочем листе вычисляется ЧПС для денежных потоков, которые начинаются в конце первого периода

Вычисление ЧПС в ячейке B16 требует использования следующей формулы:

$$= \text{ЧПС}(B3; B7: B14) * (1 + B3)$$

Вычисления указывают на то, что мы можем позволить себе кредит в \$166 370,86 при выплате учетной ставки 10%.

В этом примере используется другой метод перекрестной проверки результата (столбцы C и D). Столбец C содержит формулы вычисления коэффициента текущего значения для каждого платежа. Формула в ячейке C7 следующая:

$$= (1 + \$B\$3) ^ -A7$$

Текущие значения вычисляются в столбце D в результате умножения каждого платежа на коэффициент соответствующего текущего значения. Формула в ячейке D7 следующая:

$$= C7 * B7$$

Столбец D содержит все вычисляемые текущие значения, и сумма значений этого столбца представляет сумму текущих значений. По определению, сумма текущих значений (ячейка D16) должна равняться результату, возвращаемому ЧПС.

ПРИМЕР 3

В этом примере (рис. 12.3) вычисляется чистая приведенная стоимость денежных потоков с начальным (точка 0) положительным значением.

Время	Денежные потоки
0	\$40 000,00
1	\$40 000,00
2	\$30 000,00
3	\$30 000,00
4	\$20 000,00
5	\$20 000,00
6	\$15 000,00
7	\$15 000,00

ЧПС = \$165 939,65

Рис. 12.3. На этом рабочем листе вычисляется чистая приведенная стоимость для начального потока наличных

Вычисление чистой приведенной стоимости проводится в ячейке B15, которая содержит следующую формулу:

$$= \text{ЧПС} (B3 ; B6 : B13) * (1 + B3)$$

Вычисление указывает на то, что мы можем инвестировать \$165 939,65 и получить денежные потоки с учетной ставкой 10%. В этом случае, однако, мы инвестируем \$165 939,65 и имеем право на получение в начальный момент времени поток размером \$40 000.

Этот пример может показаться необычным, но подобные ситуации распространены при покупке недвижимого имущества, в которых рента выплачивается авансом. На практике, завершение платежей редко совпадает с датой выплаты ренты. В этом случае баланс аренды, оплаченной заранее за период после даты завершения выплат, включается в расчет, описанный выше.

Если мы не знаем начальное значение, то помещаем в столбец для точки 0 общий капитал, и ЧПС возвращает необходимое значение, используя нужный учетный процент. Если мы знаем начальное значение, то можем поместить его со знаком “минус” в точке 0, а ЧПС возвратит приближительную сумму всех выплат, необходимую для получения требуемого учетного процента.

ПРИМЕР 4

В этом примере (рис. 12.4) вычисляется чистая приведенная стоимость с окончательным значением и авансовыми денежными потоками.

Этот пример описывает типичную арендную плату за недвижимое имущество с выплатами, предоставляемыми ежегодно авансом, с допустимой продажей после семи лет за \$450 000. Обратите внимание на начальную и конечную выплаты. В этом случае, инвестор собирается получить первую арендную плату размером \$30 000 немедленно, а также \$40 000 в конце. Что может не соответствовать действительности, и, если последняя оплата не взимается, то надо сделать ее равной \$0.

Время	Входящий поток	Окончательный платеж	Выплачиваемая сумма
0	\$30 000,00		\$30 000,00
1	\$30 000,00		\$30 000,00
2	\$30 000,00		\$30 000,00
3	\$35 000,00		\$35 000,00
4	\$35 000,00		\$35 000,00
5	\$35 000,00		\$35 000,00
6	\$40 000,00		\$40 000,00
7	\$40 000,00	\$450 000,00	\$490 000,00
ЧПС =			\$428 026,29

Рис. 12.4. Эта рабочая книга демонстрирует выплаты с конечным значением

Вычисление ЧПС в ячейке D15 следующее:

$$= \text{ЧПС} (B3 ; D6 : D13) * (1 + B3)$$

ПРИМЕР 5

Этот пример, показанный на рис. 12.5, похож на предыдущий, но в нем продемонстрирован случай (в ячейке B14), когда конечное значение суммируется с последним платежом.

Формула в ячейке B16 следующая:

$$= \text{ЧПС} (B3 ; B7 : B14) * (1 + B3)$$

Примеры 4 и 5 отличаются только способом представления дат выплаты. Если вы хотите отделить капитал от дохода, предпочтителен вариант, используемый в примере 4. Отделение дохода и капитала (как в примере 4) делает достаточно легким понимание структуры выплат без изучения используемой формулы.

Время	Входящие потоки
0	\$30 000,00
1	\$30 000,00
2	\$30 000,00
3	\$35 000,00
4	\$35 000,00
5	\$35 000,00
6	\$40 000,00
7	\$490 000,00
ЧПС = \$428 026,29	

Рис. 12.5. Эта рабочая книга демонстрирует выплаты с конечным значениям

Время	Входящие потоки	Основные платежи	Денежные потоки
0	\$30 000,00	(\$280 000,00)	(\$250 000,00)
1	\$30 000,00		\$30 000,00
2	\$30 000,00		\$30 000,00
3	\$35 000,00		\$35 000,00
4	\$35 000,00		\$35 000,00
5	\$35 000,00		\$35 000,00
6	\$40 000,00		\$40 000,00
7	\$40 000,00	\$450 000,00	\$490 000,00
ЧПС = \$148 026,29			

Рис. 12.6. Этот рабочий лист демонстрирует денежные потоки с предельными значениями

ПРИМЕР 6

Этот пример является упрощенной моделью, в которой используется начальное и окончательное значения (рис. 12.6). Он представляет типичный пример инвестиции, в котором цель — определить насколько запрашиваемая цена превышает предполагаемую прибыль.

Следующая формула указывает, что, при запрашиваемой цене \$280 000, превышение предполагаемой прибыли равняется \$148 026,29:

$$= \text{ЧПС} (B3 : D8 : D15) * (1 + B3)$$

С другой стороны, инвестор мог выплачивать всего \$428 026,29 и все-таки достичь учетной ставки 10%.

ПРИМЕР 7

В предыдущих примерах процент рассчитывался соответственно периодам времени, через которые проводятся выплаты. Очень часто вы будете сталкиваться с ситуациями несоответствия единиц измерения учетной ставки и периодов платежей. Наиболее часто учетная ставка рассматривается как ежегодный эффективный процент, а денежные потоки поступают ежемесячно или ежеквартально.

Наиболее простое решение — использовать функцию (AnnEff_Effx, которая рассматривалась в предыдущей главе). Это пользовательская функция VBA, которая легко преобразует годовую процентную ставку в ежемесячную эффективную основу, используемую при расчете ежемесячного поступления платежей.

На рис. 12.7 видно, что арендная плата \$12 000 выплачивается ежеквартально авансом. Также видно, что начальная цена составляет \$700 000 и продажа (через три года) осуществляется за \$900 000. Обратите внимание на то, что рента выплачивалась авансом, потому покупатель выплачивает исправленную начальную цену. Однако, по истечении трех лет (12 кварталов), применяется то же правило, и арендная плата, выплачиваемая в следующем квартале, будет получена новым владельцем. Если годовая эффективная ставка равна 7%, то ЧПС возвращает \$166 099,72.

Example 01-09 (NPV).xls				
	A	B	C	D
1	Квартальные платежи			
2				
3	Тип ставки		Effective	
4	Количество платежей		1	
5	Годовая учетная ставка		7%	
6	Пересчитанное количество платежей		4	
7	Ставка за новый период		1,70585%	Квартально
8				
9	Период	Поточные платежи	Освоение платежей	Денежные потоки
10	0	\$12 000,00	(\$700 000,00)	(\$688 000,00)
11	1	\$12 000,00		\$12 000,00
12	2	\$12 000,00		\$12 000,00
13	3	\$12 000,00		\$12 000,00
14	4	\$12 000,00		\$12 000,00
15	5	\$12 000,00		\$12 000,00
16	6	\$12 000,00		\$12 000,00
17	7	\$12 000,00		\$12 000,00
18	8	\$12 000,00		\$12 000,00
19	9	\$12 000,00		\$12 000,00
20	10	\$12 000,00		\$12 000,00
21	11	\$12 000,00		\$12 000,00
22	12		\$900 000,00	\$900 000,00
23			ЧПС =	\$166 099,72
24				

Рис. 12.7. Вычисление ЧПС с использованием ежеквартальных денежных потоков

Часто арендные платежи начисляются ежегодно. Сейчас это воспринимается в диковинку. Однако до появления калькуляторов и компьютеров, это был подход, общепринятый оцен-

щиками, которые использовали таблицы предварительных расчетов и специальные константы перерасчетов. На рис. 12.8 показаны те же данные, но на этот раз мы предположили, что рента \$48 000 в год выплачивается ежегодно с задержкой. Если годовая эффективная ставка равна 7%, то ЧПС возвращает \$160 635,26.

Пример 01-09 (NPV).xls				
	A	B	C	D
26	Расчет ежегодных платежей			
27				
28	Период	Поточные платежи	Основные платежи	Денежные потоки
29	0		(\$700 000,00)	(\$700 000,00)
30	1	\$48 000,00		\$48 000,00
31	2	\$48 000,00		\$48 000,00
32	3	\$48 000,00	\$900 000,00	\$948 000,00
33				
34			ЧПС =	\$160 635,26
35				
36	Ошибка вычислений			\$5 464,46
37				3,29%
38				
39				
40				

Рис. 12.8. Вычисление ЧПС в результате преобразования ежеквартальных потоков в ежегодные

Использование функции ЧПС для вычисления накопленных сумм

В этом разделе представлено два примера использования функции ЧПС для вычисления будущего значения или накопленного капитала. Эти примеры основаны на том факте, что:

$$БС = ПС * (1 + \text{ставка})$$

ПРИМЕР 8

Данные для этого примера показаны на рис. 12.9. Вычисление чистой приведенной стоимости выполняется по следующей формуле в ячейке В15:

$$= \text{ЧПС}(В3; В6:В13) * (1 + В3)$$

Будущее значение вычисляется по такой формуле (в ячейке В17):

$$= \text{ЧПС}(В3; В6:В13) * (1 + В3) * (1 + В3)^7$$

Время	Денежные потоки	Процент	Баланс
0	\$100 000,00		\$100 000,00
1	\$40 000,00	\$10 000,00	\$150 000,00
2	\$30 000,00	\$15 000,00	\$195 000,00
3	\$20 000,00	\$19 500,00	\$234 500,00
4	\$50 000,00	\$23 450,00	\$307 950,00
5	\$20 000,00	\$30 795,00	\$358 745,00
6	\$50 000,00	\$35 874,50	\$444 619,50
7	\$30 000,00	\$44 461,95	\$519 081,45
ЧПС =		\$266 370,86	
Будущее значение		\$519 081,45	Расчитано по суммарному балансу

Рис. 12.9. Вычисление БС с использованием функции ЧПС

Результат проверяется в столбце D, в котором вычисляется текущий баланс после выплат процентов. Результат вычисления будущего значения соответствует совокупному проценту. Процент вычисляется по учетной ставке, которая умножается на баланс предыдущего месяца. Текущий баланс — это сумма предыдущего баланса, процента и текущих месячных платежей.

Очень важно должным образом направление денежных потоков. Если текущий баланс для предыдущего месяца отрицательный, процент тоже будет отрицательный. Изучение направления потоков крайне важно в приведенных выше формулах.

ПРИМЕР 9

В главе 11 описано использование функции ПЛТ для вычисления эквивалента платежей для данного текущего значения. Также можно использовать функцию ЧПС, вложенную в функцию ПЛТ, для вычисления эквивалента одноразового платежа на основе ряда уже проведенных платежей.

Это типичная задача, в которой усредненный по времени один платеж заменяет ряд изменяемых платежей. Пример — соглашение, в котором план различных платежей арендной платы заменяется постоянными платежами. В примере, показанном на рис. 12.10, следующая формула возвращает (в ячейке C27) \$10 923,24, что является значением платежа, которое неизменно и эффективно заменяет разные платежи в столбце B:

=ПЛТ (C7; C6; -B25; 0; C8)

Пример этого раздела предоставляет гибкость пользователю в выборе ставки и частоты выплат. Проверка данных используется, чтобы позволить выбор: использовать эффективную или нормальную ставку в ячейке C3. Этот тип вычислений часто используется для получения альтернативных фиксированных и переменных арендных выплат.

Использование функции ВСД

В Excel функция ВСД возвращает учетную ставку, которая принимается для инвестиций, состоящих из платежей, проведенных за одинаковые периоды времени. Другими словами, функция ВСД — это особый вид ЧПС, и мы будем использовать ее при разработке системы автоматической перекрестной проверки. Синтаксис функции ВСД следующий:

ВСД (значения; предположения)

Example 01-09 [NPV].xls				
A	B	C	D	E
Calculating Equivalent Payments with NPV				
1	Тип ставки	Effective		
2	Количество платежей	12		
3	Годовая учетная ставка	7%		
4	Пересчитанное количество платежей	12		
5	Ставка за новый период	0,56541%	Ежемесячно	
6	Тип платежей:	1		
7				
8				
9				
10	Время	Денежные потоки		
11	0	\$10 000,00		
12	1	\$10 000,00		
13	2	\$10 000,00		
14	3	\$10 500,00		
15	4	\$10 500,00		
16	5	\$10 500,00		
17	6	\$11 250,00		
18	7	\$11 250,00		
19	8	\$11 250,00		
20	9	\$12 000,00		
21	10	\$12 000,00		
22	11	\$12 000,00		
23	12			
24	ЧПС=	\$127 100,53		
25				
26				
27	Эквивалентный одноразовый платеж	\$10 923,24		
28	Проверка:	\$127 100,53		
29				

Рис. 12.10. Вычисление эквивалентных платежей с помощью ЧПС



Аргумент значения содержит несколько значений. Пустые ячейки рассматриваются как содержащие значение нуль. Если диапазон значений содержит нули или текст, функция ВСД не возвращает ошибку. Скорее, она возвращает неправильный результат. Так, если диапазон B1:B40 содержит текст в ячейках B11:B20, ВСД будет брать за основу вычислений 30 последовательных выплат. Подобное представление данных плохо анализируется Excel, если текст содержит символы пробела, дефиса или (хуже всего) нуля.

В большинстве случаев, ВСД может быть подсчитан только итерацией. Аргумент предположения, если таковой используется, служит “основой” для итерационного вычисления. Доказано, что прогноз -0,9 всегда приводит к конечному ответу. Другие прогнозы, например 0, обычно (но не всегда) приводят к ответу.

Существенное требование к функции ВСД заключается в том, что в задаче должны использоваться как отрицательные, так и положительные денежные потоки. Для получения прибыли должны существовать поступления и платежи. Нет необходимости в списке сначала указывать поступления. При анализе займа с помощью ВСД, например, сумма займа будет отрицательной (и получаемой первой), а последующие выплаты — положительными.

ВСД — очень могучий инструмент, и его применение больше, чем простое вычисления дохода инвестиции. Эта функция может использоваться в любой ситуации, в которой надо вычислить средний доход.

ПРИМЕР 10

В этом примере задается основная матрица вычислений ВСД (рис. 12.11). В нем демонстрируется постоянная проблема частоты поступления платежей и определения ВСД для текущей частоты. Так, если платежи производятся ежемесячно, функция возвратит ежемесячную ВСД. В примере проводится проверка данных, что позволяет пользователю указать тип выплат (1, 2, 4, 12, 13, 26, 52, 365, 366). Сделанный выбор определяет способ преобразования процентной ставки, а также влияет на текст в ячейках строки 5, которые содержат формулы, использующие текст в ячейке D3.

Месяц	Входящие платежи	Основные платежи	Ежемесячные выплаты
0	\$0,00	(\$2 000 000,00)	(\$2 000 000,00)
1	\$50 000,00		\$50 000,00
2	\$50 000,00		\$50 000,00
3	\$50 000,00		\$50 000,00
4	\$50 000,00		\$50 000,00
5	\$50 000,00		\$50 000,00
6	\$50 000,00		\$50 000,00
7	\$50 000,00		\$50 000,00
8	\$50 000,00		\$50 000,00
9	\$50 000,00		\$50 000,00
10	\$50 000,00		\$50 000,00
11	\$50 000,00		\$50 000,00
12	\$50 000,00	\$2 500 000,00	\$2 550 000,00
Месечная ставка		4,14958%, Ежемесячно	
Годовая ставка		62,88844% За год	
Проверка ЧПС		(\$0,00)	

Рис. 12.11. Этот рабочий лист позволяет пользователю выбрать период времени поступления денежных потоков

Следующая формула в ячейке D22 проверяет достоверность ПОЛУЧЕННЫХ данных:
 $=\text{ЧПС}(D20;D6:D18) * (1+D20)$

ВСД — это учетная ставка, при которой денежные выплаты приводят к ЧПС, равному 0. Формула в ячейке D22 использует ВСД в функции ЧПС применимо к тем же выплатам. Подстановка ВСД в ЧПС (квартально) приводит к результату \$0,00 — так проверяется результат основных вычислений.

ПРИМЕР 11

Вам может понадобиться вычислить средний темп прироста или среднюю норму прибыли. Из-за сложных процентов простое арифметическое усреднение не выдаст правильный ответ. Хуже того, если платежи выполняются в обоих направлениях, то арифметическое усреднение не будет учитывать знаки.

Решение состоит в использовании функции ВСД для вычисления средней геометрической нормы прибыли. Это простое вычисление, которое заключается в определении отдельного процента за период, который аналогичен нескольким другим.

В примере 11 (рис. 12.12) показана функция ВСД, используемая в вычислении средней геометрической нормы прибыли, основанной на коэффициентах столбца В. Вычисления темпа прироста на протяжении каждого года приведены в столбце С. Например, формула в ячейке С5 следующая:

$$=(B5-B4)/B4$$

В остальных столбцах показан средний геометрический темп прироста за различные периоды. Формулы в строке 10 применяют функцию ВСД для вычисления внутренней нормы прибыли. Например, формула в ячейке F10 возвращает 5,241%:

$$=\text{ВСД}(F4:F8; -0,9)$$

Другими словами, темпы прироста 5,21%, 4,86% и 5,66% для трех отдельных периодов эквивалентны среднему геометрическому темпу прироста 5,241% за один общий период.

В вычислении ВСД учитываются направление платежей.

Год	Коэффициент	Ежегодный прирост	1996-1997	1996-1998	1996-1999	1996-2000
1996	100,00		-100,00	-100,00	-100,00	-100,00
1997	105,21	5,21%	105,21	0	0	0
1998	110,32	4,86%		110,32	0	0
1999	116,56	5,66%			116,56	0
2000	119,94	2,90%				119,94
Средней прирост с 1996 года			5,210%	5,033%	5,241%	4,650%

Рис. 12.12. Использование функции ВСД для вычисления среднего геометрического темпа прироста

ПРИМЕР 12

На рис. 12.13 показан рабочий лист, в котором проводится проверка текущего значения ВСД. Эта проверка основывается на определении ВСД: сумма положительных и отрицательных денежных потоков равна 0.

Период	Потоки	Проверка ВСД	Проверка ЧПС
0	(\$100 000,00)	(\$100 000,00)	(\$100 000,00)
1	\$14 000,00	\$13 570,24	\$12 727,27
2	\$14 000,00	\$13 153,68	\$11 570,25
3	\$14 000,00	\$12 749,90	\$10 518,41
4	\$14 000,00	\$12 358,52	\$9 562,19
5	\$14 000,00	\$11 979,15	\$8 692,90
6	\$15 000,00	\$12 440,82	\$8 467,11
7	\$15 000,00	\$12 058,92	\$7 697,37
8	\$15 000,00	\$11 688,75	\$6 997,61
ЧПС	(\$23 766,89)		
ВСД	3,167%	0	(\$23 766,89)

Рис. 12.13. Проверка ВСД и ЧПС в результате суммирования текущих значений

Чистая приведенная стоимость вычисляется в ячейке В16:

$$= \text{ЧПС} (D3; B6 : B14) * (1 + D3)$$

Внутренняя ставка доходности вычисляется в ячейке В17:

$$= \text{ВСД} (B6 : B14; -0,9)$$

В столбце С с помощью формул вычисляется текущее значение. Они используют результат определения ВСД (в ячейке В17) в качестве учетного процента, а количество периодов (в столбце А) в качестве показателя степени. Например, формула в ячейке С6 следующая:

$$= B6 * (1 + \$B\$17) ^{-A6}$$

Сумма значений в столбце С равна 0.

Формулы в столбце D используют учетную ставку (в ячейке D3) для вычисления текущего значения. Например, формула в ячейке D6 следующая:

$$= B6 * (1 + \$D\$3) ^{-A6}$$

Сумма значений в столбце D равна чистой приведенной стоимости.

Для серьезных задач функций ЧПС и ВСД — это взаимозаменяемые средства перекрестного контроля.

Несколько ставок в функциях ВСД и МВСД

В стандартных денежных потоках знак платежей изменяется только один раз: от минуса к плюсу или от плюса к минусу. Однако существуют финансовые ситуации, в которых знак может измениться больше, чем единожды. В этих случаях возможно существование больше одной ВСД.

ПРИМЕР 13

На рис. 12.14 показан пример с двумя ставками ВСД, каждая из которых использует различное “начальное значение” в качестве аргумента предположения. Как вы видите, формула возвращает различные результаты.

Несколько процентных ставок						
Период	Потоки	Ставка 13,88%	Баланс	Ставка 7,04%	Баланс	
0	(\$14 375,00)	\$0,00	(\$14 375,00)	\$0,00	(\$14 375,00)	
1	\$6 250,00	(\$1 995,53)	(\$10 120,53)	(\$1 012,58)	(\$9 137,58)	
2	\$6 250,00	(\$1 404,93)	(\$5 275,46)	(\$643,65)	(\$3 531,23)	
3	\$6 250,00	(\$732,34)	\$242,20	(\$248,74)	\$2 470,03	
4	\$6 250,00	\$33,62	\$6 525,82	\$173,99	\$8 894,02	
5	\$0,00	\$905,91	\$7 431,73	\$626,50	\$9 520,52	
6	\$0,00	\$1 031,67	\$8 463,40	\$670,63	\$10 191,14	
7	\$0,00	\$1 174,89	\$9 638,29	\$717,87	\$10 909,01	
8	\$0,00	\$1 337,98	\$10 976,28	\$768,43	\$11 677,44	
9	(\$12 500,00)	\$1 523,72	(\$0,00)	\$822,56	(\$0,00)	
Всего		\$1 875,00	Всего	\$1 875,00		
ВСД (1)	13,88197%	(\$0,00)				
ВСД (2)	7,0440%	(\$0,00)				

Рис. 12.14. Рабочий лист, демонстрирующий многократные ВСД

Формула ВСД в ячейке B21 (которая возвращает результат 13,88%) следующая:
 $\text{=ВСД}(B7 : B16 ; B3)$

Формула ВСД в ячейке B22 (которая возвращает результат 7,04%) следующая:
 $\text{=ВСД}(B7 : B16 ; B4)$

Таким образом, какая ставка правильная? К сожалению, обе. На рис. 12.14 показан учетная ставка и балансовые вычисления для обоих значений ВСД. Оба результата показывают, что инвестор может выплачивать и получать любую процентную ставку и может обеспечить конечный баланс \$0. Интересно, что суммарный доход (\$1 875) одинаковый.

Но в подобных вычислениях существует один недостаток. Этот пример иллюстрирует “сценарий наихудшего случая” практической ошибки многих вычислений ВСД. Анализ ЧПС и ВСД принимает два предположения:

- ♦ Можно действительно получить прогнозируемый (для ЧПС) или вычисленный (для ВСД) процент на невыплаченном балансе.
- ♦ Процент не изменяется в зависимости от знака денежных поступлений.

Первое предположение может быть правильным и неправильным. Возможна ситуация, когда баланс изменяется внешним образом (но в рассматриваемых примерах изменяемой процентной ставки во времени это не имеет места). Но существует реальная проблема со вторым предположением. Банки просто никогда не назначают одинаковую ставку за заем и вклад.

ПРИМЕР 14

Функция МВСД пытается решить последнюю задачу нескольких ставок. В примере этого раздела демонстрируется эффективность использования функции МВСД.

На рис. 12.15 показан рабочий лист, в котором используются те же данные, что и в примере 13. Отдельные ставки предоставляются для займа (ячейка В3) и для вклада (ячейка В4). Они используются в качестве аргументов функции МВСД (ячейка В19), а результат равен 6,1279 %, который отличается для обоих вычислений ВСД:

=МВСД(В7:В16;В3;В4)

Несколько процентных ставок				
Период	Приходящие потоки	Исходящие потоки	Пересмотренные платежи	
0	(\$14 375,00)	\$0,00	(\$14 375,00)	(\$20 130,35)
1	\$6 250,00	\$6 250,00	\$0,00	\$0,00
2	\$6 250,00	\$6 250,00	\$0,00	\$0,00
3	\$6 250,00	\$6 250,00	\$0,00	\$0,00
4	\$6 250,00	\$6 250,00	\$0,00	\$0,00
5	\$0,00	\$0,00	\$0,00	\$0,00
6	\$0,00	\$0,00	\$0,00	\$0,00
7	\$0,00	\$0,00	\$0,00	\$0,00
8	\$0,00	\$0,00	\$0,00	\$0,00
9	(\$12 500,00)	\$0,00	(\$12 500,00)	\$34 380,83
ЧПС:	\$22 162,19	(\$20 130,35)		
МВСД	6,1279%	ВСД (пересмотренный)		6,1279%

Рис. 12.15. Модифицированная внутренняя ставка доходности

Функция МВСД разделяет отрицательные и положительные денежные потоки, вычисляя их по разным ставкам: ставка займа (для отрицательных потоков) и депозитная ставка (для положительных потоков).

Мы можем проанализировать алгоритм вычислений МВСД, видоизменив денежный поток, в котором сравнивается два значения ЧПС (см. рис. 12.15; столбцы С:Е). Отрицательное значение ЧПС размещается в точке 0, а положительное значение выражается, как эквивалентное будущее значение (в результате накопления на депозите) в конце срока инвестиции. ВСД измененного денежного потока та же, что и МВСД исходного денежного потока.

Этот пример показывает, что применяемая методика вычислений может быть разной. При разделении отрицательных и положительных потоков в МВСД используется подход, в котором процент рассчитывается на основе денежных потоков. Банки, конечно, рассчитывают процент, исходя из балансовой стоимости. Попытка решить подобную задачу продемонстрирована в следующем примере.

ПРИМЕР 15

Функция МВСД использует две ставки: одну для отрицательных, а другую для положительных платежей. Фактически, процентные ставки основываются на балансе, а не на денеж-

ных поступлений. В примере этого раздела применяются различные ставки в отрицательных и положительных балансах. Вычисление модифицированного процента базируется на использовании функции ЕСЛИ для определения используемой ставки.

Анализируя проект, в котором процент выплачивается и начисляется, несложно предположить, что суммарный баланс должен равняться 0. Если он больше 0, то вы действительно получили больше, чем установлено в депозитной ставке. Если же он меньше 0, то вы еще и должны деньги, а ставка займа недооценена. В этом примере принимается фиксированная ставка займа и вычисляется депозитная ставка, необходимая для обеспечения нулевого конечного баланса.

В нашем примере ставка займа устанавливается пользователем. Процент, получаемый на положительных балансах, изначально “подбирается” пользователем. Процент на отрицательных балансах — это и есть ставка займа. Процент на положительных балансах определяется по “подобранной” ставке. Если “подобранная” ставка точно равняется доходу, то конечный баланс равен 0. Команда Сервис⇒Поиск решения Excel используется для определения точной ставки в результате подбора процентной ставки на положительных балансах для получения конечного баланса равного 0. Описанный метод демонстрируется в примере на рис. 12.16.

Период	Процент	Баланс	Потоки
0	\$0,00	(\$14 375,00)	(\$14 375,00)
1	\$6 250,00	(\$1 293,75)	\$4 956,25
2	\$6 250,00	(\$847,69)	\$5 402,31
3	\$6 250,00	(\$361,48)	\$5 888,52
4	\$6 250,00	\$160,61	\$6 410,61
5	\$0,00	\$710,61	\$710,61
6	\$0,00	\$771,58	\$771,58
7	\$0,00	\$837,77	\$837,77
8	\$0,00	\$909,65	\$909,65
9	(\$12 500,00)	\$987,69	(\$11 512,31)
		ЧПС	0,0000%

Рис. 12.16. Накопление баланса для нескольких ВСД

Измененный поток, полученный в результате модификации текущего баланса, должен иметь значение ВСД, приближенное к нулю.

Но что все это значит? Это означает, что, если оплачивать 9% на отрицательных балансах, то в этом проекте предоставляется ставка 8,579% на положительных балансах. Название проекта переводится как “Эквивалент ставки риска ВСД” и свидетельствует о том, что он определяет, насколько эффективно возвращаются деньги, инвестированные в банк.

Не требуется, чтобы ставка займа была фиксированной. Банк мог бы построить вычисления таким же образом, но фиксировать депозитную ставку а с помощью средства Поиск решения вычислить эквивалентную ставку займа.

Использование функции БЗРАСПИС

Функция БЗРАСПИС вычисляет будущее значение начальной суммы после применения со временем ряда разных ставок. Ее синтаксис следующий:

БЗРАСПИС (первичное ; план)



Функция БЗРАСПИС доступна только при установке надстройки "Пакета анализа".

ПРИМЕР 16

В этом примере, показанном на рис. 12.17, используется функция БЗРАСПИС для вычисления накопленной суммы вместе с другими формулами, которые используют предоставленные данные для вычисления коэффициента и среднего геометрического темпа прироста.

Этот рабочий лист содержит коэффициент цен на акции между 1997 и 2001 годами; для 1997 года задан коэффициент 100. Этот пример может ответить на следующий запрос: *Если бы мы купили акции в 1997 на сумму \$1 000, какую стоимость они имели бы в 2001, и какой должен быть средний темп прироста?*

Стоимость акций, показанная в ячейке В13, будет составлять \$1 296,81. Это эквивалент использования ставки 6,714% для начальной инвестиции \$1 000.

E:\example 16 (FVSCHEDULE).xls					
A	B	C	D	E	F
БЗРАСПИС					
1					
2					
3	Начальное значение:	\$1 000			
4					
5		Прирост	Коэффициент	Годовой прирост	Накопленная сумма
6					
7	1997		100,00		\$1 000,00
8	1998	6,50%	106,50	6,50%	\$1 065,00
9	1999	7%	113,96	6,75%	\$1 139,55
10	2000	8,90%	124,10	7,46%	\$1 240,97
11	2001	4,50%	129,68	6,71%	\$1 296,81
12	Среднее	6,73%			
13	Накоплено	\$1 296,81			
14	Среднее геометрическое	6,714%			
15	Среднее геометрическое	6,714%	(метод одной формулы)		
16			Ошибка		
17	Проверка суммы	\$1 296,81	\$0,00		
18	Проверка среднего	\$1 296,81	\$0,00		
19					

Рис. 12.17. Использование функции БЗРАСПИС

Накопленная сумма (ячейка В13) вычисляется по следующей формуле:

=БЗРАСПИС (В3 ; В7 : В10)



Обратите внимание, что функция БЗРАСПИС не зависит от знака. Она возвращает будущее значение с тем же знаком, что и текущее значение. Также обратите внимание, что темпы прироста задаются в виде эффективных ставок за соответствующий период времени. В данном примере период времени вычисляется в годах, так что темпы прироста также выражены в ежегодных сроках.

Формула в ячейке В14 вычисляет средний геометрический темп прироста:

=СТАВКА (4 ; 0 ; -В3 ; В17 ; 0)

Обратите внимание, что в формуле используется знак "минус" в третьем аргументе (текущее значение). Можно также вычислить среднюю геометрическую ставку доходности, используя еще одну формулу (ячейка В15):

=СТАВКА (4 ; 0 ; -В3 ; БЗРАСПИС (В3 ; В7 : В10) ; 0)

В этом примере также демонстрируется удобный способ вычисления коэффициента цен, основанного на плане темпов прироста (столбец С). Эта тема детально раскрывается в следующей главе.

Вычисления обесценивания

В этом разделе рассматривается обесценивание, критический элемент во многих расчетах инвестирования. Excel предлагает пять функций вычисления обесценивания актива во времени. Обесценивание актива рассматривается на протяжении определенного периода времени и основано на исходном значении и сроке эксплуатации. Выбранная функция зависит от типа метода обесценивания.

В табл. 12.1 приведены функции обесценивания Excel и аргументы, используемые каждой из них. Для получения детальной информации об этих функциях обратитесь к справочной системе Excel.

Таблица 12.1. Функции обесценивания Excel

Функция	Метод обесценивания	Аргументы*
АПЛ	Возвращает величину амортизации актива за один период, рассчитанную линейным методом.	нач_стоимость; ост_стоимость; время_эксплуатации
ФУО	Возвращает величину амортизации актива за заданный период, рассчитанную методом фиксированного уменьшения остатка.	нач_стоимость; ост_стоимость; время_эксплуатации; период; [месяцы]
ДДОБ	Возвращает значение амортизации актива за данный период, используя метод двойного уменьшения остатка или иной явно указанный метод.	нач_стоимость; ост_стоимость; время_эксплуатации; период; [коэффициент]
АСЧ	Возвращает величину амортизации актива за данный период, рассчитанную методом "суммы (годовых) чисел".	нач_стоимость; ост_стоимость; время_эксплуатации; период
ПУО	Возвращает величину амортизации актива для любого выбранного периода, в том числе для частичных периодов, с использованием метода двойного уменьшения остатка или иного указанного метода.	нач_стоимость; ост_стоимость; время_эксплуатации; нач_период; кон_период; [коэффициент]; [без_переключения]

*Аргументы в скобках необязательны.

Аргументы для функций обесценивания описываются следующим образом:

- ◆ **нач_стоимость.** Первоначальная стоимость актива.
- ◆ **ост_стоимость.** Стоимость актива после полного обесценивания.
- ◆ **время_эксплуатации.** Ряд периодов, в течение которых фонд обесценится.
- ◆ **период.** Период времени, для которого производится вычисление.
- ◆ **месяцы.** Число месяцев в первом году; если опустить, Excel по умолчанию использует 12.
- ◆ **коэффициент.** Ставка снижения баланса; если опустить, предполагается равной 2 (то есть, двойное понижение).
- ◆ **ставка.** Процентная ставка за период. Например, если делать ежемесячные платежи, то надо делить ежегодную процентную ставку на 12.
- ◆ **без переключения.** ИСТИНА или ЛОЖЬ. Уточняет, переключаться ли на линейное начисление амортизации, если обесценивание больше величины, рассчитанной методом снижающегося остатка.

На рис. 12.18 показано вычисление обесценивания, с использованием описанных выше функции. Первоначальная стоимость актива \$10 000 устанавливается такой, чтобы иметь срок 10 лет с остаточным значением \$1 000. Диапазон *Declaration Amount* показывает ежегодное обесценивание актива. Диапазон *Value of Asset* показывает обесцененное значение актива на протяжении срока.

depreciation.xls		A	B	C	D	E
1	Активы		Мебель			
2	Начальная стоимость		\$10 000			
3	Срок службы (годы)		10			
4	Ликвидационная стоимость		\$1 000			
5						
6	Обесценивание					
7		Год	АПЛ	ПУО	ДДОБ	АСЧ
8		1	\$900.00	\$2 060.00	\$2 000.00	\$1 636.36
9		2	\$900.00	\$1 635.64	\$1 600.00	\$1 472.73
10		3	\$900.00	\$1 298.70	\$1 200.00	\$1 309.09
11		4	\$900.00	\$1 031.17	\$1 024.00	\$1 145.45
12		5	\$900.00	\$818.75	\$819.20	\$981.82
13		6	\$900.00	\$650.08	\$655.36	\$918.18
14		7	\$900.00	\$516.17	\$524.29	\$854.55
15		8	\$900.00	\$409.84	\$419.43	\$490.91
16		9	\$900.00	\$325.41	\$335.64	\$327.27
17		10	\$900.00	\$258.38	\$258.44	\$163.64
18						
19						
20	Активы					
21		Год	АПЛ	ПУО	ДДОБ	АСЧ
22		0	\$10 000.00	\$10 000.00	\$10 000.00	\$10 000.00
23		1	\$9 100.00	\$7 940.00	\$8 000.00	\$8 363.64
24		2	\$8 200.00	\$6 304.36	\$6 400.00	\$6 890.91
25		3	\$7 300.00	\$5 005.66	\$5 120.00	\$5 581.82
26		4	\$6 400.00	\$3 974.50	\$4 096.00	\$4 436.36
27		5	\$5 500.00	\$3 155.75	\$3 276.80	\$3 464.55
28		6	\$4 600.00	\$2 505.67	\$2 621.44	\$2 636.36
29		7	\$3 700.00	\$1 989.50	\$2 087.15	\$1 981.82
30		8	\$2 800.00	\$1 579.66	\$1 677.72	\$1 490.91
31		9	\$1 900.00	\$1 254.25	\$1 342.18	\$1 163.64
32		10	\$1 000.00	\$995.88	\$1 073.74	\$1 000.00

Рис. 12.18. Сравнение четырех функций обесценивания

На рис. 12.19 показана диаграмма изменения актива. Как вы видите, функция АПЛ имеет вид прямой линии; другие функции изогнуты из-за того, что обесценивание больше на ранних стадиях срока службы актива.

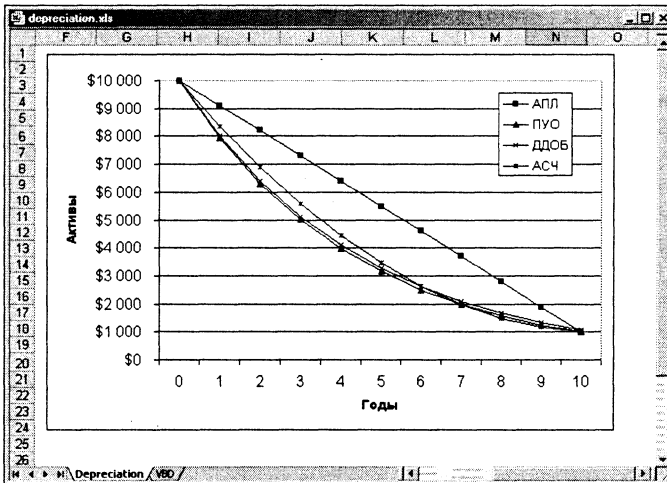


Рис. 12.19. На этой диаграмме показано значение актива, изменяемое во времени, рассчитанное по четырем функциям обесценивания

Функция ПУО полезна, если надо вычислить обесценивание на протяжении нескольких периодов (например, годы 2 и 3). На рис. 12.20 показан рабочий лист, настроенный на вычисление обесценивания с помощью функции ПУО. Формула в ячейке В12 следующая:

=ПУО (82; 84; 83; 86; 87; 88; 89)

Эта формула показывает обесценивание на протяжении первых трех лет актива (начальный период 0 и заключительный период 3).

	A	B	C
1	Активы	Мебель	
2	Начальная стоимость	\$10 000	
3	Срок службы (годы)	10	
4	Ликвидационная стоимость	\$1 000	
5			
6	нач. период	0	
7	кон. период	3	
8	коэффициент	2	
9	без переключения	ИСТИНА	
10			
11			
12	Обесценивание	\$4 880.00	
13			

Рис. 12.20. Использование функции ПУО для вычисления обесценивания на протяжении нескольких периодов

Резюме

В этой главе, мы завершили рассмотрение основных средств, требуемых для сложного финансового анализа.

В следующей главе описывается применение этих инструментов и иллюстрируется ряд полезных формул и вспомогательных методов.

Глава 13

Дополнительное использование финансовых функций и формул

В этой главе...

- ◆ Создание динамических финансовых планов
- ◆ Создание планов амортизации
- ◆ Задание параметров займа с помощью таблицы данных
- ◆ План накопления
- ◆ Планы будущего поступления
- ◆ Функции ЧИСТВНДОХ и ЧИСТНЗ
- ◆ Резюме

Приведенные в этой главе сведения базируются на материале предыдущих двух глав. В ней содержатся полезные примеры из широкого круга финансовых задач.

Создание динамических финансовых планов

Финансовый план — детальный список денежных потоков. Как правило, каждая строка представляет собой период времени (например, месяц), а информация на протяжении этого периода времени отображается в столбцах. Как известно, электронные крупноформатные таблицы идеальны для создания финансовых планов.

Наиболее полезный тип финансового плана — *динамический*. Он использует ячейки с вводимыми значениями (представляющие переменные) для самонастройки. Лучший динамичный план тот, который обеспечивает максимальную гибкость, и позволяет пользователю изменять любую из ключевых переменных, используемых в вычислениях. Легко предугадать, что пользователю захочется избежать жестко заданных значений внутри формул. Скорее всего значения должны сохраняться в ячейках, на которые ссылаются формулы.

Это задание становится немного неоднозначным, когда план включает переменные периоды времени — например, если пользователь вводит в ячейку срок займа. В таком случае количество строк в плане будет изменяться.

Наиболее динамичные планы содержат, по крайней мере, четыре основных элемента:

- ◆ Вводимые пользователем данные
- ◆ Промежуточные вычисления
- ◆ Отображаемые результаты
- ◆ Непосредственно план

Эти элементы можно сохранить на одном рабочем листе или в нескольких рабочих листах. В остальной части этой главы представлены примеры типичных финансовых планов.

Создание планов амортизации

В самой простой форме план амортизации отслеживает платежи (включая процент и основные выплаты) и баланс определенного займа. В этом разделе представлено несколько примеров планов амортизации.

Простой план амортизации

В этом примере описывается простой заем для демонстрации основных понятий, используемых для создания действующего плана. Обратитесь к рабочему листу на рис. 13.1.

The screenshot shows an Excel spreadsheet with the following structure:

- Row 1:** Title "Расписание амортизации простого займа" (Amortization schedule of a simple loan).
- Section 3: "Задаётся пользователем" (User-defined):**
 - 4 Заем: \$150,000.00
 - 5 Ставка: 8%
 - 6 Тип: Nominal
 - 7 Выплаты: 4 Quarterly
 - 8 Количество выплат: 4
 - 9 Срок (годы): 2
- Section 11: "Промежуточные вычисления" (Intermediate calculations):**
 - 12 Учетная ставка за период: 2,00000% Квартально
 - 13 Количество периодов: 8
 - 14 Платеж: (\$20,476.47)
- Section 16: "Сводные данные" (Summary statistics):**
 - 17 Суммарный выплачиваемый процент: (\$13,811.76)
- Section 19: "Расписание амортизации" (Amortization schedule):**

Период	Платеж	Процент	Основной платеж	Баланс
0				\$150,000.00
1	(\$20,476.47)	(\$3,000.00)	(\$17,476.47)	\$132,523.53
2	(\$20,476.47)	(\$2,650.47)	(\$17,826.00)	\$114,697.53
3	(\$20,476.47)	(\$2,293.95)	(\$18,182.52)	\$96,515.01
4	(\$20,476.47)	(\$1,930.30)	(\$18,546.17)	\$77,968.84
5	(\$20,476.47)	(\$1,559.38)	(\$18,917.09)	\$59,051.75
6	(\$20,476.47)	(\$1,181.03)	(\$19,295.43)	\$39,756.31
7	(\$20,476.47)	(\$795.13)	(\$19,681.34)	\$20,074.97
8	(\$20,476.47)	(\$401.50)	(\$20,074.97)	(\$0.00)
9	\$0.00	\$0.00	\$0.00	\$0.00

Рис. 13.1. Простой план амортизации

Вводимые данные

Область вводимых пользователем данных — диапазон В4:В9. В этом примере ячейка В6 содержит простой список, позволяющий проводить проверку рассчитанных данных, задающий одну из двух строк: Nominal или Effective. В ячейке С7 содержится формула, в которой применена пользовательская функция VBA:

=FreqName (В7)

Эта формула возвращает текстовую строку, которая описывает тип ставки, вводимой в ячейку В7. Все другие ячейки в разделе вводимых пользователем данных содержат значения.

Промежуточные вычисления

В этом примере промежуточные вычисления выполняются в диапазоне В12:В14. В ячейке В12 используются пользовательские функции VBA для вычисления периодической учетной ставки на основе информации из раздела входных данных:

```
=ЕСЛИ(В5="Nominal";Nomx_Effy(В4;В6;В7);Effx_Effy(В4;В6;В7))
```

В ячейке В13 содержится простая формула, вычисляющая количество рассматриваемых периодов (то есть, количество строк в плане):

```
=В9*В8
```

Ячейка В14 использует функцию ПЛТ для вычисления периодического платежа:

```
=ПЛТ(В12;В13;84;0;0)
```

Возвращаемые данные

В этом примере, в разделе возвращаемых данных, содержится только одна формула — в ячейке В17. Эта формула вычисляет суммарный выплаченный процент:

```
=СУММ(С21:С381)
```



Размещение выходных данных непосредственно сверху плана устраняет необходимость прокрутки в конец рабочего листа.

План

План амортизации начинается в строке 20, которая содержит описательные названия столбцов. Стандартный подход заключается в жестком задании “нулевого” и первого периодов, а также использовании формул для получения последующих периодов выплат. В этом примере ячейки А21 и А22 содержат жестко заданные значения. А вот ячейки А23 и ниже содержат формулы. Формула в ячейке А23 следующая:

```
=ЕСЛИ(А22<=$В$13;ЕСЛИ(А22=0;0;А22+1);0)
```

Эта формула копируется вниз до ячейки А381. Формула добавляет к количеству периодов времени 1 до тех пор, пока не будет достигнуто общее количество периодов. Когда текущий период превышает общее число, формула возвращает 0. В нашем примере это происходит в ячейке А30.

Каждая ячейка с формулой (столбцы В: F) в плане обращается к периоду времени в соответствующей строке. Если период времени не равен 0, формула возвращает результат. В противном случае она возвращает 0.

Формула в ячейке В22, содержащая периодический платеж, следующая:

```
=ЕСЛИ(А22=0;0;$В$14)
```

Процент вычисляется в результате умножения предыдущего баланса на процентную ставку за период. Основной платеж равен разнице платежа и процента. Наконец, новый баланс вычисляется как сумма (отрицательной) начального капитала и предыдущего баланса. Формула вычисления процента в ячейке С22 следующая:

```
=ЕСЛИ(А22=0;0;-Е21*$В$12)
```

Основной платеж вычисляется по следующей формуле (ячейка D22):

```
=ЕСЛИ(А22=0;0;В22-С22)
```

Новый баланс (ячейка E22) вычисляется по следующей формуле:

=ЕСЛИ (A22=0; 0; E21+D22)

Эти формулы копируются вниз, насколько позволяет разумный максимум количества строк (в этом примере они копируются вниз до строки 381). Обратите внимание, что эти формулы возвращают ненулевое значение, только если в столбце А содержится ненулевой период.



Чтобы скрыть нули в неиспользованных строках, можно выполнить команду Параметры из меню Сервис, перейти вкладку Вид и сбросить флажок опции Нулевые значения. Другой вариант — использовать пустую строку ("") вместо 0 в формулах. Существует и третий вариант — использовать команду Автофильтр для скрытия неиспользованных строк.

Планы амортизации займа самопроверяющиеся. Если все значения заданы правильно, конечный баланс в конце срока равен 0 (или очень близок к 0, учитывая ошибку округления). Другой способ проверки — добавить начальные значения. Сумма этих значений должна равняться исходной сумме займа.

Детальный план амортизации

Пример этого раздела основан на предыдущем примере. На рис. 13.2 показан более детальный план амортизации займа, который рассматривает влияние платы на получение займа, затрат на поддержку счета и влияние налога на процентную ставку.

detailed loan amortization schedule.xls									
A	B	C	D	E	F	G	H	I	J
Детальное расписание амортизации займа									
2 Задается пользователем									
3	Заем		\$3 000 000						
4	Срок займа (годы)			2	Срок займа* количество платежей не должно превышать 360				
5	Номинальная годовая ставка		15,00%						
6	Количество платежей за год			4	Срок займа* количество платежей не должно превышать 360				
7	Создание счета (% от суммы займа)		2,00%						
8	Содержание счета (% от суммы займа)		0,05%						
9	Налоги		49,25%						
10									
11 Вычисляемые данные					Сводная информация				
12	Общее количество периодов			8	Общая сумма выплат (со стоимостью счета)				\$3 599 961
13	Эффективная сумма займа		\$2 940 000		Общий процент				\$527 961
14	Выплаты по займу (без стоимости счета)		(\$440 995)		Содержание счета				\$72 000
15	Выплаты по займу (со стоимостью счета)		(\$442 495)		Общий платеж				\$3 000 000
16	Эффективный эквивалент номинальной ставки		15,8650%		Выплаты минус общий платеж				\$599 961
17	Эффективная стоимость займа без учета налогов		18,4411%		Налоги				\$295 481
18	Эффективная стоимость займа с учетом налогов		9,0004%		Всего				\$304 480
19									
20 Amortization Schedule									
21	Период	Платежи без стоимости счета	Платежи со стоимостью счета	Процент	Основной платеж	Баланс	Налоги	Денежные потоки без налогов	Денежные потоки с налогами
22	0		(\$60 000)			\$3 000 000	\$29 550	\$2 940 000	\$2 969 550
23	1	(\$440 995)	(\$442 495)	\$112 500	\$328 495	\$2 671 505	\$56 145	(\$442 495)	(\$386 350)
24	2	(\$440 995)	(\$442 495)	\$100 181	\$340 814	\$2 330 691	\$50 078	(\$442 495)	(\$392 417)
25	3	(\$440 995)	(\$442 495)	\$87 401	\$353 594	\$1 977 097	\$43 784	(\$442 495)	(\$398 711)
26	4	(\$440 995)	(\$442 495)	\$74 141	\$366 954	\$1 610 243	\$37 253	(\$442 495)	(\$405 242)
27	5	(\$440 995)	(\$442 495)	\$60 384	\$380 611	\$1 229 632	\$30 478	(\$442 495)	(\$412 017)
28	6	(\$440 995)	(\$442 495)	\$46 111	\$394 884	\$834 748	\$23 449	(\$442 495)	(\$419 047)
29	7	(\$440 995)	(\$442 495)	\$31 303	\$409 692	\$425 056	\$16 155	(\$442 495)	(\$426 340)

Рис. 13.2. Детальный план амортизации

При рассмотрении этого примера учтите следующие моменты:

- ♦ Эффективный заем определен в главе 11 как занимаемая сумма минус стоимость получения займа. Выплаты займа основываются на сумме займа, но эффективная стоимость основывается на эффективном займе.
- ♦ Платежи вычисляются с использованием функции ПЛТ, но фактические выплаты включают сумму на обслуживание счета.

- ◆ В этом примере учет налога сказывается только на процентном компоненте займа. Налоговые законы могут изменяться.
- ◆ Вычисление эффективного эквивалента номинальной ставки проводится по функции ЭФФЕКТ.
- ◆ Эффективная стоимость займа перед учетом налога (ячейка D17) вычисляется с помощью функции ВСД в столбце Н. Эффективная стоимость займа после включения налога (ячейка D18) вычисляется с помощью функции ВСД в столбце I.
- ◆ План имеет общее количество 360 периодов. При превышении этого количества появится сообщение об ошибке.
- ◆ План самопроверяющийся. Конечный баланс равен нулю, а общая выплаченная сумма равняется первоначальной сумме займа.

План амортизации займа с переменной ставкой

Все планы амортизации, представленные в этой главе, основываются на займах с фиксированной процентной ставкой. Много займов, однако, являются займами с *переменной процентной ставкой* и используют разные процентные ставки во время срока выплат. Обычно эти займы построены таким образом, что платежи изменяются наряду со ставкой.

Главное изменение по сравнению с предыдущим примером — использование относительно простой формулы вычисления повторных платежей займа перед выплатами за открытие счета (столбец С).

Платежи займа (без платы за открытие счета) на протяжении каждого периода рассчитываются по функции ПЛТ, организованной следующим образом: ставка займа базируется на ставке для определенного периода (в столбце В), разделенной на частоту выплат займа. Срок займа для каждого периода вычисляется как максимальный срок займа минус количество периодов для предыдущей строки. Таким образом, срок займа повторно определяется для каждого платежа в столбце. Текущий заем (ПС) — баланс, известный для предыдущего периода. Опять-таки, повторно вычисляется заем для каждого платежа. Конечная формула для платежей за первый период (ячейка С23) следующая:

=ЕСЛИ (A23=0; 0; ПЛТ (B23/\$E\$5; МАКС (A22:A382) -A22; G22; 0; 0))

Ячейка В23 содержит процентную ставку для периода, а ячейка Е5 — частоту выплат.



Этот план работает, потому что в любой момент на протяжении займа рассчитываемые платежи производятся из невыплаченного баланса до окончания срока. Если заемщик решил вместо изменения платежей модифицировать срок займа, то нужно подкорректировать значения в столбце сроков с помощью функции ЕСЛИ, в которой используется функция КПЕР.

На рис 13.3 показан динамический план амортизации займа с переменной процентной ставкой. Пользователь может ввести ставки займа в столбец В. Конечно, главная проблема состоит в том, что ставки займа часто базируются на коэффициенте, так что ставки не известны заранее. Значит этот тип плана амортизации основывается на *предположениях* о будущих ставках.

variable rate loan amortization schedule.xls										
A	B	C	D	E	F	G	H	I	J	K
Расписание амортизации займа с переменной ставкой										
2	Задается пользователем									
3	Основной заем			\$3 000 000						
4	Срок займа (год)			2	Срок займа* количество платежей не должно превышать 360					
5	Количество выплат в год			4	Срок займа* количество платежей не должно превышать 360					
6	Создание счета (% от суммы займа)			2,00%						
7	Содержание счета (% от суммы займа)			0,05%						
8	Налоги			49,25%						
11	Вычисляемые данные				Сводная информация					
12	Общее количество периодов			8	Общая сумма выплат (со стоимостью счета)				\$3 321 416	
13	Эффективная сумма займа			\$2 940 000	Общий процент				\$249 416	
14	Эффективная стоимость займа без учета налогов			9,7899%	Содержание счета				\$72 000	
15	Эффективная стоимость займа с учетом налогов			4,8515%	Общий платеж				\$3 000 000	
16					Выплаты минус общий платеж				\$321 416	
17					Налоги				\$158 298	
18					Всего				\$183 119	
20	Расписание амортизации									
21	Период	Ставка	Платежи без стоимости счета	Платежи со стоимостью счета	Процент	Основной платеж	Баланс	Налоги	Денежные потоки без налогов	Денежные потоки с налогами
22	0			(\$60 000)			\$3 000 000	\$29 550	\$2 940 000	\$2 968 550
23	1	7,00%	(\$405 129)	(\$406 629)	\$52 500	\$352 629	\$2 647 371	\$26 595	(\$406 629)	(\$380 034)
24	2	7,00%	(\$405 129)	(\$406 629)	\$48 328	\$358 800	\$2 288 571	\$23 556	(\$406 629)	(\$363 073)
25	3	7,00%	(\$405 129)	(\$406 629)	\$40 050	\$365 079	\$1 923 493	\$20 463	(\$406 629)	(\$346 165)
26	4	7,00%	(\$405 129)	(\$406 629)	\$33 661	\$371 468	\$1 552 025	\$17 317	(\$406 629)	(\$329 312)
27	5	8,00%	(\$407 599)	(\$409 099)	\$31 041	\$376 558	\$1 176 467	\$16 026	(\$409 099)	(\$393 072)
28	6	8,00%	(\$407 599)	(\$409 099)	\$23 509	\$384 089	\$791 378	\$12 317	(\$409 099)	(\$396 782)
29	7	7,50%	(\$406 852)	(\$408 352)	\$14 838	\$392 014	\$399 364	\$8 047	(\$408 352)	(\$400 305)

Рис. 13.3. План амортизации займа с переменной процентной ставкой

Задание параметров займа с помощью таблицы данных

Команда Данные⇒Таблица в Excel является удобным инструментом представления различных параметров займа. В этом разделе описывается метод создания таблиц данных с одной и двумя переменными.

Создание односторонней таблицы данных

Односторонняя таблица данных представляет результаты любого числа вычислений для различных значений, введенных в одной ячейке.

На рис. 13.4 изображена односторонняя таблица данных (в B10: I13), в которой показаны три вычисления (платеж, основной платеж и общий процент) для займа с семью процентными ставками от 7,00% до 8,50%. В данном примере ячейкой вводимых данных является B2.

loan data tables.xls										
A	B	C	D	E	F	G	H	I	J	
1	Заем		\$10 000,00							
2	Годовая ставка		7,25%							
3	Период выплат (месяцы)		1							
4	Number of Periods		36							
5										
6	Платеж		\$309,92							
7	Общая сумма выплат		\$1 156,95							
8	Общий процент		\$1 156,95							
10				7,00%	7,25%	7,50%	7,75%	8,00%	8,25%	8,50%
11	Платеж		\$309,92	\$306,77	\$309,92	\$311,06	\$312,21	\$313,36	\$314,52	\$315,68
12	Общая сумма выплат		\$1 156,95	\$1 115,75	\$1 156,95	\$1 198,24	\$1 239,62	\$1 281,09	\$1 322,66	\$1 364,31
13	Общий процент		\$1 156,95	\$1 115,75	\$1 156,95	\$1 198,24	\$1 239,62	\$1 281,09	\$1 322,66	\$1 364,31

Рис. 13.4. Использование односторонней таблицы данных для представления трех результатов вычисления займа для различных процентных ставок

Для создания таблицы данных с одной переменной выполните следующие действия:

1. Введите формулы, которые возвращают результаты, используемые в таблице данных. В этом примере формулы находятся в диапазоне В6 : В8.
2. Введите различные значения для одной ячейки вводимых данных в последующих столбцах. В этом примере вводимое значение — процентная ставка, а значения различных процентных ставок отображаются в ячейках С10 : I10.
3. Создайте ссылку на ячейки с формулами в столбце слева от вводимых значений. В нашем примере диапазон В11 : В13 содержит простые формулы со ссылками на другие ячейки. Например, В11 содержит следующую формулу:

=В6

4. Выделите прямоугольный диапазон, который содержит элементы предыдущих действий. Для данного примера выберите диапазон В10 : I13.
5. Из меню Данные выберите команду Таблица подстановки. Excel отобразит диалоговое окно Таблица подстановки, показанное на рис. 13.5.
6. В поле Подставлять значения по столбцам в определите ссылку на ячейку, которая соответствует переменной в вашей таблице подстановки. Для этого примера ячейкой вводимых данных является В2.
7. Оставьте поле Подставлять значения по строкам в пустым.
8. Щелкните на кнопке ОК. Excel вставит формулу массива, которая создается командой Таблица подстановки с одной переменной.
9. При желании можно изменить форматирование таблиц данных. Например, можно изменить заливку заголовков строк и столбцов.

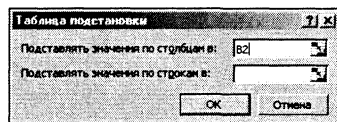


Рис. 13.5. Диалоговое окно Таблица подстановки

Обратите внимание, что формула массива не вводится в весь диапазон, выделенный в п. 4. Первый столбец и первая строка выделения не изменяются.



При создании таблицы подстановки, крайний слева столбец таблицы данных (столбец, который содержит ссылки, вводимые в п. 3) содержит вычисленные значения для ячейки ввода. В этом примере значения повторяются в столбце D. Можно скрыть значения в столбце E, если сделать шрифт одного цвета с фоном.

Создание двухсторонней таблицы подстановки

В двухсторонней таблице подстановки показаны результаты вычислений для различных значений в двух ячейках ввода данных. На рис. 13.6 изображена таблица подстановки с двумя переменными (в В10 : I16), демонстрирующая вычисление (сумму платежа) для займа с использованием семи процентных ставок и шести сумм займа.

	A	B	C	D	E	F	G	H	I
1	Заем	\$10 000,00							
2	Годовая ставка	7,25%							
3	Период выплат (месяцы)	1							
4	Number of Periods	36							
5									
6	Платеж	\$309,92							
7	Общая сумма выплат	\$11 156,95							
8	Общий процент	\$1 156,95							
9									
10		\$309,92	7,00%	7,25%	7,50%	7,75%	8,00%	8,25%	8,50%
11		\$9 000,00	\$277,89	\$278,92	\$279,96	\$280,99	\$282,03	\$283,07	\$284,11
12		\$9 500,00	\$293,33	\$294,42	\$295,51	\$296,60	\$297,70	\$298,79	\$299,89
13	Заем	\$10 000,00	\$308,77	\$309,92	\$311,06	\$312,21	\$313,36	\$314,52	\$315,68
14		\$10 500,00	\$324,21	\$325,41	\$326,62	\$327,82	\$329,03	\$330,24	\$331,46
15		\$11 000,00	\$339,65	\$340,91	\$342,17	\$343,43	\$344,70	\$345,97	\$347,24
16		\$11 500,00	\$355,09	\$356,40	\$357,72	\$359,04	\$360,37	\$361,70	\$363,03
17									
18									
19									

Рис. 13.6. Использование таблицы данных с двумя переменными для получения величины платежей для различных сумм займа и процентных ставок

Для создания этой таблицы подстановки с двумя переменными выполните следующие действия:

1. Введите формулу, возвращающую результаты, которые будут использоваться в таблице подстановки. Для этого примера формула находится в ячейке В6. В диапазоне В7 : В8 формулы не используются.
2. Введите различные значения для первого вводимого параметра в следующих столбцах. Для этого примера первый параметр — это процентная ставка, а значения для различных процентных ставок появляются в С10 : I10.
3. Задайте различные значения второго вводимого параметра в следующих строках, левее и ниже значений первого параметра. В этом примере второй переменной будет сумма займа, а значения для различных сумм займа располагаются в В11 : В16.
4. Создайте ссылку на формулу, которая подсчитывается в таблице. Эта ссылка расположена в верхнем левом углу диапазона таблицы подстановки. В этом примере в ячейке В10 содержится следующая формула:

=В6

5. Выделите прямоугольный диапазон, который содержит данные предыдущих пп. В этом примере выделите В10 : I16.
6. Выберите Данные⇒Таблица Подстановки. Excel отобразит диалоговое окно Таблица подстановки.
7. В поле Подставлять значения по столбцам в определите ссылку на ячейку, которая соответствует первой переменной в вашей таблице подстановки. Для этого примера первой ячейкой вводимых данных является В2.
8. В поле Подставлять значения по строкам в определите ссылку на ячейку, которая соответствует второй переменной в вашей таблице подстановки. Для этого примера второй ячейкой вводимых данных является В1.
9. Щелкните на кнопке ОК. Excel вставит формулу массива, которая использует функцию ТАБЛИЦА с двумя переменными.

После создания двухсторонней таблицы подстановки можно изменить выходные данные благодаря заданию другой ссылки на ячейку в верхней левой ячейке таблицы данных. В этом

примере можно изменить формулу в ячейке B10 на =B8 — в таблице подстановки данных теперь отобразится скорее общий процент, чем сумма платежа.



Если вы находите, что использование таблиц данных замедляет вычисление рабочей книги, выберите команду Параметры из меню Сервис. В диалоговом окне Параметры перейдите на вкладку Вычисления и выставьте переключатель Автоматически кроме таблиц.

План накопления

План накопления похож на план амортизации, но потоки наличных в нем могут быть как входящими, так и исходящими. Можно использовать план накопления для детального вычисления счета с разными уровнями регулярных вкладов и выплат или случайных капиталовложений. На рис. 13.7 показан пример такого плана.

Период	Регулярные выплаты	Разовые вклады	Регулярные выплаты	Разовые выплаты	Ставка	Процент	Баланс
0	\$250,00	\$5 000,00	(\$200,00)				\$5 050,00
1	\$250,00		(\$200,00)		7,00%	\$26,55	\$5 126,55
2	\$250,00		(\$200,00)		7,00%	\$29,00	\$5 207,55
3	\$250,00		(\$200,00)	(\$3 000,00)	7,00%	\$29,44	\$2 267,00
4	\$250,00		(\$200,00)		6,50%	\$12,03	\$2 349,03
5	\$250,00		(\$200,00)		6,50%	\$12,36	\$2 411,39
6	\$250,00	\$10 000,00	(\$200,00)		6,50%	\$12,69	\$12 474,06
7	\$250,00		(\$200,00)		6,50%	\$65,63	\$12 589,71
8	\$250,00		(\$200,00)		6,50%	\$66,24	\$12 705,95
9	\$250,00		(\$200,00)	(\$3 000,00)	6,00%	\$61,85	\$9 617,80
10	\$250,00		(\$200,00)		6,00%	\$47,79	\$9 915,59
11	\$250,00		(\$200,00)		6,00%	\$48,26	\$10 013,85
12	\$250,00		(\$200,00)		6,00%	\$48,74	\$10 112,60

Рис. 13.7. План накопления

Наиболее сложная часть этого плана связана с учетной ставкой и вычислением процента. Пользователь вводит ежегодную ставку в столбец F, выбирает ее тип (ячейка C3), указывает количество периодов сложных процентов (ячейка C4) и частоту платежей (ячейка C5).

Вычисление процента зависит от выбранного типа ставки и проводится стандартным методом, описанным в предыдущей главе, в котором используются пользовательские функции VBA. Формула в ячейке G10, например, имеет следующий вид:

=ЕСЛИ(\$C3="Nominal";Nomx_Effy(F10;\$C\$4;\$C\$5);Effx_Effy(F10;\$C\$4;\$C\$5))*H9

В этой формуле:

- ♦ Ячейка C3 — абсолютная ссылка на тип процентной ставки (номинальная или эффективная).
- ♦ Ячейка F10 — ставка для текущего периода.
- ♦ Ячейка C4 — абсолютная ссылка на количество платежей за период.
- ♦ Ячейка C5 — абсолютная ссылка на частоту платежей (количество периодов).
- ♦ Ячейка H9 — баланс для предыдущего периода. Баланс — сумма предыдущего баланса, платежей и начислений.

Мы рассмотрели здесь только 12 периодов, но план может продолжаться столько времени, сколько угодно.

Планы будущего поступления

План будущего поступления — это методика анализа инвестиции, в которой участвуют функции ЧПС и ВДС, описывающие положительные и отрицательные денежные потоки. Функция ЧПС вычисляет разницу между будущими положительными и отрицательными начислениями. Функция ВДС представляет доходность за период анализа денежных поступлений.

Планы будущих поступлений могут быть очень обширными и включать сложные расчеты главных элементов. Однако основы вычислений относительно просты и требуют совсем небольшого преобразования формул и функций.

На рис. 13.8 показан основной план будущих поступлений со всеми необходимыми элементами, включая следующие:

- ♦ Частота платежей (ячейка С3), которая измеряется в тех же единицах, что и ВДС. ВДС (ячейка С7) рассматривается как ставка на протяжении периода платежа и используется для вычисления ЧПС.
- ♦ Начальное значение (ячейка С4), которое принимается за основной капитал и имеет отрицательное значение.
- ♦ Конечное значение (ячейка С5), которое принимается за будущий капитал и имеет положительное значение.
- ♦ Ставка (ячейка С10), необходимая для вычисления ЧПС.
- ♦ Непосредственно план, детализирующий капитал, доходы и издержки. Они суммируются, чтобы представить денежный поток за период.

discounted cash flow schedule.xls					
	A	B	C	D	E
1	План поступления				
2					
3	Частота платежей			4	Quarterly
4	Начальное значение		(\$1 000 000,00)		
5	Конечное значение		\$1 200 000,00		
6					
7	ВДС			4,22%	Quarterly
8	Годовой доход			18,00%	
9					
10	Ставка			11,00%	р. а. Effective
11	ЧПС @ 0,11 р. а. Effective		\$169 808,70		
12	С учетом ЧПС		(\$1 169 808,70)		
13					
14	период	Основной капитал	Поступления	Отчисления	Платежи
15	0	(\$1 000 000,00)	\$34 000,00	(\$8 000,00)	(\$974 000,00)
16	1		\$34 000,00	(\$8 000,00)	\$26 000,00
17	2		\$34 000,00	(\$8 000,00)	\$26 000,00
18	3		\$34 000,00	(\$8 000,00)	\$26 000,00
19	4		\$36 000,00	(\$10 000,00)	\$26 000,00
20	5		\$36 000,00	(\$10 000,00)	\$26 000,00
21	6		\$36 000,00	(\$10 000,00)	\$26 000,00
22	7		\$36 000,00	(\$10 000,00)	\$26 000,00
23	8		\$43 000,00	(\$12 500,00)	\$30 500,00
24	9		\$43 000,00	(\$12 500,00)	\$30 500,00
25	10		\$43 000,00	(\$12 500,00)	\$30 500,00
26	11		\$43 000,00	(\$12 500,00)	\$30 500,00
27	12	\$1 200 000,00			\$1 200 000,00
28					

Рис. 13.8. План будущих поступлений

В этом примере платежи проводятся поквартально. Таким образом, ВДС — эффективная квартальная ставка. Для превращения ее в ежегодный эффективный эквивалент используется пользовательская функция `VBA Effx_AnnEff`. Формула в ячейке С8 следующая:

`=Effx_AnnEff (С7;С3)`

Учетная ставка требуется для вычислений ЧПС и определяется как ежегодная эффективная ставка в ячейке С10. Учетная ставка преобразуется перед использованием в функции ЧПС. Формула в ячейке С11 следующая:

`=ЧПС (AnnEff_Effx (С10;С3) ; E15 : E27) * (1+AnnEff_Effx (С10;С3))`

В этой формуле ячейка С10 содержит учетную ставку, С3 — частоту платежей, а диапазон E15 : E27 — суммы платежей (включая платеж в точке 0).

Исходя из главы 12, следующая формула используется для вычисления ЧПС при задании начального потока:

`=ЧПС (ставка ; диапазон) * (1+ставка)`

Вычислив ЧПС, становится возможным получить производное начальное значение, основанное на ставке 11%. Это начальное значение получается в результате вычитания подсчитанного ЧПС из существующего начального значения \$1 000 000.

Этот пример показал, что план будущих поступлений — это предмет первой необходимости в арсенале любого экономиста. На практике все описанные выше вычисления могут применяться к самым разным задачам.

Функции ЧИСТВНДОХ и ЧИСТНЗ

Как обсуждалось в главе 12, функции ВДС и ЧПС подразумевают регулярные периодические денежные потоки. В некоторых ситуациях, однако, денежные потоки не регулярны. В таком случае можно использовать функции ЧИСТВНДОХ и ЧИСТНЗ. Эти функции вычисляют ВДС и ЧПС незапланированных денежных потоков, и они используют ежедневный эффективный эквивалент указанной или (в случае ЧИСТНЗ) подсчитанной ежегодной эффективной ставки.



Функции ЧИСТВНДОХ и ЧИСТНЗ доступны после установки надстройки «Пакет анализа».

Функция ЧИСТВНДОХ возвращает ежегодную эффективную ставку доходности и имеет следующий синтаксис (обязательные параметры выделены полужирным шрифтом):

ЧИСТВНДОХ (**значения** ; **даты** ; предположение)

Синтаксис функции ЧИСТНЗ следующий (все параметры обязательны):

ЧИСТНЗ (**ставка** ; **значения** ; **даты**)

На рис. 13.9 показан рабочий лист, открытый для потоков наличных наряду с датами поступления.

Формула в ячейке В15 следующая:

`=ЧИСТВНДОХ (В4 : В13 ; А4 : А13)`

Обратите внимание, что ЧИСТВНДОХ принимает ежегодную эффективную ставку, которая базируется на предположении 365-дневного годового периода. План дат должен быть последовательным от самой ранней до самой поздней, при этом даты не должны повторяться.

Вычисление ЧИСТВНДОХ можно проверить с помощью функции ЧИСТНЗ, возвращающей

ставку в подсчитанном ЧИСТВНДОХ. Учетная ставка рассматривается как ежегодная эффективная ставка.

Формула в ячейке В16, которая возвращает 0, следующая:


=ЧИСТНЗ(В15;В4:В13;А4:А13)

На рис. 13.10 демонстрируется функция ЧИСТНЗ и показан открытый рабочий лист с потоками наличных наряду с датами поступления.

Тип процентной ставки (ячейка В4) определяет способ проверки данных и позволяет пользователю выбрать номинальную или эффективную норму. Преобразование ставки в ежегодную эффективную проводится с помощью пользовательской функции VBA. Формула в ячейке В7 следующая:

=ЕСЛИ(В4="Nominal";Nomx_AnnEff(В3;В5);Efx_AnnEff(В3;В5))

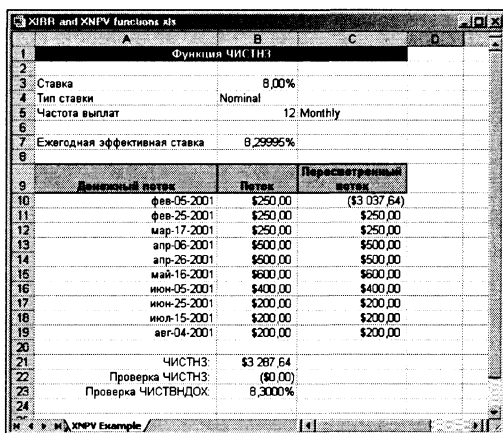
Если указана номинальная ставка, то она преобразуется в ежегодную эффективную ставку, запрашиваемую функцией ЧИСТНЗ. Если указана эффективная ставка, то она преобразуется в ежегодную эффективную ставку.



Дата	Поступления
фев-05-2001	(\$3 000,00)
фев-25-2001	\$250,00
мар-17-2001	\$250,00
апр-06-2001	\$500,00
апр-26-2001	\$500,00
май-16-2001	\$600,00
июн-05-2001	\$400,00
июн-25-2001	\$200,00
июл-15-2001	\$200,00
авг-04-2001	\$200,00

ЧИСТВНДОХ: 13,7506%
Проверка ЧИСТНЗ: (\$0,00)

Рис. 13.9. Использование функции ЧИСТВНДОХ



Денежный поток	Платеж	Пересмотренный платеж
фев-05-2001	\$250,00	(\$3 037,64)
фев-25-2001	\$250,00	\$250,00
мар-17-2001	\$250,00	\$250,00
апр-06-2001	\$500,00	\$500,00
апр-26-2001	\$500,00	\$500,00
май-16-2001	\$600,00	\$600,00
июн-05-2001	\$400,00	\$400,00
июн-25-2001	\$200,00	\$200,00
июл-15-2001	\$200,00	\$200,00
авг-04-2001	\$200,00	\$200,00

ЧИСТНЗ \$3 287,64
Проверка ЧИСТНЗ (\$0,00)
Проверка ЧИСТВНДОХ 8,3000%

Рис. 13.10. Использование функции ЧИСТНЗ



В отличие от функции ЧПС, нет надобности умножить ЧИСТНЗ на коэффициент $(1+ставка)$. По всей видимости, Excel использует стандартное определение ЧПС (глава 12). Однако при использовании ежедневных эффективных ставок разница несущественна.

Результат, возвращаемый ЧИСТНЗ, проверяется следующим образом: определяется исправленный денежный поток (в столбце С), равный ЧИСТНЗ к которому добавлен начальный платеж с противоположным знаком. Исправленный поток приводит к ЧИСТНЗ, равному 0, если использовать тот же процент, а ЧИСТВНДОХ возвращает процент, применяемый для вычисления исходного ЧИСТНЗ.



Функция ЧИСТВНДОХ плохо применима при использовании нескольких внутренних ставок доходности. В подобном случае ЧИСТВНДОХ равен 0, даже если ЧИСТНЗ не равно 0. Соответственно, там где используется несколько значений ЧПС (если знак изменяется больше одного раза), необходимо проверить ЧИСТВНДОХ с помощью функции ЧИСТНЗ. Если результат не равен 0, то ответ можно получить с помощью вычисления текущих значений для каждого платежа, используя команду Поиск решения для получения ставки, при которой сумма текущих значений и равняется 0. К счастью, проблема очень редка даже при изменении знака платежей и появляется только в случае, если денежный поток производится в начальный момент.

Резюме

В этой главе представлены примеры обычных финансовых исследований. В примерах используются основные понятия временного изменения денежных потоков и эквивалентных процентных ставок.

На этом часть, посвященная финансовым формулам, завершается. В следующей части рассматриваются дополнительные способы вычислений.

Часть IV

Формулы массива

ГЛАВА 14

“Введение в массивы”

ГЛАВА 15

“Магия формул массива”

Глава 14

Введение в массивы

В этой главе...

- ◆ Введение в формулы массива
- ◆ Понятие размерности массива
- ◆ Присвоение имени массивам констант
- ◆ Работа с формулами массива
- ◆ Использование формул массива, возвращающих несколько значений
- ◆ Использование формул массива, возвращающих одно значение
- ◆ Резюме

Одна из наиболее интересных и наиболее сложных возможностей Excel — работа с формулами массива. С их помощью можно творить настоящие чудеса и стать мастером в Excel. Главное — понять концепцию формул массива, именно этому и посвящена настоящая глава. Глава 15 продолжает изложение материала этой главы на основе многочисленных примеров.

Введение в формулы массива

В компьютерном программировании очень часто вводится понятие массива для решения сложных задач. *Массив* — это совокупность элементов, которые воспринимаются как единый объект вместе или по отдельности. В Excel массивы могут быть одномерными (их еще называют векторами) или двумерными (матрицы). Размерность массива зависит от количества строк и столбцов в нем. Например, *одномерный массив* может храниться в диапазоне, состоящем из одной строки (горизонтальный массив) или столбце (вертикальный массив). *Двумерный массив* хранится в прямоугольном диапазоне ячеек. Excel (в отличие от языка программирования VBA) не поддерживает трехмерные массивы.

Однако, как будет показано далее, массивы вовсе не обязательно хранить в ячейках. Можно производить операции над массивами, которые существуют только в памяти Excel. Можно применять *формулы массива* для выполнения операций и получения результатов вычислений. Формула массива может находиться во многих ячейках или в одной единственной.

В этой главе представлено два вида формул массива: формула массива, которая возвращает несколько значений, и другая, которая возвращает только одно значение.

Формула массива, возвращающая несколько значений

Рис. 14.1 демонстрирует рабочий лист, на котором производится простейший расчет объема продажи товаров. Для этого вычисляют значение в столбце D (объем продаж каждого товара), используя приведенную формулу, а затем копируют ее в расположенные ниже ячейки:

=B2*C2

После копирования этой формулы рабочий лист содержит шесть формул в столбце D. Альтернативный путь вычислений — использование *единой* формулы (формулы массива) для вычисления всех шести значений в диапазоне D2 : D7. Эта единственная формула занимает шесть ячеек и возвращает массив из шести значений.

Для создания единой формулы массива нужно произвести следующие операции:

1. Выделить диапазон, в котором будет сохраняться результат вычислений. В данном случае это диапазон D2 : D7.
2. Ввести формулу:
 $=B2 : B7 * C2 : C7$
3. Обычно после введения формулы нажимается клавиша <Enter>, но только не в случае формулы массива. Так как введенная формула является формулой массива, нажимается комбинация клавиш <Ctrl+Shift+Enter>.

Формула добавляется во все шесть выделенных ячеек, при этом в строке формул появится следующее выражение:

Product	Units Sold	Unit Price	Total
AR-988	3	\$50	\$150
BZ-011	10	\$100	\$1,000
MR-919	5	\$20	\$100
TR-811	9	\$10	\$90
TS-333	3	\$60	\$180
ZL-001	1	\$200	\$200

{=B2 : B7 * C2 : C7}

Excel заключил формулу в фигурные скобки. Это говорит о том, что данная формула является формулой массива.

Эта формула производит вычисления и возвращает массив из шести элементов. В данном случае формула массива производит операции над двумя другими массивами, оба из которых хранятся в диапазонах ячеек. Значения первого массива хранятся в диапазоне B2 : B7, а второго — в C2 : C7.

Рис. 14.1. Диапазон D2 : D7 содержит одну и ту же формулу массива

Для отображения итогового массива требуется шесть ячеек, потому что в одной ячейке нельзя отобразить более чем одно значение. Этим объясняется, зачем нужно выделять шесть ячеек перед введением формулы.

Конечно, эта формула массива возвращает точно такие же значения, что и шесть формул, введенных в соответствующие ячейки диапазона D2 : D7:

$=B2 * C2$
 $=B3 * C3$
 $=B4 * C4$
 $=B5 * C5$
 $=B6 * C6$
 $=B7 * C7$

Использование единой формулы массива предпочтительнее, чем использование нескольких формул, так как предоставляет определенные преимущества:

- ◆ Легко убедиться, что все формулы в диапазоне идентичны.
- ◆ Использование формул массива, возвращающих сразу несколько значений, избавляет от случайных изменений в формуле. Нельзя просто изменить одну ячейку в формуле массива.
- ◆ Использование формул массива почти всегда предотвращает вмешательство начинающих пользователей, которые хотят внести свои изменения в формулы.

Формула массива, возвращающая одно значение

Перейдем к рассмотрению формулы массива, возвращающей одно значение. Обратимся опять к рис. 14.1. Следующая формула массива размещается в одной ячейке:

```
{=СУММ(A1:A5*B1:B5)}
```

Эту формулу можно ввести в любую ячейку. При вводе формулы не нужно использовать фигурные скобки, а после введения не забудьте использовать комбинацию клавиш <Ctrl+Shift+Enter>.

Эта формула массива возвращает общую сумму, на которую продано товаров. Важно понимать, что значение, возвращаемое этой формулой, не зависит от информации в столбце D. Фактически можно удалить столбец D, а формула будет выполняться по-прежнему.

Эта формула оперирует значениями, находящимися в двух массивах. Оба массива хранятся в ячейках. Первый — в диапазоне ячеек B2:B7, второй — C2:C7. Формула перемножает соответствующие значения в этих двух массивах и создает новый массив (который существует только в памяти компьютера). Функция СУММ выполняет операцию над этим новым массивом и возвращает значение суммы его элементов.

Создание массива констант

В предыдущих примерах использовались массивы, хранящиеся в диапазоне рабочего листа. Примеры этого раздела демонстрируют важное положение: массив не обязательно должен храниться в диапазоне ячеек. Массив, который хранится исключительно в памяти, называется массивом констант.

Для создания массива констант нужно список его элементов заключить в фигурные скобки. Вот пример горизонтального массива констант, состоящего из пяти элементов:

```
{1;0;1;0;1}
```

Представленная ниже формула использует функцию СУММ, аргументом которой является предыдущий массив констант. Формула возвращает значение суммы элементов массива, равное 3. Отметим, что эта формула использует массив в качестве аргумента, но при этом не является формулой массива. Поэтому не нужно использовать клавиши <Ctrl+Shift+Enter> для ввода формулы:

```
=СУММ({1;0;1;0;1})
```



В случае непосредственного задания массива (как показано выше) необходимо заключать элементы массива в фигурные скобки. С другой стороны, при вводе формулы массива фигурные скобки не ставятся.

Пока, возможно, трудно заметить, какие-либо преимущества использования массивов констант. Действительно, представленная далее формула возвращает точно такое же значение, как и предыдущая:

```
=СУММ(1;0;1;0;1)
```

Очевидные преимущества становятся ясны при дальнейшем рассмотрении.

Следующая формула использует два массива констант:

```
=СУММ({1;2;3;4}*{5;6;7;8})
```

Эта формула создает в памяти новый массив, который состоит из произведения соответствующих элементов двух других массивов. Этот новый массив имеет вид:

```
{5;12;21;32}
```

Созданный массив используется впоследствии в качестве аргумента для функции СУММ, которая возвращает результат вычислений (70). Формула эквивалентна следующей формуле, которая не использует массивы:

```
=СУММ(1*5;2*6;3*7;4*8)
```

Можно составлять формулы массивов для разных типов массивов. Например, можно составить формулу, в которой будет одновременно использоваться массив констант и массив, элементы которого хранятся в диапазоне ячеек. Для подсчета суммы значений, которые являются произведением элементов массива в диапазоне A1 : D1 и соответствующих элементов массива констант, применяется следующая формула:

```
=СУММ(A1:D1*{1;2;3;4})
```

Эта формула эквивалентна следующей:

```
=СУММ(A1*1;B1*2;C1*3;D1*4)
```

Элементы массива констант

Массивы констант могут состоять из чисел, текста, логических значений (например, ИСТИНА, ЛОЖЬ) и даже значений ошибок, таких как #Н/Д. Числа в массиве могут быть целыми, с десятичной точкой или экспоненциальными. Текст в массиве констант заключается в двойные кавычки (например, "Вторник"). Массив констант может состоять из элементов разного типа, как показано в следующем примере:

```
{1;2;3;ИСТИНА;ЛОЖЬ;ИСТИНА;"Вася";"Петя"}
```

Массивы констант не могут включать в себя формулы, функции или другие массивы. Числовые значения не могут включать символы доллара, точки с запятой (запяты), круглые скобки или символ процента. Например, ниже приведен неверный массив констант:

```
{КОРЕНЬ(32);$56,32;12,5%}
```

Понятие размерности массива

Как отмечалось раньше, массивы могут быть одномерными или двумерными. Ориентация одномерного массива может быть горизонтальной или вертикальной.

Одномерные горизонтальные массивы

Элементы в одномерном горизонтальном массиве разделяются точкой с запятой (запятыми в английской версии программы). Ниже приведен пример одномерного горизонтального массива констант:

```
{1;2;3;4;5}
```

Для отображения этого массива в диапазоне необходимо пять последовательных ячеек в строке. Чтобы ввести этот массив в диапазон, выделите диапазон ячеек, состоящий из одной строки и пяти столбцов. Затем введите:

```
={1;2;3;4;5}
```

и нажмите <Ctrl+Shift+Enter>.

Если ввести этот массив в горизонтальный диапазон, состоящий более чем из пяти ячеек, то лишние ячейки будут содержать #Н/Д (что указывает на недопустимые значения). Если

ввести этот массив в вертикальный диапазон ячеек, то в каждой ячейке диапазона появится первый элемент (1) массива.

Ниже приведен пример другого горизонтального массива. Он состоит из семи элементов и создает текстовую последовательность:

```
{ "Понедельник"; "Вторник"; "Среда"; "Четверг"; "Пятница"; "Суббота"; "Воскресенье" }
```

Для ввода этого массива выделите семь элементов в строке и введите следующий текст:

```
= { "Понедельник"; "Вторник"; "Среда"; "Четверг"; "Пятница"; "Суббота"; "Воскресенье" }
```

Нажмите <Ctrl+Shift+Enter>.

Одномерные вертикальные массивы

Элементы в одномерном вертикальном массиве разделяются двоеточием (точкой с запятой в английской версии программы). Ниже приведен пример вертикального массива констант, состоящего из шести элементов:

```
{ 10; 20; 30; 40; 50; 60 }
```

Для отображения этого массива в диапазоне необходимо шесть ячеек в столбце. Чтобы ввести этот массив в диапазон, выделите диапазон ячеек, состоящий из шести строк и одного столбца. Затем введите:

```
= { 10; 20; 30; 40; 50; 60 }
```

и нажмите <Ctrl+Shift+Enter>.

Ниже приведен пример другого вертикального массива, состоящего из четырех элементов:

```
{ "Иванов"; "Петров"; "Сидоров"; "Итого" }
```

Двумерные массивы

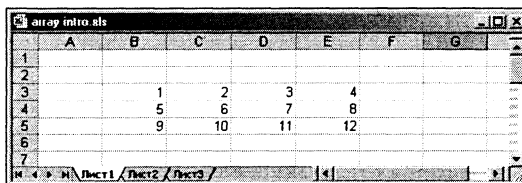
В двумерных массивах используются точки с запятыми (запятые в английской версии программы) для разделения горизонтальных элементов и двоеточия (точки с запятой в английской версии программы) для разделения вертикальных элементов. Ниже приведен пример массива констант размерностью 3×4:

```
{ 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12 }
```

Для отображения этого массива в диапазоне требуется двенадцать ячеек. Чтобы ввести этот массив в диапазон, выделите диапазон ячеек, состоящий из трех строк и четырех столбцов. Затем введите следующую формулу, после чего нажмите <Ctrl+Shift+Enter>:

```
= { 1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11; 12 }
```

Рис. 14.2 демонстрирует, как будет выглядеть этот массив, если его ввести в диапазон ячеек (в данном случае в диапазон В3:Е5).



	A	B	C	D	E	F	G
1							
2							
3			1	2	3	4	
4			5	6	7	8	
5			9	10	11	12	
6							
7							

Рис. 14.2. Массив 3×4, введенный в диапазон ячеек

Если ввести массив в диапазон, состоящий из большего числа ячеек, чем сам массив, Excel отобразит #Н/Д в лишних ячейках. На рис. 14.3 показано, что получится, если ввести массив 3×4 в диапазон ячеек 10×5.

	A	B	C	D	E	F	G	H
1								
2								
3			1	2	3	4	#Н/Д	
4			5	6	7	8	#Н/Д	
5			9	10	11	12	#Н/Д	
6		#Н/Д	#Н/Д	#Н/Д	#Н/Д	#Н/Д		
7		#Н/Д	#Н/Д	#Н/Д	#Н/Д	#Н/Д		
8		#Н/Д	#Н/Д	#Н/Д	#Н/Д	#Н/Д		
9		#Н/Д	#Н/Д	#Н/Д	#Н/Д	#Н/Д		
10		#Н/Д	#Н/Д	#Н/Д	#Н/Д	#Н/Д		
11		#Н/Д	#Н/Д	#Н/Д	#Н/Д	#Н/Д		
12								
13								

Рис. 14.3. Массив 3×4, введенный в диапазон ячеек 10×5

Каждая строчка двумерного массива должна содержать одинаковое количество элементов. Например, представленный ниже массив недопустимый, потому что его третья строка состоит только из трех элементов:

{1; 2; 3; 4; 5; 6; 7; 8; 9; 10; 11}

Excel не даст возможности ввести формулу, в которой присутствует недопустимый формат массива.

Присвоение имени массивам констант

Excel позволяет создать массив констант, присвоить ему имя и затем использовать это имя в формулах. Фактически присвоение имени массиву заключается в присвоении имени формуле.



В главе 3 объясняется понятие имени и описывается методика присвоения имени формуле.

На рис. 14.4 показано, как присвоить имя массиву, используя диалоговое окно Присвоение имени.

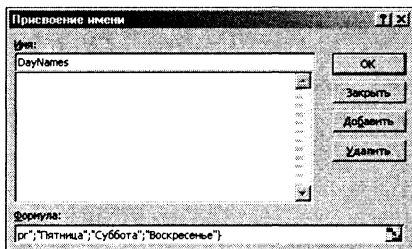


Рис. 14.4. Присвоение имени массиву констант

Массиву присвоено имя *DayNames*, которое соответствует следующему массиву констант:

{ "Понедельник" ; "Вторник" ; "Среда" ;
"Четверг" ; "Пятница" ; "Суббота" ;
"Воскресенье" }

Отметим, что в диалоговом окне Присвоение имени для задания массива необходимо использовать знак равенства (=). В противном случае массив будет воспринят как текстовая строка. Также необходимо самостоятельно взять элементы массива в фигурные скобки. Excel в этом случае не вводит скобки автоматически.

После присвоения имени массиву, его имя можно использовать в формулах. На рис. 14.5 представлен рабочий лист, в котором находится формула, вводящая массив в диапазон A1 : G1. Она имеет следующий вид:

{=DayNames}

	A	B	C	D	E	F	G	H
1	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье	
2								
3								
4								
5								

Рис. 14.5. Использование имени массива в формуле массива

Так как элементы массива разделяют точки с запятой, ориентация массива будет горизонтальной. Для создания вертикальных массивов используются двоеточия. Также можно воспользоваться функцией транспонирования для того, что бы вставить горизонтальный массив в вертикальный диапазон ячеек (см. раздел “Транспонирование массива” далее в этой главе). Для этого введите в диапазон семи вертикальных ячеек следующую формулу массива:

```
{=ТРАНСП(ДейNames)}
```

При использовании функции Excel ИНДЕКС можно получить доступ к отдельному элементу массива. Например, ниже приведенная формула возвращает значение “Четверг”, четвертого элемента массива *DayNames*:

```
=ИНДЕКС(ДейNames; 4)
```

Работа с формулами массива

В этом разделе обсуждаются вопросы выделения ячеек, содержащих массивы, ввод и редактирование формул массива. Эти процедуры немного отличаются от методов управления обычными диапазонами ячеек и формулами.

Ввод формулы массива

Для того чтобы Excel знал, какая вводится формула (массива или обычная), необходимо при вводе формулы массива в диапазон ячеек придерживаться определенных правил. Обычная формула вводится в ячейку после нажатия <Enter>. Формула массива — после нажатия <Ctrl+Shift+Enter>.

Формулу массива легко отличить, в строке формул она заключается в фигурные скобки. Например, следующая формула является формулой массива:

```
{СУММ(ДЛСТР(A1:A5))}
```

Не вводите фигурные скобки при создании формулы массива. Excel сам их вставит при нажатии <Ctrl+Shift+Enter>. Если результат вычисления формулы массива состоит более чем из одного значения, необходимо вначале выделить весь диапазон ячеек, в который помещается результат. И только потом приступайте к вводу формулы. Если этого не сделать, отобразится только первый элемент возвращаемых значений.

Выделение диапазона формулы массива

Можно выделить диапазон ячеек, в котором находится формула массива, используя обычную процедуру выделения ячеек. Либо воспользоваться следующими методами:

- ◆ Активизировать любую ячейку в диапазоне формулы массива. Выбрать Правка⇒Перейти или нажать клавишу <F5>. Щелкните на кнопке Выделить и затем выставьте переключатель Текущий массив. Щелкните на кнопке ОК для закрытия диалогового окна.
- ◆ Активизировать любую ячейку в диапазоне формулы массива и нажать <Ctrl+/> для выделения всего массива.

Редактирование формулы массива

Если формула массива находится не в одной ячейке, то необходимо редактировать весь диапазон. Важно помнить, что нельзя изменять содержимое только одного элемента формулы массива. Если попытаться это сделать, Excel выдаст предупреждение, показанное на рис. 14.6.

Существует ряд правил, которых нужно придерживаться при работе с формулой массива, расположенной в нескольких ячейках. Если попытаться не выполнить определенное правило, Excel сразу даст знать об этом.

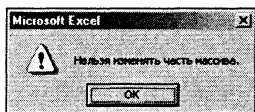


Рис. 14.6. Предупреждающее сообщение Excel напоминает о невозможности редактирования только одной ячейки формулы массива

- ◆ Нельзя изменять содержимое любой отдельной ячейки, которая входит в формулу массива.
- ◆ Нельзя перемещать ячейки, которые входят в формулу массива (но можно переместить всю формулу массива).
- ◆ Нельзя удалять ячейки, которые входят в формулу массива (но можно удалить весь массив).
- ◆ Нельзя вставлять новые элементы в диапазон массива. Это правило распространяется и на вставку строк или столбцов, которые добавляют новые ячейки в диапазон массива.

Для редактирования формулы массива выделите все ячейки диапазона массива. Перейдите в строку формул, щелкнув в ней или нажав клавишу <F2>. Excel удалит фигурные скобки вокруг формулы. Отредактируйте формулу и нажмите <Ctrl+Shift+Enter> для сохранения внесенных изменений. Внесенные изменения отобразятся во всех ячейках массива.



Если после редактирования формулы массива по ошибке нажать <Ctrl+Enter> (вместо <Ctrl+Shift+Enter>), формула введется в каждую выделенную ячейку, но это больше не будет формула массива.

Хотя и нельзя изменить отдельную ячейку, входящую в состав формулы массива, можно применить к ней индивидуальное форматирование.

Расширение или сокращение формулы массива

Часто возникает необходимость увеличить или уменьшить количество ячеек в формуле массива. Для этого целесообразно проделать следующие действия:

1. Выделите весь диапазон, содержащий формулу массива.
2. Нажмите <F2> для перехода в режим редактирования.
3. Нажмите <Ctrl+Enter>. Это действие вводит идентичную формулу (не формулу массива) в каждую выделенную ячейку.

4. Измените диапазон выделения, добавив или исключив ячейки.
5. Нажмите <F2>.
6. Нажмите <Ctrl+Shift+Enter>.

Формулы массива: взгляд изнутри

Если проанализировать предыдущее содержимое этой главы, можно прийти к выводу о существовании весомых преимуществ формул массивов. Конечно, главным преимуществом использования формул массива является возможность использования ранее недоступных методов вычислений. Преимущества использования массивов очевидны, однако есть и существенные недостатки.

Формулы массива являются одной из самых мало понимаемых возможностей Excel. Естественно, желательно избегать применения формул массива, если предполагается совместное редактирование рабочей книги. Многие пользователи Excel при столкновении с формулами массива чувствуют себя совершенно беспомощными.

Нужно учитывать, что очень легко забыть нажать <Ctrl+Shift+Enter> для ввода формулы массива. При редактировании существующего массива необходимо всякий раз использовать эту комбинацию клавиш после завершения редактирования. Помимо логических ошибок, выше означенная ошибка — наиболее распространенная среди пользователей. Если при редактировании формулы массива по ошибке нажата клавиша <Enter>, необходимо нажать <F2>, т.е. вернуться обратно в режим редактирования, и затем нажать <Ctrl+Shift+Enter>.

Другая потенциальная проблема при управлении формулами массивов — замедление пересчета рабочего листа. Особенно эта проблема сказывается при использовании очень больших массивов. Для ее разрешения можно использовать более современные компьютеры. К тому же использование формул массивов почти всегда быстрее, чем использование пользовательских VBA-функций.

Использование формул массива, возвращающих несколько значений

Этот раздел содержит примеры, демонстрирующие дополнительные возможности формул массива, возвращающих несколько значений (формулы массива, которые вводятся в диапазон ячеек). Эти формулы массива открывают новые возможности: создание массивов из значений, использование операторов, функций, транспонирование массивов и генерирование последовательных чисел.

Создание массива из значений в диапазоне

Приведенная ниже формула создает массив из значений в ячейках. Рис. 14.7

представляет рабочий лист с данными, введенными в Диапазон A1:C4. Диапазон D8:F11 содержит формулу массива:

```
{=A1:C4}
```

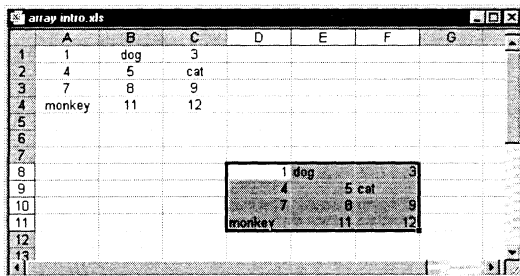


Рис. 14.7. Создание массива из диапазона

Массив в диапазоне D8:F11 связан с диапазоном A1:C4. Внесение изменений в значения диапазона A1:C4 приведет к соответствующим изменениям в ячейках D8:F11.

Создание массива констант из значений диапазона

В предыдущем примере формула массива в D8:F11 связана с диапазоном ячеек A1:C4. Существует возможность разорвать эту связь и создать массив констант, состоящий из значений в диапазоне A1:C4.

Для этого выделите ячейки, которые включают формулу массива (диапазон D8:F11 в этом примере). Нажмите <F2> для запуска режима редактирования. Нажмите <F9> для преобразования ссылок на ячейки в значения. Нажмите <Ctrl+Shift+Enter> для ввода формулы массива констант. Массив констант будет иметь вид:

```
{1;"dog";3;4;5;"cat";"monkey";11;12}
```

На рис. 14.8 показано, как выглядит после этого строка формул.

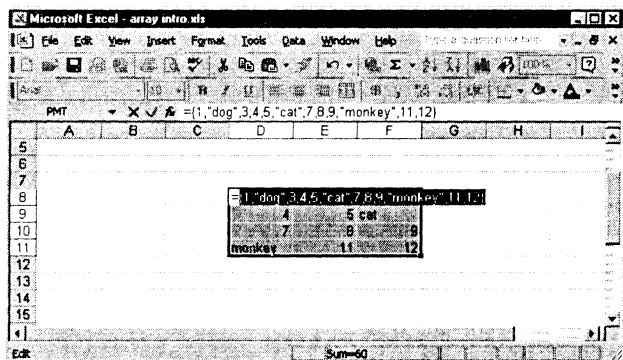


Рис. 14.8. После нажатия клавиши <F9> строка формул отображает массив констант

Выполнение действий в массиве

До настоящего времени большинство примеров в этой главе были посвящены простому вводу массивов в диапазон. Представленная ниже формула массива создает прямоугольный массив, каждый элемент которого получается путем умножения на 2:

```
={1;2;3;4;5;6;7;8;9;10;11;12}*2}
```

На рис. 14.9 показан результат ввода этой формулы в диапазон.

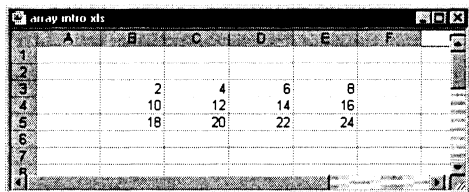


Рис. 14.9. Выполнение математических операций в массиве

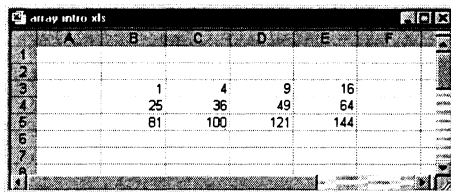


Рис. 14.10. Умножение каждого элемента массива на самого себя

Следующая формула массива умножает каждый элемент массива на самого себя (результат ввода этой формулы в диапазон ячеек представлен на рис. 14.10):

```
{={1;2;3;4;5;6;7;8;9;10;11;12}*{1;2;3;4;5;6;7;8;9;10;11;12}}
```

Следующая формула массива приводит к тому же результату, но более простым способом:

```
{={1;2;3;4;5;6;7;8;9;10;11;12}^2}
```

Если массив хранится в диапазоне ячеек (в данном случае A1:C4), для возведения в квадрат каждого элемента этого массива можно воспользоваться следующей формулой массива:

```
{=A1:C4^2}
```

Использование функций в операциях с массивами

Естественно, есть возможность использовать функции при работе с массивами. Например, можно создать формулу массива, которая вычисляет квадратный корень из каждого из десяти элементов массива констант:

```
{=КОРЕНЬ({1:2:3:4:5:6:7:8:9:10})}
```

В случае, когда массив хранится в диапазоне (в данном случае A1:A10), можно вычислить квадратный корень каждого элемента, входящего в массив, используя следующую формулу массива:

```
{=КОРЕНЬ(A1:A10)}
```

Транспонирование массива

При транспонировании в действительности происходит преобразование строк в столбцы и столбцов в строки. Другими словами можно преобразовать горизонтальный массив в вертикальный или наоборот. Для транспонирования в Excel используется функция транспонирования массива ТРАНСП.

Рассмотрим следующий одномерный горизонтальный массив:

```
{1;2;3;4;5}
```

Можно ввести данный массив в вертикальный диапазон ячеек, используя функцию ТРАНСП. Для этого выделите диапазон из пяти ячеек, который располагается в пяти строках и одном столбце. Затем введите формулу и нажмите <Ctrl+Shift+Enter>:

```
{=ТРАНСП({1;2;3;4;5})}
```

Горизонтальный массив транспонировался, и его элементы появились в вертикальном диапазоне.

Транспонирование двумерных массивов происходит подобным образом. На рис. 14.11 показан двумерный массив, введенный в диапазон ячеек без транспонирования и с использованием функции ТРАНСП. В диапазоне A1:D3 находится формула массива:

```
{={1;2;3;4;5;6;7;8;9;10;11;12}}
```

Диапазон A6:C9 содержит формулу:

```
{=ТРАНСП({1;2;3;4;5;6;7;8;9;10;11;12})}
```

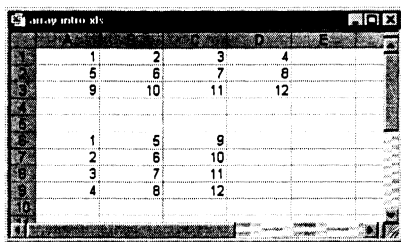


Рис. 14.11. Применение функции ТРАНСП для транспонирования двумерного массива

Как и раньше, функцию транспонирования можно применять и к массиву, хранящемуся в диапазоне ячеек. Например, приведенная ниже формула использует массив, хранящийся в A1:C4 (четыре строчки, три столбца). Формула массива вводится в диапазон из трех строк и четырех столбцов:

```
{=ТРАНСП(A1:C4)}
```

Генерирование массива последовательных целых чисел

Как будет показано в главе 15, для решения многих задач полезно генерировать массивы, состоящие из последовательных целых чисел, а затем использовать их в формулах массива. Идеальным инструментом для этого является функция СТРОКА, возвращающая значение номера строки. Представленная формула введена в вертикальный диапазон из 12 ячеек:

```
{СТРОКА(1:12)}
```

Эта формула генерирует массив из 12 элементов, содержащий целые числа от 1 до 12. Выделите диапазон, состоящий из 12 строк и 1 столбца, и введите формулу массива. Диапазон ячеек будет заполнен последовательными целыми числами (рис. 14.12).

Представленная выше формула хороша для создания последовательности чисел, но не идеальна. Чтобы познакомиться с трудностями ее использования, вставьте новую строку над диапазоном. Excel внесет изменения в формулу, и теперь она будет иметь вид:

```
{СТРОКА(2:13)}
```

Формула, которая поначалу возвращала значения целых чисел от 1 до 12, теперь генерирует значения от 2 до 13.

К лучшим результатам приводит формула:

```
{=СТРОКА(ДВССЫЛ("1:12"))}
```

Эта функция использует функцию ДВССЫЛ, которая в качестве аргумента принимает текстовую последовательность. Excel не изменяет ссылку, заключенную в аргумент функции ДВССЫЛ. Поэтому эта формула массива *всегда* возвращает целые числа от 1 до 12.



В главе 15 приведено несколько интересных примеров генерирования последовательных целых чисел.

Использование формул массива, возвращающих одно значение

Все примеры предыдущего раздела описывали применение формул массива, возвращающих несколько значений, при этом в диапазон ячеек вводилась одна формула. Но почувствовать настоящую мощь формул массива можно, применяя формулы массива, возвращающие одно значение. Такие примеры рассматриваются в настоящем разделе.

Обычные функции, возвращающие массив

Некоторые стандартные функции Excel используют массивы. Нужно ввести формулу, использующую одну из этих функций, в множество ячеек также, как и формулу массива. К таким функциями относятся: ПРЕДСКАЗ, ЧАСТОТА, РОСТ, ЛИНЕЙН, ЛГРФПРИБЛ, ТЕНДЕНЦИЯ и т.д.

Подсчет количества символов в диапазоне

Предположим, что имеется диапазон ячеек, содержащих текст (рис. 14.13).

При традиционном подсчете количества символов в ячейках вначале создают формулу, подобную приведенной ниже:

=ДЛСТР (A1)

Затем копируют ее вниз для вычисления количества символов в каждой ячейке. Затем, используя формулу СУММ, вычисляют сумму промежуточных значений.

Формула массива делает всю работу сразу, без использования промежуточных формул:

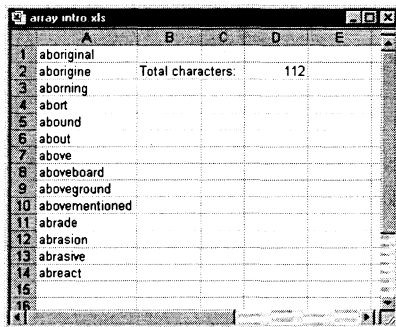
{=СУММ(ДЛСТР(A1:A14))}

Формула массива использует функцию ДЛСТР для создания массива (в памяти), который содержит количество знаков в каждой ячейке диапазона. В нашем случае новый массив имеет вид:

{8;11;5;7;10;7;11;7;11;5;6;10;6;10}

Формулу массива теперь можно представить себе проще:

=СУММ({8;11;5;7;10;7;11;7;11;5;6;10;6;10})



	A	B	C	D	E
1	aboriginal				
2	aborigine	Total characters:		112	
3	aborning				
4	abort				
5	abound				
6	about				
7	above				
8	aboveboard				
9	aboveground				
10	abovementioned				
11	abrade				
12	abrasion				
13	abrasive				
14	abreact				
15					
16					

Рис. 14.13. Подсчет количества символов в ячейках с помощью формулы массива

Сложение трех наименьших элементов диапазона

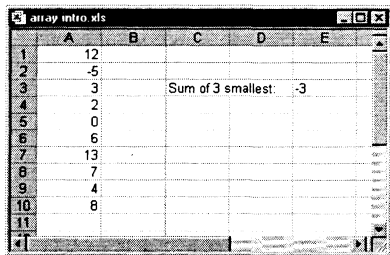
Представленная ниже формула возвращает сумму трех наименьших значений в диапазоне с именем *Data*:

{=СУММ(НАИМЕНЬШИЙ(Data;{1;2;3}))}

В этой формуле используется массив констант в качестве второго аргумента функции НАИМЕНЬШИЙ. Функция генерирует новый массив, который состоит из трех наименьших значений диапазона. Затем массив обрабатывается функцией СУММ, которая возвращает сумму значений массива.

На рис. 14.14 показан пример массива *Data*, находящийся в диапазоне A1:A10.

Функция НАИМЕНЬШИЙ вычисляет трижды наименьшее значение массива при разных значениях второго аргумента. В первый раз в качестве второго аргумента выступает число 1, поэтому вычисляется первое наименьшее значение. Функция НАИМЕНЬШИЙ возвращает значение -5. Во второй раз вторым аргументом является число 2, и



	A	B	C	D	E
1	12				
2	-5				
3	3				
4	2				
5	0				
6	6				
7	13				
8	7				
9	4				
10	8				
11					

Рис. 14.14. Формула массива возвращает сумму трех наименьших значений массива

функция **НАИМЕНЬШИЙ** возвращает второе наименьшее значение — 0. В третий раз второй аргумент — число 3, а возвращаемое значение — 2 (третье наименьшее).

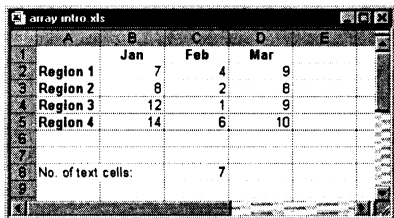
Поэтому функции **СУММ** передается массив:

{-5; 0; 2}

Формула возвращает сумму элементов этого массива (-3).

Подсчет количества текстовых ячеек в диапазоне

Функция **ЕСЛИ** используется в примере этого раздела для того, чтобы проверить, является ли содержимое ячейки текстом. В результате формируется новый массив, состоящий из нулей и единиц в зависимости от того, является ячейка текстовой или нет. Следует отметить, что новый массив имеет такой же размер и ориентацию, что и исходный. Затем новый массив передается функции **СУММ**, которая возвращает сумму элементов в массиве. Таким образом, формула предоставляет результат — количество текстовых ячеек в диапазоне:



	Jan	Feb	Mar	
Region 1	7	4	9	
Region 2	8	2	8	
Region 3	12	1	9	
Region 4	14	6	10	
No. of text cells:		7		

Рис. 14.15. Формула массива возвращает количество ячеек с текстом в указанном диапазоне

{=СУММ(ЕСЛИ(ЕТЕКСТ(A1:D5); 1; 0))}

Результат применения описанной выше формулы находится в ячейке C8 (рис. 14.15), а массив, созданный

функцией **ЕСЛИ**, имеет вид:

{0; 1; 1; 1; 1; 0; 0; 0; 0; 1; 0; 0; 0; 0; 1; 0; 0; 0; 1; 0; 0; 0}



Такой тип формулы массива (а именно, функция **ЕСЛИ**, вложенная в функцию **СУММ**), очень полезна в любых расчетах. Глава 7 содержит дополнительные примеры по этому вопросу.

К такому же результату приводит использование формулы:

{=СУММ(ЕТЕКСТ(A1:D5) * 1)}

В этой формуле не используется функция **ЕСЛИ**, вместо этого берутся во внимание следующие равенства:

ИСТИНА * 1 = 1

и

ЛОЖЬ * 1 = 0

Устранение промежуточных формул

Одним из главных достоинств использования формул массива является возможность устранения промежуточных формул в электронной таблице. Это делает рабочие таблицы менее громоздкими и исключает необходимость отображения несущественных вычислений. На рис. 14.16 показана электронная таблица с баллами, набранными на первом и втором этапе тестирования.

Столбец D содержит разницу между баллами в столбцах B и C. Ячейка D17 содержит формулу, вычисляющую среднее значение в столбце D:

=СРЗНАЧ(D2:D15)

При применении формулы массива можно не использовать промежуточные вычисления. Ниже представлена формула, вычисляющая среднюю разницу баллов без использования значений в столбце D:

```
{=СРЗНАЧ (C2:C15-B2:V15) }
```

Как она работает? Формула использует два массива, элементы которых хранятся в двух диапазонах (B2:V15 и C2:V15). Формула создает *новый* массив, состоящий из разности соответствующих элементов исходных массивов. Этот новый массив хранится только в памяти Excel, а не в ячейках электронной таблицы. Функция СРЗНАЧ использует этот новый массив в качестве аргумента. Новый массив состоит из следующих элементов:

```
{11;15;-6;1;19;2;0;7;15;1;8;23;21;-11}
```

Следовательно, формула сводится к следующей:

```
=СРЗНАЧ ( {11;15;-6;1;19;2;0;7;15;1;8;23;21;-11} )
```

Student	Pre-Test	Post-Test	Change
Andy	56	67	11
Beth	59	74	15
Cindy	98	92	-6
Duane	78	79	1
Eddy	81	100	19
Francis	92	94	2
Georgia	100	100	0
Hilda	92	99	7
Isabel	54	69	15
Jack	91	92	1
Kent	80	88	8
Linda	45	68	23
Michelle	71	92	21
Nancy	94	83	-11
Average Change:			7.57

Рис. 14.16. Без использования формулы массива для подсчета средней разницы баллов требуются промежуточные формулы в столбце D

	A	B	C	D	E	F	G
1	Число >	9	Девять	1	Один		
2				2	Два		
3				3	Три		
4				4	Четыре		
5				5	Пять		
6				6	Шесть		
7				7	Семь		
8				8	Восемь		
9				9	Девять		
10				10	Десять		

Рис. 14.17. Таблицу преобразований, находящуюся в D1:E10, можно заменить на массив констант

На основе данных этого примера можно вычислить и другие значения, используя формулы массива.

Например, следующая формула возвращает наибольшую разницу (характеризующее наибольшее улучшение успеваемости), равное 23 и соответствующее результатам тестирования студентки Linda:

```
{МАКС (C2:C15-B2:V15) }
```

Приведенная ниже формула возвращает наименьшую разницу (характеризующую ухудшение успеваемости), равное -11 и соответствующее результатам тестирования студентки Nancy:

```
{МИН (C2:C15-B2:V15) }
```

Использование массива вместо диапазона ссылок

Некоторые функции в качестве аргумента используют диапазон ссылок. Если в формуле используется такая функция, есть возможность заменить этот диапазон ссылок на массив констант. Это полезно, если значения в диапазоне ссылок не изменяются.



Исключение составляют функции работы с базами данных (например, ВДСУММ). Эти функции в качестве обязательного аргумента используют критерий, находящийся в диапазоне ячеек. К сожалению, этот критерий нельзя задать, используя массив констант.

На рис. 14.17 показана электронная таблица, в которой используется таблица преобразования для отображения слова, соответствующего числу. Например, числу 9 в таблице преобразования (диапазон D1:E10) соответствует значение *Девять*. Формула, возвращающая это значение, находится в ячейке C1:

```
=ВПР (B1;D1:E10;2;ЛОЖЬ)
```

Вместо таблицы преобразований можно использовать двумерный массив. Приведенная формула возвращает такой же результат, что и предыдущая, но не требует наличия таблицы преобразований:

```
=ВПР (B1;{1;"Один":2;"Два":3;"Три":4;"Четыре":5;"Пять":6;"Шесть":7;"Семь":8;"Восемь":9;"Девять":10;"Десять"};2;ЛОЖЬ)
```

Резюме

В этой главе введено понятие *массива* — совокупности элементов, находящихся в диапазоне ячеек или памяти Excel. *Формулы массива* оперируют диапазоном значений и возвращают либо одно значение, либо массив значений. В следующей главе продолжается обсуждение вопросов применения формул массива и приведено много полезных примеров.

Глава 15

Магия формул массива

В этой главе ...

- ◆ Работа с формулами массива, возвращающими одно значение
- ◆ Работа с формулами массива, возвращающими несколько значений
- ◆ Резюме

Предыдущая глава была вводной в понятия массива и формулы массива. Она содержала примеры, которые призваны вызвать повышенный интерес к применению массивов. В настоящей главе продолжается повествование о формулах массивов. Она содержит полезные примеры, которые продолжают демонстрировать поразительную эффективность формул массива.

В этой главе многие пользователи найдут самые разнообразные и полезные примеры использования формул массива. Большинство из них можно непосредственно использовать в работе, нужно только в ряде случаев изменить имена диапазонов или ссылки. Во многие примеры легко внести изменения, чтобы они работали именно так, как нужно.

Работа с формулами массива, возвращающими одно значение

Как описывалось в предыдущей главе, формулу массива можно вводить в одну ячейку (а не в диапазон ячеек). Такие формулы массива обрабатывают массивы, хранящиеся либо в диапазоне ячеек, либо в памяти Excel. В этом разделе представлены некоторые дополнительные примеры таких формул массива.

Суммирование в диапазоне, который содержит ошибки

Вероятно, многие пользователи Excel знают, что функция СУММ не работает, если попытаться просуммировать диапазон ячеек, содержащих сообщения об ошибках (например, #ДЕЛ/0! или #Н/Д). Рис. 15.1 иллюстрирует такой случай. Формула СУММ в ячейке С9 возвращает сообщение об ошибке, так как в диапазоне ячеек С2:С8, которые она складывает, содержатся сообщения об ошибках.

Следующая формула массива возвращает сумму значений в диапазоне с именем *Data*, даже если этот диапазон содержит сообщения об ошибке:

```
{=СУММ(ЕСЛИ(ЕОШИБКА(Data);"";Data))}
```

Эта формула создает новый массив, включающий значения из диапазона с именем *Data*, но без сообщений об ошибке. Функция ЕСЛИ заменяет сообщения об ошибке на пустые зна-

	A	B	C	D
1	Total	Number	Per Unit	
2		80	10	8.00
3		120	6	20.00
4		144	12	12.00
5				#ДЕЛ/0!
6				#ДЕЛ/0!
7		100	20	5.00
8		50	5	10.00
9		TOTAL:		#ДЕЛ/0!
10				
11				

Рис. 15.1. Формула массива может складывать значения в диапазоне, содержащем ошибки

чения. Затем функция СУММ складывает элементы этого “отфильтрованного” массива. Такую же методику обработки диапазонов, содержащих сообщения об ошибках, можно применить и для других функций, например МИН и МАКС.



Возникают случаи, когда нужно использовать другие функции вместо ЕОШИБКА. Дело в том, что ЕОШИБКА возвращает значение ИСТИНА при любых значениях ошибки (#Н/Д, #ЗНАЧ!, #ССЫЛКА!, #ДЕЛ/0!, #ЧИСЛО!, #ИМЯ? или #ПУСТО!). Функция ЕОШ возвращает значение ИСТИНА при любых значениях ошибки кроме #Н/Д, а функция ЕНД — только при значениях ошибки #Н/Д.

Подсчет количества ошибок в диапазоне

Представленная ниже формула массива похожа на формулу предыдущего примера, только возвращает количество ошибок в диапазоне с именем *Data*:

```
{=СУММ(ЕСЛИ(ЕОШИБКА(Data);1;0))}
```

Данная формула создает массив, состоящий из единиц (если соответствующая ячейка содержит сообщение об ошибке) и нулей (если в соответствующей ячейке отсутствует сообщение об ошибке).

Можно немного упростить формулу, удалив третий аргумент функции ЕСЛИ. Теперь, если условие не выполняется (ячейка не содержит сообщение об ошибке), функция ЕСЛИ будет возвращать значение ЛОЖЬ. Усовершенствованная формула возвращает такой же результат, что и предыдущая, но не использует третий аргумент функции ЕСЛИ:

```
{=СУММ(ЕСЛИ(ЕОШИБКА(Data);1))}
```

В действительности можно еще более упростить эту формулу:

```
{=СУММ(ЕОШИБКА(Data)*1)}
```

В этой версии формулы берутся во внимание следующие равенства:

ИСТИНА * 1 = 1

и

ЛОЖЬ * 1 = 0

Суммирование при выполнении условия

В ряде случаев возникает необходимость просуммировать значения, которые удовлетворяют одному и более условий. Например, представленная формула массива складывает положительные значения (исключая при этом отрицательные) в диапазоне с именем *Data*:

```
{=СУММ(ЕСЛИ(Data>0;Data))}
```

Функция ЕСЛИ создает массив, состоящий из положительных значений, и значений ЛОЖЬ вместо отрицательных. Этот массив обрабатывается функцией СУММ, которая отбрасывает логические значения ЛОЖЬ и возвращает сумму чисел массива. Массив *Data* может состоять из любого количества строк и столбцов.

Для решения поставленной задачи можно, конечно, использовать функцию СУММЕСЛИ. Следующая формула не является формулой массива и возвращает такой же результат, что и предыдущая формула массива

```
=СУММЕСЛИ(Data;">0")
```

Однако использование функции СУММЕСЛИ при задании сложных условий может приводить к появлению довольно замысловатых формул:

=СУММЕСЛИ(Data; ">0";Data) –СУММЕСЛИ(Data; ">5";Data)

Например, представленная выше формула (не формула массива) складывает значения диапазона с именем *Data*, находящиеся в интервале от 0 до 5. Эта формула работает следующим образом. Вначале складываются значения, большие 0. Потом из полученной суммы вычитается результат сложения значений, больших 5. Такая формула может любого сбить с толку, к тому же ее использование неэффективно.

При применении формулы массива эту же задачу можно решить намного изящнее:

{=СУММ((Data>0)*(Data<=5)*Data)}

Отметим, что эта формула имеет одно ограничение: она будет возвращать значение ошибки, если диапазон с именем *Data* содержит хотя бы одно нечисловое значение.



Вопреки возможным ожиданиям, в формулах массива нельзя использовать функцию И. Хотя, с точки зрения формальной логики, приведенная ниже формула абсолютно верна, она не дает правильного результата:

{=СУММ(ЕСЛИ(И(Data>0;Data<=5);Data))}

Нелогичное поведение логических функций

Функции И и ИЛИ относятся к категории логических функций, которые возвращают значения ИСТИНА или ЛОЖЬ. К сожалению, эти функции нельзя применять в формулах массива.

На представленном ниже рисунке столбцы А и В содержат логические значения. Функция И возвращает значение ИСТИНА, если все ее аргументы ИСТИНА. Столбец С содержит обычные формулы, которые работают так, как они и должны работать. Например, ячейка С3 содержит следующую функцию:

=И(А3;В3)

	A	B	C	D	E
1			Обычные формулы	Массив с оператором И (=И(А3:А6;В3:В6))	Формула массива (=А3:А6*В3:В6)
2	Условие 1	Условие 2			
3	ИСТИНА	ИСТИНА	ИСТИНА	ЛОЖЬ	1
4	ИСТИНА	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	0
5	ЛОЖЬ	ИСТИНА	ЛОЖЬ	ЛОЖЬ	0
6	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	ЛОЖЬ	0

Диапазон D3 : D6 содержит следующую формулу массива:

{=И(А3 : А6 ; В3 : В6)}

Можно ожидать, что формула массива возвратит следующий массив:

{ИСТИНА ; ЛОЖЬ ; ЛОЖЬ ; ЛОЖЬ}

Однако она возвращает только одно значение ЛОЖЬ. Фактически обе функции, как И так и ИЛИ, всегда возвращают одно единственное значение и никогда не возвращают массив. Даже если использовать массив констант, функция И все равно возвратит только одно значение. Например, следующая формула массива не возвращает массив:

{=И({ИСТИНА;ИСТИНА;ЛОЖЬ;ЛОЖЬ};{ИСТИНА;ЛОЖЬ;ИСТИНА;ЛОЖЬ})}

Автору книги не известно, является ли такое поведение функций И и ИЛИ ошибкой или программной особенностью. В любом случае несомненно, что все другие функций работают не так, как эти две.

Столбец E содержит другую формулу массива, которая, как ей и полагается, возвращает массив. Массив создается следующей формулой, состоит из 0 и 1, причем эти значения соотносятся со значениями ИСТИНА и ЛОЖЬ.

```
{=A3:A6*B3:B6}
```

Не возвращает массива только массив формула, в которой применяется функция ИЛИ:

```
{=ИЛИ(A3:A6;B3:B6)}
```

Для получения корректных результатов предпочтительнее использовать следующую формулу массива, которая *выполняет* обработку соответствующих элементов по правилам логического сложения:

```
{=A3:A6+B3:B6}
```

Формулы массива позволяют применять и более сложные условия отбора. Например, следующая формула складывает значения меньше 0 или больше 5:

```
{=СУММ(ЕСЛИ((Data<0)+(Data>5);Data))}
```



Так же как и в случае с функцией И, нельзя использовать функцию ИЛИ в формулах массива. Например, следующая формула возвращает некорректный результат:

```
{=СУММ(ЕСЛИ(ИЛИ(Data<0;Data>5);Data))}
```

Для уяснения обходного пути, необходимого при использовании логических функций в формулах массива, обратитесь к врезке “Нелогичное поведение логических функций”.

Суммирование n-го числа наибольших значений в диапазоне

Следующая формула возвращает значение суммы десяти наибольших значений в диапазоне с именем *Data*:

```
{=СУММ(НАИБОЛЬШИЙ(Data;СТРОКА(ДВССЫЛ("1:10"))))}
```

Функция НАИБОЛЬШИЙ выполняется 10 раз, каждый раз с другим вторым аргументом (1, 2, 3, ... 10). Результаты этих вычислений хранятся в новом массиве, который выступает аргументом для функции суммирования СУММ.

Для суммирования другого количества значений нужно заменить число 10 в аргументе функции ДВССЫЛ на другое значение. Для суммирования *n-го числа наименьших* значений в диапазоне нужно заменить функцию НАИБОЛЬШИЙ функцией НАИМЕНЬШИЙ.

Вычисление среднего без учета нулевых значений

На рис. 15.2 представлен рабочий лист, на котором производится вычисление среднего уровня объема продаж. В ячейку B11 помещена формула:

```
=СРЗНАЧ(B2:B9)
```

Эта формула вычисляет среднее значение в диапазоне B2:B9. Двое служащих из торгового персонала отсутствовали на работе некоторое время. Таким образом, представленная выше формула не точно показывает средний уровень продаж для принимавших участие в анализе продавцов.

	A	B	C	D	E	F
1	Sales Person	Sales				
2	Abner	23,991				
3	Baker	15,092				
4	Charleston	0				
5	Davis	11,893				
6	Ellerman	32,116				
7	Flugelhart	29,089				
8	Galloway	0				
9	Harrison	33,211				
10						
11		18,174	← Average with zeros			
12		24,232	← Average without zeros (array formula)			
13						

Рис. 15.2. Вычисление среднего значения, исключая ячейки, содержащие ноль



Функция СРЗНАЧ не учитывает пустые ячейки, зато принимает в расчет ячейки, содержащие нулевые значения.

Следующая формула массива возвращает среднее значение в диапазоне и при этом исключает ячейки, содержащие нулевые значения:

```
{=СРЗНАЧ(ЕСЛИ(B2:B9<>0;B2:B9))}
```

Эта формула создает новый массив, состоящий исключительно из ненулевых значений. Функция СРЗНАЧ впоследствии использует этот массив в качестве своего аргумента. Тот же результат получается при использовании обычной формулы (не формулы массива):

```
=СУММ(B2:B9)/СЧЕТЕСЛИ(B2:B9;"<>0")
```

Вначале рассчитывается сумма всех значений ячеек в диапазоне. Затем эта сумма делится на количество ненулевых ячеек диапазона, полученное с помощью функции СЧЕТЕСЛИ.

Поиск значения в диапазоне

Для того чтобы определить, присутствует ли определенное значение в диапазоне ячеек, необходимо выбрать команду Правка⇒Найти и произвести поиск значения на рабочем листе. Можно поступить другим способом, и решить эту задачу с помощью формулы массива.

Формула массива в ячейке D1 проверяет имя, введенное в ячейку C1, которой присвоено имя *TheName*. Если имя присутствует в списке имен, формула выводит текст *Found*. В противном случае, она отображает фразу *Not Found*.

Формула в ячейке C1 имеет вид:

```
{=ЕСЛИ(ИЛИ(TheName=NameList);"Found";"Not Found")}
```

Эта формула сравнивает *TheName* с каждым элементом в диапазоне *NameList*. Она создает новый массив, который состоит из логических значений ИСТИНА или ЛОЖЬ. Функция ИЛИ возвращает значение ИСТИНА, если хотя бы одно значение в новом массиве — ИСТИНА. Функция ЕСЛИ использует этот результат для определения возвращаемого сообщения.

Более простой вариант этой формулы представлен ниже. Эта формула возвращает значение ИСТИНА, если имя найдено, в противном случае — ЛОЖЬ.

```
{=ИЛИ(TheName=NameList)}
```

На рис. 15.3 показан рабочий лист со списком имен в диапазоне A3 : E22, которому присвоено имя *NameList*.

1	A	B	C	D	E
2	Enter a Name ->		Michelle	I found	
3	Al	Daniel	Harold	Lyle	Richard
4	Allen	Dave	Ian	Maggie	Rick
5	Andrew	David	Jack	Margaret	Robert
6	Anthony	Dennis	James	Manlyn	Rod
7	Arthur	Don	Jan	Mark	Roger
8	Barbara	Donald	Jeff	Marvin	Ronald
9	Bernard	Doug	Jeffrey	Mary	Russ
10	Beth	Douglas	Jerry	Matt	Sandra
11	Bill	Ed	Jim	Mel	Scott
12	Bob	Edward	Joe	Merle	Simon
13	Brian	Eric	John	Michael	Stacy
14	Bruce	Fran	Joseph	Michelle	Stephen
15	Cark	Frank	Karl	Mike	Steven
16	Carl	Fred	Kathy	Norman	Stuart
17	Charles	Gary	Keith	Patrick	Susan
18	Chris	George	Kenneth	Paul	Terry
19	Chuck	Glenn	Kevin	Peter	Thomas
20	Clark	Gordon	Larry	Phillip	Timothy
21	Curt	Greg	Leonard	Ray	Vincent
22	Dan	Gregory	Louise	Rebecca	William
23					

Рис. 15.3. Использование формулы массива для определения наличия значения в диапазоне

Подсчет отличающихся значений в двух диапазонах

Представленная ниже формула массива сравнивает соответствующие значения в двух диапазонах (имеющих имена *MyData* и *YourData*), и возвращает количество различных элементов в этих двух диапазонах. Если содержимое двух диапазонов идентично, формула возвращает значение 0.

```
{=СУММ(ЕСЛИ(MyData=YourData;0;1))}
```

Два диапазона должны иметь одинаковый размер и ориентацию.

Эта формула создает новый массив такого же размера, как и сравниваемые диапазоны. Функция ЕСЛИ заполняет этот новый массив нулями и единицами (ноль — если есть различие, единица — если нет различия у соответствующих ячеек). Функция СУММ возвращает сумму значений этого массива.

Тот же результат можно получить, применив более простую формулу:

```
{=СУММ(1*(MyData<>YourData))}
```

Эта версия формулы базируется на следующих равенствах:

ИСТИНА * 1 = 1

и

ЛОЖЬ * 1 = 0

Возвращение расположения максимального значения диапазона

Для определения номера строки, в которой находится максимальное значение одномерного вертикального массива с именем *Data*, можно воспользоваться формулой:

```
{=МИН(ЕСЛИ(Data=МАКС(Data);СТРОКА(Data);""))}
```

Функция ЕСЛИ создает новый массив, соотносящийся с диапазоном с именем *Data*. Если соответствующая ячейка содержит максимальное значение диапазона *Data*, то новый массив содержит номер ее строки. В противном случае, он содержит пустую текстовую строку. Функция МИН использует этот массив как второй аргумент и возвращает минимальное значение, которое соответствует номеру строки, в которой находится максимальное значение диапазона *Data*.

Если диапазон *Data* содержит более чем одну ячейку с максимальным значением, то возвращается номер первой ячейки с максимальным значением.

Представленная ниже формула массива подобна предыдущей, но она возвращает действительный адрес ячейки с максимальным значением в диапазоне *Data*. Формула использует функцию АДРЕС, которая имеет два аргумента: номер строки и номер столбца.

```
{=АДРЕС(МИН(ЕСЛИ(Data=МАКС(Data);СТРОКА(Data);""));СТОЛБЕЦ(Data))}
```

Поиск номера строки, в которой находится *n*-ое значение, совпадающее с заданным

Приведенная ниже формула ищет значения в одномерном вертикальном диапазоне с именем *Data*, совпадающие со значением в ячейке *Value*, и возвращает номер строки *n*-го совпадающего элемента.

```
{=НАИМЕНЬШИЙ(ЕСЛИ(Data=Value;СТРОКА(Data);"");n)}
```

Функция ЕСЛИ создает новый массив, состоящий из номеров строк, в которых значения из диапазона *Data* совпадают со значением *Value*. Если значение из *Data* не совпадает со значением *Value*, то новый массив содержит пустую текстовую строку. Функция НАИМЕНЬШИЙ обрабатывает этот новый массив и возвращает *n*-ый наименьший номер столбца.

Если *n* превышает количество значений в диапазоне *Data*, равных *Value*, или значение *Value* не найдено, формула выдает сообщение #ЧИСЛО!

Возвращение самого длинного текста в диапазоне

Следующая формула отображает содержимое ячейки диапазона *Data*, в которой содержится наибольшее количество символов. Если в нескольких ячейках содержится самый длинный текст, то возвращается содержимое первой такой ячейки.

```
{=ИНДЕКС(Data;ПОИСКПОЗ(МАКС(ДЛСТР(Data));ДЛСТР(Data);ЛОЖЬ);1)}
```

Эта формула работает с двумя массивами, каждый из которых представляет длину каждого элемента из диапазона *Data*. Функция МАКС находит наибольшее значение, которое соответствует элементу с самым длинным текстом. Функция ПОИСКПОЗ вычисляет смещение ячейки с максимальной длиной. Функция ИНДЕКС возвращает содержимое ячейки, в которой находится текст с наибольшим количеством символов. Эта формула возвращает правильный результат только в том случае, если диапазон *Data* состоит из одного столбца.

Определение допустимых значений диапазона

Иногда возникает следующая задача: необходимо проконтролировать содержимое списка на предмет допустимости значений в нем. Например, список всех допустимых чисел хранится в диапазоне с именем *Master*. При импортировании списка чисел в диапазон с именем *MyList* требуется убедиться, что все числа допустимы, путем сравнения элементов в импортируемом списке с элементами в списке *Master*.

Приведенная ниже формула массива возвращает значение ИСТИНА, если каждый элемент диапазона с именем *MyList* найден в диапазоне с именем *Master*. Оба этих диапазона должны состоять из одного столбца в независимости от числа строк.

```
{=ЕНД(ПОИСКПОЗ(ИСТИНА;ЕНД(ПОИСКПОЗ(MyList;Master;0));0))}
```

Также можно определить количество недопустимых значений. Иными словами, если нужно подсчитать количество элементов в *MyList*, которые отсутствуют в *Master*, можно воспользоваться следующей формулой:

```
{=СУММ(1*ЕНД(ПОИСКПОЗ(MyList;Master;0)))}
```

А для определения первого недопустимого элемента в *MyList* можно успешно использовать формулу:

```
{=ИНДЕКС(MyList;ПОИСКПОЗ(ИСТИНА;ЕНД(ПОИСКПОЗ(MyList;Master;0));0))}
```

Определение суммы цифр числа

Для вычисления суммы цифр в положительном целом числе, которое хранится в ячейке A1, можно применить формулу:

```
{=СУММ(ПСТР(A1;СТРОКА(ДВССЫЛ("1:"&ДЛСТР(A1))));1)*1)}
```

Например, если ячейка A1 содержит значение 409, приведенная выше формула возвратит значение 13 (4+0+9=13).

А чтобы понять, как работает эта формула, рассмотрим функцию СТРОКА в формуле:

```
{=СТРОКА(ДВССЫЛ("1:"&ДЛСТР(A1)))}
```

Эта формула возвращает массив последовательных целых чисел, начинающийся с единицы и заканчивающийся количеством цифр в числе, находящемся в ячейке A1. Например, если ячейка A1 содержит значение 409, функция ДЛСТР возвратит значение 3, функцией СТРОКА будет сгенерирован массив:

```
{1;2;3}
```



Подробная информация об использовании функции ДВССЫЛ приведена в главе 14.

Затем полученный массив используется в качестве второго аргумента функции ПСТР. Если в формулу с ПСТР подставить полученные в описанном примере значения, формула упростится и будет иметь вид:

```
ПСТР(409;{1;2;3};1)*1
```

Эта формула генерирует массив, состоящий из трех элементов:

```
{4;0;9}
```

Теперь общая формула с функцией СУММ еще более упростится и будет иметь вид:
 $=\text{СУММ}(\{4;0;9\})$

Результат вычислений равен 13.



Функция ПСТР возвращает текстовую строку. Для преобразования текста в число элементы массива, находящиеся в аргументе функции ПСТР, умножаются на 1. Альтернативный путь — использование функции ЗНАЧЕН, которая также преобразует строку текста, отображающую число, в числовое значение.

Заметьте, что описанная выше формула не работает с отрицательными значениями, так как знак минус не является цифрой. Ниже приводится формула, которая исправляет этот недостаток. Для этого применяется функция ABS, которая возвращает абсолютное значение числа. На рис. 15.4 представлен рабочий лист, содержащий в ячейке B2 следующую формулу:
 $\{=\text{СУММ}(\text{ЗНАЧЕН}(\text{ПСТР}(\text{ABS}(A2)); \text{СТРОКА}(\text{ДВССЫЛ}("1:" \& \text{ДЛСТР}(\text{ABS}(A2))))); 1))\}$

	A	B	C
1	Number	Sum of Digits	
2	132	6	
3	9	9	
4	111111	6	
5	980991	36	
6	409	13	
7		0	
8	12	3	
9			
10	123	6	
11			
12			

Рис. 15.4. Формула массива вычисляет сумму цифр любого целого числа

	A	B	C	D	E
1	Description	Quantity	Unit Price	Discount	Total
2	Widgets	6	\$11.69	5.23%	\$66.47
3	Sprockets	8	\$9.74	5.23%	\$73.84
4	Snapfolytes	3	\$9.85	5.23%	\$28.00
5	GRAND TOTAL				\$168.32
6					
7					
8					

Рис. 15.5. Применение формулы массива устраняет ошибку округления

Формула в ячейке B2 была скопирована в другие ячейки столбца B для вычисления суммы чисел, находящихся в столбце A.

Суммирование округленных значений

На рис. 15.5 представлен рабочий лист, который иллюстрирует проблему всех электронных таблиц — ошибку округления.

Как видно из рисунка, общая сумма в ячейке E5 не является точной суммой чисел в столбце E. Значения в столбце E используют числовой формат, в котором отображается две цифры после запятой. Действительные значения имеют больше знаков после запятой, которые не отображаются. Таким образом, в ячейках находятся значения, округленные до двух десятичных разрядов после запятой. Результирующее влияние погрешностей округления может привести к общему неточному результату. Общее точное значение составляет \$168,320997, отображаемое — \$168,32.

Формула массива создает новый массив, который состоит из округленных значений в столбце E:

$\{=\text{СУММ}(\text{ОКРУГЛ}(E2:E4; 2))\}$

Эта формула возвращает правильную величину суммы округленных значений — \$168,31.

Для устранения таких ошибок округления можно ввести в ячейки столбца E уже округленные значения с помощи функции ОКРУГЛ. Этот подход не требует создания формулы массива.



В главе 10 содержится подробная информация о функциях Excel, осуществляющих округление.

Суммирование каждого n -го значения в массиве

Предположим, что существует диапазон значений и необходимо высчитать сумму каждого третьего значения в нем: первого, четвертого, седьмого и т.д. Один из путей решения этой задачи — внести адреса ячеек в формулу суммирования. Однако лучшим решением является использование формулы массива.

Обратимся к данным рис. 15.6. Значения хранятся в диапазоне B2 : B20 с именем *Data*, а значение n задается в ячейке D6, названной n .

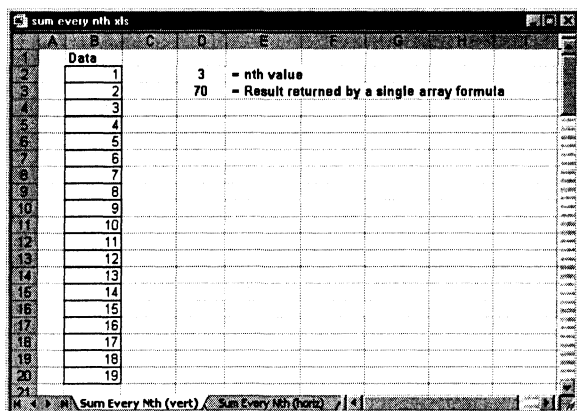


Рис. 15.6. Формула массива возвращает сумму каждого n -го значения в диапазоне

Для суммирования каждого n -го элемента диапазона воспользуемся формулой:

```
{=СУММ(ЕСЛИ(ОСТАТ(СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data)))-1;n)=0;Data;""))}
```

Эта формула создает массив последовательных целых чисел, а функция ОСТАТ использует его в качестве первого аргумента. Второй аргумент функции ОСТАТ — значение n . Функция ОСТАТ создает другой массив, состоящий из остатков деления каждого последовательного целого числа на n . При равенстве элемента этого массива 0 (это значит, что деление прошло без остатка) происходит суммирование соответствующих значений в диапазоне *Data*.

Эта формула выдает сообщение об ошибке, если принять значение n равным 0. Улучшенная формула массива включает функцию ЕСЛИ для успешной работы при $n=0$:

```
{=ЕСЛИ(n=0;0;СУММ(ЕСЛИ(ОСТАТ(СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(data)))-1;n)=0;data;""))}
```

Эта формула выполняется только в ситуации, когда диапазон *Data* состоит из одного столбца значений. Она не работает, если диапазон состоит из более чем одного столбца или даже одной строки.

Для создания формулы, работающей с горизонтальным диапазоном, необходимо транспонировать массив целых чисел, генерируемых функцией СТРОКА. Модифицированная формула массива, предназначенная для обработки горизонтального диапазона *Data*, имеет вид:

```
{=ЕСЛИ(n=0;0;СУММ(ЕСЛИ(ОСТАТ(ТРАНСП(СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))))-1;n)=0;Data;""))}
```

Исключение нечисловых символов из текстовой строки

Приведенная ниже формула проводит извлечение числа из текстовой строки. Например, при применении формулы к текстовой строке, содержащей текст ABC145Z, возвращается значение 145.

```
{=ПСТР(A1;ПОИСКПОЗ(0;(ЕОШИБКА(ПСТР(A1;СТРОКА(ДВССЫЛ("1:"&ДЛСТР(A1)))));1)*1)*1);0);ДЛСТР(A1)-СУММ((ЕОШИБКА(ПСТР(A1;СТРОКА(ДВССЫЛ("1:"&ДЛСТР(A1)))));1)*1)*1))}
```

Эта формула работает только с непрерывными цифровыми последовательностями. Например, формула не даст правильного ответа при применении к текстовой последовательности X45Z99.

Определение ближайшего значения в диапазоне

Пусть ячейка с произвольным числом имеет название *Target*. Для определения ближайшего к нему числа из массива *Data* воспользуемся формулой массива:

```
{=ИНДЕКС(Data;ПОИСКПОЗ(НАИМЕНЬШИЙ(ABS(Target-Data);1);ABS(Target-Data);0))}
```

Если в массиве существует два числа, ближайших к *Target*, формула возвратит значение первого такого числа. На рис. 15.7 представлен пример применения этой формулы.

В этом примере *Data* находится в диапазоне A2:A20, значение *Target* составляет 45. Формула массива находится в ячейке D3 и возвращает ближайшее к *Target* значение, равное 48.

	A	B	C	D	E
1	Data				
2	-12			Target Value -->	45
3	-4			Closest Match:	48
4	4				
5	12				
6	20				
7	32				
8	40				
9	48				
10	56				
11	72				
12	80				
13	88				
14	96				
15	97				
16	105				
17	137				
18	145				
19	165				
20	173				
21					

Рис. 15.7. Формула массива возвращает ближайшее значение

Возвращение последнего значения в столбце

Предположим, существует рабочий лист, в столбец A которого периодически вносятся новые данные. Необходимо отобразить последнее введенное значение. Если столбец A не содержит пустых ячеек, данную задачу можно решить сравнительно просто, не прибегая к использованию формул массива:

```
=СМЕЩ(A1;СЧЕТЗ(A:A)-1;0)
```

Эта формула использует функцию СЧЕТЗ для подсчета количества непустых ячеек в столбце A. Это значение (минус 1) используется в качестве второго аргумента функции СМЕЩ. Например, пусть последнее значение было введено в строку с номером 100. Функция СЧЕТЗ возвратит значение 100, а функция СМЕЩ — значение в ячейке, находящейся на 99 строк ниже ячейки A1.

Однако столбец А вполне может содержать хотя бы одну пустую ячейку внутри диапазона. Это приведет к тому, что предыдущая формула будет выполняться неправильно. Дело в том, что функция СЧЕТЗ не подсчитывает пустые ячейки.

Приведенная ниже формула возвращает содержимое последней непустой ячейки в первых 500 строках столбца А:

```
{=ИНДЕКС (А1 : А500 ; МАКС (СТРОКА (А1 : А500) * (А1 : А500 <> " " ) ) ) }
```

Естественно, что эту формулу можно модифицировать для работы с любым другим столбцом. Для этого вместо ссылок на столбец А введите ссылки на любой другой столбец. Если количество строк превышает 500, замените в формуле число 500 на большее. Однако, чем меньше будет это число, тем больше будет скорость вычислений.



Формулу нельзя размещать в том столбце, который она обрабатывает. Игнорирование этого правила приводит к циклической ссылке. Правда, можно немного модифицировать формулу. Например, можно разместить формулу в первой строке и изменить ссылки так, чтобы они начинались со второй строки.

Возвращение последнего значения в строке

Аналогично можно найти значение последней непустой ячейки в строке. Для этого воспользуемся формулой:

```
{=ИНДЕКС (1 : 1 ; МАКС (СТОЛБЕЦ (1 : 1) * (1 : 1 <> " " ) ) ) }
```

Эта формула осуществляет поиск в первой строке. Чтобы применить ее для других строк, нужно вместо ссылки 1 : 1 задать ссылку на соответствующую строку.

Упорядочение данных с помощью формулы массива

Часто полезным является вычисление последовательности значений в диапазоне данных. Предположим, существует рабочий лист с информацией о годовых продажах, совершенных разными менеджерами. Нас интересует, какое место по объему продаж занимает каждый менеджер.

Использование для этой цели стандартной функции Excel РАНГ может приводить не совсем к тем результатам, которые ожидаются. Например, если два значения находятся на третьем месте, функция РАНГ присваивает каждому из них значение, равное 3. Однако может возникнуть желание определить их ранг как средний относительно других рангов. Иными словами, ранг 3,5 в этой функции для обоих значений связан с третьим местом.

На рис.15.8 представлен рабочий лист, в котором применяется два метода оценки значений, находящихся в столбце В (диапазон Sales). Первый метод (столбец С) использует функцию РАНГ. В столбце D вычисляется ранг с помощью формулы массива.

В ячейке D2 находится формула массива:

```
{=СУММ (1 * (В2 <= Sales) ) - (СУММ (1 * (В2 = Sales) ) - 1) / 2 }
```

Эта же формула скопирована в нижние ячейки.



Каждый ранг вычисляется отдельной формулой массива, а не формулой массива, введенной в весь диапазон ячеек.

	A	B	C	D	E	F	G
1	Salesperson	Sales	Excel's Rank Function	Ranks With Array Formula			
2	Adams	123,000	6	6			
3	Bigelow	98,000	9	10			
4	Fredericks	98,000	9	10	Assigned middle rank		
5	Georgio	98,000	9	10			
6	Jensen	25,000	12	12			
7	Juerez	101,000	8	8			
8	Klein	305,000	1	1			
9	Lynch	145,000	3	3.5			
10	Mayne	145,000	3	3.5	Assigned average rank		
11	Roberton	121,000	7	7			
12	Slokum	124,000	5	5			
13	Wu	150,000	2	2			
14							

Рис. 15.8. Оценка данных с помощью функции РАНГ и формулы массива

Каждая формула массива работает следующим образом. Вычисляется количество значений, больших анализируемого значения. Затем определяется количество равных значений минус один. А потом из первого количества вычитается половина второго.

Создание динамических перекрестных таблиц

Перекрестные таблицы представляют обрабатываемые данные в двух измерениях. Рассмотрим данные примера, представленного на рис. 15.9.

1	Date	Category	Amount				
2	4-Jan	Food	23.50				
3	4-Jan	Transp	15.00				
4	4-Jan	Food	9.12				
5	4-Jan	Food	16.95				
6	4-Jan	Transp	145.50				
7	4-Jan	Lodging	65.95				
8	5-Jan	Transp	20.00				
9	5-Jan	Food	7.80				
10	5-Jan	Food	20.00				
11	5-Jan	Lodging	89.00				
12	6-Jan	Food	9.00				
13	6-Jan	Food	3.50				
14	6-Jan	Food	11.02				
15	6-Jan	Food	78.44				
16	6-Jan	Lodging	75.30				
17	7-Jan	Transp	11.50				
18	7-Jan	Food	15.50				
19	7-Jan	Food	9.50				
20	7-Jan	Lodging	112.00				
21							

Рис. 15.9. Использование формул массива для суммирования данных в динамических перекрестных таблицах

На этом рабочем листе представлен простейший список общих издержек. В столбцах содержится информация о дате (*Date*), статье издержек (*Category*) и общей сумме издержек (*Amount*).

Формула массива суммирует всю эту информацию, представленную в удобной таблице, которая показывает общие издержки по статьям для каждого дня. Ячейка С3 содержит формулу массива:

```
{=СУММ(( $E3=Date) * (F$2=Category) * Amount) }
```

Эта формула скопирована и в остальные 14 ячеек таблицы. Формула отображает общие итоги каждого дня по категориям.

Она суммирует значения в диапазоне *Amount* в том случае, если названия строки и столбца в сводной таблице совпадают с данными в соответствующих диапазонах *Date* и *Category*. Это достигается путем умножения двух булевых (логических) выражений на *Amount*. Если оба булевых выражения — ИСТИНА, то результат равен *Amount*. Если хотя бы одно из булевых выражения — ЛОЖЬ, результат равен 0.

Эту методику можно применить для обработки любого количества статей издержек за любой промежуток времени. Можно отказаться от даты в сводной таблице и заменить ее на фамилию, отделение или регион.



Вместо перекрестных таблиц в *Excel* для аналогичного суммирования данных можно использовать сводные таблицы. Однако сводные таблицы не обновляются автоматически при изменении данных. Таким образом, метод с применением формул массива, описанный здесь, имеет определенное преимущество.

Работа с формулами массива, возвращающими несколько значений

В предыдущей главе содержалась предварительная информация об использовании формул массива, возвращающих несколько значений. В этой главе представляется еще несколько полезных примеров применения таких формул массива. Большинство представленных формул возвращают одно значение или много значений в диапазоне, которые введены по отдельности.

Извлечение положительных значений из диапазона

Представленная ниже формула массива работает с одномерным вертикальным массивом, имеющим имя *Data*. Формула массива расположена в диапазоне, имеющем такой же размер, что и *Data*. Для возвращения только положительных значений (нулевые и отрицательные значения игнорируются) из *Data* используется формула массива:

```
{=ИНДЕКС(Data; НАИМЕНЬШИЙ(ЕСЛИ(Data>0; СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))))); СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))))}
```

Как видно из рис. 15.10, эта формула работает, но несовершенна.

Диапазон *Data* находится в A2:A21, а формула массива введена в диапазон C2:C21. Однако формула массива выдает сообщение об ошибке #ЧИСЛО! для ячеек, которые не содержат значения.

Более комплексная формула массива, избегающая это сообщение об ошибке, имеет вид:

```
{=ЕСЛИ(ЕОШ(НАИМЕНЬШИЙ(ЕСЛИ(Data>0; СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))))); СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))))); "" ; ИНДЕКС(Data; НАИМЕНЬШИЙ(ЕСЛИ(Data>0; СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))))); СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))))}
```

	A	B	C
1	Data		Positive Vals
2	33		33
3	-33		44
4	44		4
5	4		43
6	-5		99
7	0		5
8	43		6
9	-1		7
10	-2		8
11	-3		9
12	-33		10
13	99		11
14	5		12
15	6		#ЧИСЛО!
16	7		#ЧИСЛО!
17	8		#ЧИСЛО!
18	9		#ЧИСЛО!
19	10		#ЧИСЛО!
20	11		#ЧИСЛО!
21	12		#ЧИСЛО!

Рис. 15.10. Применение формулы массива для возвращения из диапазона только положительных значений

Извлечение непустых ячеек из диапазона

Формула этого раздела — вариация формулы из предыдущего. Она работает с одномерным вертикальным массивом с именем *Data*. Формула массива вводится в такой же по размерности диапазон, что и *Data*, и возвращает только непустые ячейки из диапазона *Data*:

```
{=ЕСЛИ(ЕОШ(НАИМЕНЬШИЙ(ЕСЛИ(Data<>"",СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data)))));СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data)))));"";ИНДЕКС(Data;НАИМЕНЬШИЙ(ЕСЛИ(Data<>"",СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data)))));СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))))))}
```

Изменение на противоположный порядка следования элементов в диапазоне

Ниже представленная формула массива работает с одномерным вертикальным массивом, имеющим имя *Data*. Она расположена в диапазоне, имеющем такой же размер, что и *Data*.

Для возвращения массива с противоположным порядком следования элементов, нежели в *Data*, используется формула массива:

```
{=ЕСЛИ(ИНДЕКС(Data;ЧСТРОК(Data)-СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))+1))="";"";ИНДЕКС(Data;ЧСТРОК(Data)-СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))+1))}
```

Присвоим диапазону A2:A20 имя *Data*, а формулу массива введем в диапазон C2:C20. На рис. 15.11 продемонстрирована работа представленной выше формулы массива.

	A	B	C
1	Data Entry Range	Reversed	
2	first	10	
3	second	9	
4	third	8	
5	fourth	7th	
6	5th	6th	
7	6th	5th	
8	7th	fourth	
9	8	third	
10	9	second	
11	10	first	
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			

Рис. 15.11. Порядок следования элементов массива изменяется на противоположный с помощью формулы массива

Динамическая сортировка величин в диапазоне

Предположим, существует рабочий лист с одномерным вертикальным диапазоном с именем *Data*. Для осуществления сортировки значений массива по убыванию воспользуемся формулой массива:

```
{=НАИБОЛЬШИЙ(Data;СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))))}
```

Она должна вводиться в такой же по размеру диапазон, что и *Data*, и производить сортировку только числовых значений.

Для сортировки значений *Data* по возрастанию применяется формула:

```
{=НАИМЕНЬШИЙ(Data;СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data))))}
```

Эти формулы полезны в том случае, когда нужно производить немедленную сортировку введенных данных. Для этого вначале присвойте диапазону, в который вводятся данные, имя *Data*. Затем введите формулу массива в другой диапазон с таким же количеством строк, что и *Data*.

Для ячеек в *Data*, не имеющих значения, формула массива будет выдавать сообщение об ошибке #ЧИСЛО!. Этот факт может раздражать при вводе данных. Улучшенная версия формулы, исключающая появление сообщений об ошибке, имеет вид:

```
{=ЕСЛИ(ЕОШ(НАИБОЛЬШИЙ(Data;СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data)))));"";НАИБОЛЬШИЙ(Data;СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data)))))}
```

Возвращение списка уникальных значений диапазона

Если существует одномерный диапазон с именем *Data*, для вывода списка уникальных элементов этого диапазона можно воспользоваться формулой:

```
{=ИНДЕКС (Data ; НАИМЕНЬШИЙ (ЕСЛИ (ПОИСКПОЗ (Data ; Data ; 0) = СТРОКА (ДВССЫЛ ("1 : "&ЧСТРОК (Data) ) ) ; ПОИСКПОЗ (Data ; Data ; 0) ; " " ) ; СТРОКА (ДВССЫЛ ("1 : "&ЧСТРОК (Data) ) ) ) ) }
```

Эта формула не работает, если диапазон *Data* содержит хотя бы одну пустую ячейку. В незаполненных ячейках формулы массива будет отображаться сообщение об ошибке #ЧИСЛО!. На рис. 15.12 представлен пример применения формулы массива. Диапазон A2 : A20 имеет имя *Data*, а формула массива введена в диапазон C2 : C20.

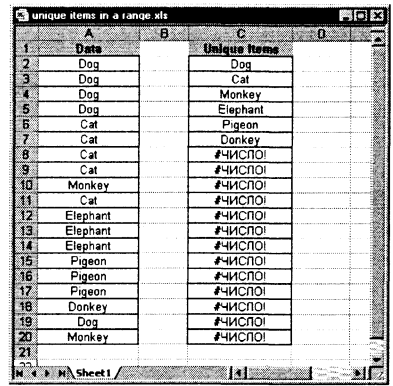


Рис. 15.12. Формула массива применяется для создания списка уникальных элементов диапазона данных

Отображение календаря в диапазоне

На рис. 15.13 представлен календарь, отображаемый в диапазоне ячеек.

Рабочий лист имеет два определенных имени: *m* (для месяцев), *y* (для лет). Единая формула массива вводится во все 42 ячейки, отображающие числа календаря. Диапазон B6 : H11 содержит формулу массива:

```
{=ЕСЛИ (МЕСЯЦ (ДАТА (y ; m ; 1) ) <> МЕСЯЦ (ДАТА (y ; m ; 1) - (ДЕНЬНЕД (ДАТА (y ; m ; 1) ) - 1) + {0 : 1 : 2 : 3 : 4 : 5} * 7 + {1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7} - 1) ; " " ; ДАТА (y ; m ; 1) - (ДЕНЬНЕД (ДАТА (y ; m ; 1) ) - 1) + {0 : 1 : 2 : 3 : 4 : 5} * 7 + {1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7} - 1) }
```

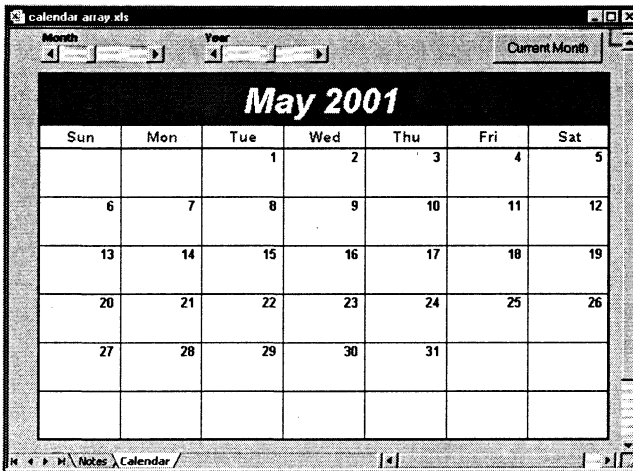


Рис. 15.13. Отображение календаря с помощью одной формулы массива

В действительности, формула массива возвращает значение даты, а ячейки отформатированы так, чтобы отображать только день из даты. Отметим, что формула массива использует массив констант. Для небольшого упрощения формулы массива, можно исключить функцию ЕСЛИ:

```
{=ДАТА(у;м;1)-(РАБДЕНЬ(ДАТА(у;м;1)-1)+{0;7;14;21;28;35})+{0;1;2;3;4;5;6}}
```



В главе 14 можно найти дополнительную информацию о массиве констант.

Эта версия формулы отображает дни из предыдущего и следующего месяцев. Функция ЕСЛИ в первоначальной формуле проверяет каждую дату на предмет принадлежности к текущему месяцу. Если дата не принадлежит к текущему месяцу, возвращается пустая строка.

Возвращение массива с помощью VBA-функций

Финальный пример главы демонстрирует один из случаев, когда нельзя произвести вычисления отдельной формулой массива. Если Excel не предоставляет инструментов для реализации желаемого, необходимо их создать.

Например, автор этих строк убил несколько часов в бесплодных попытках создать формулу массива, возвращающую отсортированный список текстовых данных. Хотя подобная операция успешно проводилась с помощью формулы массива для числовых величин (см. “Динамическая сортировка величин в диапазоне”), произвести то же самое для текста оказалось довольно проблематично.

Представленная формула работает, но только в том случае, если диапазон *Data* не содержит повторяющихся значений:

```
{=ИНДЕКС(Data;ПОИСКПОЗ(СТРОКА(ДВССЫЛ("1:"&ЧСТРОК(Data)));СЧЕТЕСЛИ(Data;"<="&Data);0)}}
```

Поэтому создана пользовательская VBA-функция, названная *SORTED*:

```
Function SORTED(rng, Optional ascending) As Variant
    Dim SortedData() As Variant
    Dim CellCount As Long
    Dim Temp As Variant, i As Long, j As Long
    CellCount = rng.Count
    ReDim SortedData(1 To CellCount)

    If IsMissing(ascending) Then ascending = True

    If rng.Columns.Count > 1 Then
        SORTED = CVErr(xlErrValue)
        Exit Function
    End If

    For i = 1 To CellCount
        SortedData(i) = rng(i)
        If TypeName(SortedData(i)) = "Empty" _
            Then SortedData(i) = ""
    Next i
    On Error Resume Next
    For i = 1 To CellCount
        For j = i + 1 To CellCount
```



```

If SortedData(j) <> "" Then
  If ascending Then
    If SortedData(i) > SortedData(j) Then
      Temp = SortedData(j)
      SortedData(j) = SortedData(i)
      SortedData(i) = Temp
    End If
  Else
    If SortedData(i) < SortedData(j) Then
      Temp = SortedData(j)
      SortedData(j) = SortedData(i)
      SortedData(i) = Temp
    End If
  End If
End If
Next j
Next i

```

```

SORTED = Application.Transpose(SortedData)
End Function

```



Обратитесь к части V для получения информации о создании пользовательских VBA-функций.

Функция SORTED имеет два аргумента: диапазон ссылок и дополнительный второй аргумент, который определяет порядок сортировки. Порядок сортировки по умолчанию — по возрастанию. Если в качестве второго аргумента использовать логическое значение FALSE, произойдет сортировка по убыванию.

Однажды введя процедуру функции SORTED в VBA-модуле, можно использовать функцию SORTED во всех формулах. Приведенная формула массива, например, возвращает содержимое одномерного вертикального массива *Data*, отсортировав его по возрастанию:

```
{=SORTED(Data)}
```

Вводить формулу следует в диапазон такого же размера, что и диапазон *Data*.

Для сортировки содержимого диапазона *Data* по убыванию нужно воспользоваться формулой массива:

```
{=SORTED(Data;False)}
```

Как можно увидеть, применение пользовательских функций приводит к значительному сокращению размера формул. Однако пользовательские функции, как правило, работают медленнее, чем встроенные в Excel.

На рис. 15.14 показан пример того, как эта функция применяется в формуле массива. Диапазон A2 : A17 имеет имя *Data*, а формула массива введена в диапазон C2 : C17.

	A	B	C	D
1	Data Entry		Sorted Data	
2	Arias		Arias	
3	Davis		Colangelo	
4	Trammell		Darr	
5	Jackson		Davis	
6	Magadan		Gonzalez	
7	Mendez		Gwynn	
8	Kotsay		Henderson	
9	Colangelo		Jackson	
10			Klesko	
11	Newin		Kotsay	
12	Henderson		Magadan	
13	Klesko		Mendez	
14	Perez		Newin	
15	Gwynn		Perez	
16	Gonzalez		Trammell	
17	Darr			
18				

Рис. 15.14. Использование пользовательской функции в формуле массива

Резюме

В этой главе приводится большое количество полезных примеров использования формул массива. Эти формулы можно использовать непосредственно или же адаптировать для решения конкретных задач. В главе также представлена пользовательская VBA-функция, возвращающая массив.

В следующей главе описываются умышленные циклические ссылки.

Часть V

Совершенное владение формулами

ГЛАВА 16

“Умышленные циклические ссылки”

ГЛАВА 17

“Методы построения диаграмм”

ГЛАВА 18

“Сводные таблицы”

ГЛАВА 19

“Условное форматирование
и проверка вводимых значений”

ГЛАВА 20

“Создание мегаформул”

ГЛАВА 21

“Инструменты
и методы отладки формул”

Умышленные циклические ссылки

В этой главе...

- ◆ Что такое циклические ссылки?
- ◆ Умышленные циклические ссылки
- ◆ Определение параметров вычислений и итераций
- ◆ Примеры формул, использующих циклические ссылки
- ◆ Возможные проблемы, связанные с использованием циклических ссылок
- ◆ Резюме

ДЛЯ БОЛЬШИНСТВА пользователей, работающих с электронными таблицами, термин *циклические ссылки* ассоциируется с возникновением ошибки. В большинстве случаев циклические ссылки приводят к аварийной ситуации и требуют применения определенных действий по их устранению. Однако в некоторых случаях циклические ссылки могут оказаться полезными. В данной главе представлено несколько примеров использования умышленных циклических ссылок.

Что такое циклические ссылки?

При вводе формул в электронный лист возможно появление сообщения Excel, подобного показанному на рис. 16.1. Таким образом программа сообщает о том, что введенная формула приведет к возникновению *циклической ссылки*. Такая ссылка возникает в случае, когда формула ссылается на ячейку, в которую она занесена, прямо или косвенно. Например, циклическая ссылка возникнет, если в ячейке A10 ввести приведенную ниже формулу (формула обращается к ячейке, содержащей ее):

=СУММ(A1:A10)

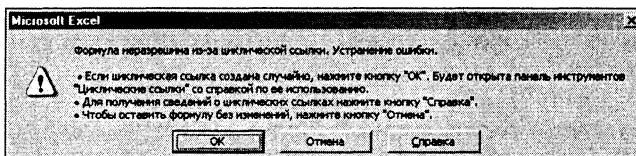


Рис. 16.1. Таким образом Excel сообщает о том, что формула содержит циклическую ссылку

После каждого вычисления формулы в ячейке A10 она снова пересчитывается, так как содержимое данной ячейки изменилось. Теоретически, вычисление должно продолжаться до тех пор, пока значение в ячейке A10 не достигнет бесконечности.

Исправление случайных циклических ссылок

После вывода сообщения о возникновении циклической ссылки Excel предлагает три варианта последующих действий:

- ◆ Щелкните на кнопке **ОК** для обнаружения циклической ссылки (откроется панель инструментов **Циклические ссылки**). В этом варианте решения вам придется постоянно наблюдать окно справочной системы, независимо от того, нужна она или нет.
- ◆ Щелкните на кнопке **Отмена** для ввода формулы в том виде, в каком она есть.
- ◆ Щелкните на кнопке **Справка** для получения информации о циклических ссылках в справочной системе.

Большинство случаев возникновения ошибок циклических ссылок обусловлены просто ошибками ввода символов или неверным указанием диапазонов. Например, при создании формулы в ячейке A10 мог быть ошибочно введен аргумент A1:A10 вместо правильного A1:A9.

Если источник ошибки известен, щелкните на кнопке **Отмена**. В строке состояния Excel выведет сообщение, напоминающее о существовании циклической ссылки. В данном случае это будет сообщение "Цикл: A10". Если активизировать другую рабочую книгу или лист, отображается сообщение "Цикл" (без указания ссылки на ячейку). После этого можно исправить формулу и устранить ошибку.

Если вы получили сообщение о возникновении циклической ссылки и не знаете, где она находится, можно щелкнуть на кнопке **Справка**. При этом будет выведено окно справочной системы с информацией по циклическим ссылкам и отображена панель инструментов **Циклические ссылки** (рис. 16.2). На данной панели инструментов выберите первую ячейку в раскрывающемся списке **Найти циклическую ссылку** и затем изучите формулу в ячейке.

Если вы не можете определить, приводит ли содержимое данной ячейки к возникновению циклической ссылки, щелкните на следующей ячейке в списке **Найти циклическую ссылку**. Продолжайте анализ формул до тех пор, пока в строке состояния не перестанет отображаться сообщение **Цикл**.

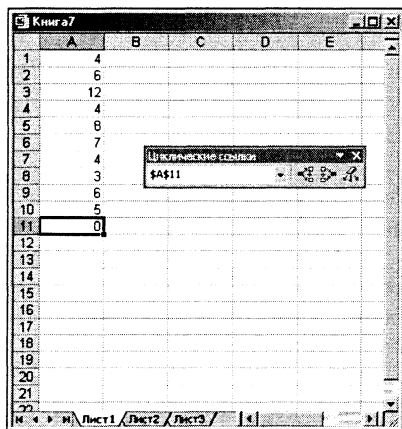


Рис. 16.2. Панель инструментов **Циклические ссылки**

О циклических ссылках

Использование циклических ссылок на реальных примерах описано во врезке "Еще о циклических ссылках" далее в данной главе.



Если опция вычисления итераций включена, Excel не будет отображать диалоговое окно **Циклические ссылки**. Данная функция включается установкой флажка в диалоговом окне **Параметры** (на вкладке **Вычисления**). О данном параметре подробно рассказано ниже.

Косвенные циклические ссылки

Обычно, циклические ссылки достаточно заметны, поэтому их можно легко обнаружить и устранить. Однако в некоторых случаях имеют место непрямые ссылки. Другими словами, одна формула может ссылаться на другую, которая ссылается еще на одну формулу, которая в свою очередь ссылается на исходную. Иногда требуется провести довольно много времени для обнаружения ошибки.



Подробнее об отслеживании циклических ссылок сказано в главе 21.

Умышленные циклические ссылки

Как упоминалось выше, циклические ссылки иногда могут умышленно использоваться для выполнения определенных задач. Циклические ссылки, в случае правильного задания, могут выступать функциональным эквивалентом циклических структур, используемым в языках программирования, например, в VBA. Посредством умышленных циклических ссылок можно проводить рекурсию или итерацию. Каждый промежуточный “результат” вычисления цикла используется в последовательных вычислениях. В итоге решение стремится к конечному значению.

По умолчанию Excel не производит итерационные вычисления. Можно указать программе, что вы желаете, чтобы в рабочих книгах такие вычисления производились. Это можно сделать на вкладке Вычисления диалогового окна Параметры (рис. 16.3).

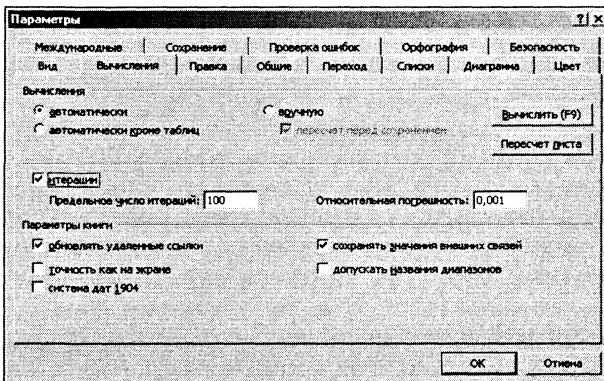


Рис. 16.3. Для вычисления циклических ссылок необходимо установить флажок опции итерации

На рис. 16.4 показан пример рабочего листа, использующего умышленную циклическую ссылку. Компания отдает пять процентов своей чистой прибыли на благотворительность. Однако сам этот вклад относится к расходам и, таким образом, вычитается из значения чистой прибыли.

	A	B	C	D	E	F	G
1	Gross Income	250,000					
2	Expenses	189,500					
3	Contributions	2,881	5% of Net Profit				
4	Net Profit	57,619	Gross Income - Expenses - Contributions				
5							
6							
7							

Рис. 16.4. Создав умышленную циклическую ссылку компания вычитает из чистой прибыли пятипроцентное вложение в виде расходов



Циклическую ссылку нельзя вычислить, если не включена опция поддержки итераций.

Текст в столбце A соответствует именованным ячейкам в столбце B, а ячейка C3 названа Pct. Ячейка Contribution (B3) содержит следующую формулу:

=Pct*Net_Profit

Ячейка Net_Profit (B4) содержит следующую формулу:

=Gross_Income-Expenses-Contribution

Данная формула создает решаемую циклическую ссылку. Excel продолжает вычисления до тех пор, пока результат формулы не приведет к решению.



Читателям первого издания данной книги известно о другом варианте решения данной задачи без использования циклических ссылок. Для вычисления ячейки Net_Profit используется формула:

= (Gross_Income-Expenses) / (1+Pct)

Затем выполняется вычисление ячейки Contribution с помощью формулы:

Pct*Net_profit

Вкладка Вычисления диалогового окна Параметры содержит три группы параметров, относящихся к вычислению циклических ссылок:

- ♦ Флажок итерации: если не установлен, Excel не производит итерационные вычисления и в случае создания формулы с циклической ссылкой отображает диалоговое окно предупреждения. Если создается умышленная циклическая ссылка, необходимо установить данный флажок.
- ♦ Поле Предельное число итераций: определяет максимальное число итераций, производимых Excel. Данное значение не может быть больше 32767.
- ♦ Поле Относительная погрешность: определяет, когда заканчиваются итерации. Например, если установка равна 0,01, итерации прекратятся, когда будет получен результат вычисления, отличающийся на 1 процент от предыдущего значения.



Вычисления продолжают до тех пор, пока не будет достигнуто заданное число итераций, или пока повторное вычисление не будет изменять все ячейки на значение меньше, чем указано в поле Относительная погрешность (до выполнения одного из этих условий). В зависимости от задачи может потребоваться изменить значение в поле Предельное число итераций или Относительная погрешность. Для получения более точного результата нужно уменьшить значение допустимой погрешности. Если после выполнения 100 итераций результат не сходится, можно увеличить их число.

Для того чтобы прочувствовать, как все это работает, откройте пример рабочей книги, представленный в предыдущем разделе. Затем:

1. Перейдите на вкладку Вычисления диалогового окна Параметры и убедитесь, что флажок итерации установлен.
2. Установите Предельное число итераций, равное 1.
3. Установите значение Относительная погрешность, равным 0, 01.
4. Введите другое значение в ячейке Net_profit (ячейка B1).
5. Нажмите клавишу <F9> для вычисления листа.

Так как максимальное число итераций задано равным 1, нажатие клавиши <F9> приведет к выполнению лишь одного цикла вычислений. Нетрудно заметить, что значение ячейки Contribution не сходится с решением. Нажмите клавишу <F9> еще несколько раз, и вы сможете увидеть, как значение ячейки становится более близким к искомому решению. После того как решение найдено, нажатие клавиши <F9> не производит существенных изменений. Если задано большое число итераций, решение появляется фактически незамедлительно (если оно не включает сложные вычисления).

Определение параметров вычислений и итераций

Примите к сведению, что все открытые рабочие книги используют одинаковые установки вычислений и итераций. Если, например, есть две открытые рабочие книги, нельзя для одной установить автоматическое выполнение вычислений, а в другой выполнять вычисления вручную. Хотя рабочая книга может быть сохранена со своими особыми параметрами (например, вычисления вручную без итераций), данные установки могут изменяться при открытии другой рабочей книги.

При определении параметров вычислений и итерации, которые необходимо использовать, Excel руководствуется следующими правилами:

- ◆ Первая открытая рабочая книга использует режим вычислений, который был сохранен с ней. Если открыть другие книги, в них будет использоваться тот же режим вычислений.

Например, имеется две рабочие книги Книга1 и Книга2. В Книга1 флажок итерации сброшен (значение по умолчанию), а в Книга2 (использующей умышленные циклические ссылки) данный флажок установлен. Если сначала открыть Книга1, а затем Книга2, в обеих рабочих книгах функция вычисления итераций будет отключена. Но если открыть их в обратной последовательности, данная функция будет включена для обеих книг.

- ◆ Изменение режима вычислений для одной книги приводит к изменению режима вычислений всех рабочих книг.

Если обе рабочие книги, Книга1 и Книга2, открыты, изменение режима вычислений в одной из них коснется их обеих.

- ◆ Все рабочие листы книги используют один и тот же режим вычислений.
- ◆ Если все рабочие книги закрыты, и вы создаете новую, она будет использовать те же установки, что имела последняя из закрытых рабочих книг. Исключение имеет место в случае создания рабочей книги на основе шаблона. В этом случае используются параметры, заданные в шаблоне.
- ◆ Если режим вычислений в книге изменен, а файл пересохранен, к рабочей книге будет применен текущий режим вычислений.

Примеры формул, использующих циклические ссылки

Ниже приведено несколько примеров использования умышленных циклических ссылок. Они демонстрируют применение циклических ссылок для создания “временного снимка” ячейки, вычисления всегда-самого-большого значения, решения рекурсивного уравнения и решения системы уравнений.



Для работы приведенных ниже примеров флажок итерации должен быть установлен. Выберите команду Сервис⇒Параметры и щелкните на вкладке Вычисления. Убедитесь, что флажок итерации установлен.

Создание временного снимка ячейки

На рис. 16.5 показан рабочий лист, разработанный таким образом, что элементы в столбце А отмечаются “временным снимком” в столбце В. Формулы в столбце В отслеживают значения в соответствующих ячейках столбца А. При внесении записи в столбце А формула возвращает текущие дату и время.

В ячейке В2 содержится формула, скопированная в другие ячейки столбца В:

```
=ЕСЛИ(ЕПУСТО(А2); ""; ЕСЛИ(В2=""; ТДАТА()  
(); В2))
```

Данная формула использует функцию ЕСЛИ для проверки ячейки А2. Если ячейка пуста, формула возвращает пустую строку. Если ячейка А2 не пустая, формула проверяет значение в ячейке В2 (имеет место циклическая ссылка). Если В2 пуста, формула возвращает дату и время. Второй оператор ЕСЛИ используется для того, чтобы предотвратить выполнение повторного пересчета.

	A	B	C	D	E
1	Date Entry	Time Stamp			
2	54.5	6:10:99 1:32 PM			
3	509.32	6:10:99 1:33 PM			
4	434	5:1:01 12:58 PM			
5	290.4	5:1:01 12:58 PM			
6	43.34	5:1:01 12:59 PM			
7					
8					
9					
10					
11					

Рис. 16.5. Использование формулы с циклической ссылкой для создания временных снимков записей в ячейках столбца А

Вычисление всегда-самого-большого значения

На рис. 16.5 показан рабочий лист, отображающий продажи отдельных торговых представителей. Данный лист обновляется ежемесячно: новые показатели продаж заменяют значения в столбце В.

Формула в ячейке С1 отслеживает всегда-самые-большие показатели — максимальное значение, введенное когда-либо в столбце В. Эта формула, использующая циклическую ссылку, выглядит следующим образом:

```
=МАКС(В:В; С)
```

Формула использует функцию МАКС для возврата максимального значения столбца В или ячейки С1. В данном примере отображается значение 98223. Оно относится к одному из предыдущих месяцев (другими словами, в текущее время данное значение не содержится в столбце В).

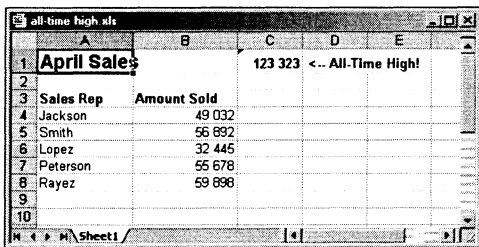


Рис. 16.6. Использование формулы с циклической ссылкой для постоянного отслеживания максимального значения когда-либо введенного в столбце B

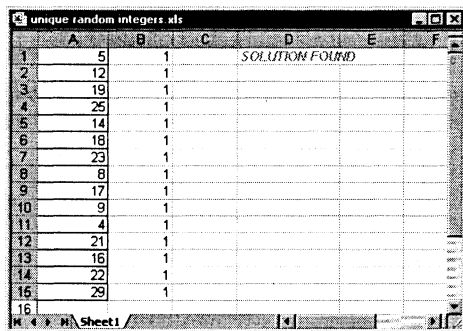


Рис. 16.7. Использование формул с циклическими ссылками для генерирования уникальных случайных чисел в столбце A

Генерирование уникальных случайных чисел

Циклические ссылки могут использоваться для генерирования уникальных случайных чисел из заданного диапазона. Рабочий лист, представленный на рис. 16.7, генерирует 15 случайных чисел в диапазоне от 1 до 30 в столбце A. Числа генерируются уникальными, то есть они не повторяются. Данный метод может применяться для получения случайных выигрышных номеров лотереи.

Столбец B содержит формулы, подсчитывающие, сколько раз использовалось каждое отдельное число в диапазоне A1:A15. Например, приведенная ниже формула из ячейки B1 показывает, сколько раз в диапазоне A1:A15 использовалось значение, содержащееся в ячейке A1.

`=СЧЕТЕСЛИ(A1:A15;1)`

Каждая формула в столбце A содержит циклическую ссылку. Формула проверяет сумму значений ячеек в столбце B. Если сумма не равна 15, генерируется новое случайное число. Когда сумма ячеек столбца B становится равной 15, значит все значения в столбце A уникальны. Вот какой вид имеет формула в ячейке A1:

`=ЕСЛИ(СУММ(B1:B15) <> 15; ЦЕЛОЕ(СЛЧИС()*30+1); A1)`

Ячейка D1 содержит формулу, которая отображает статус операции. Если сумма значений ячеек столбца B не равна 15, то в D1 отображается `CALC AGAIN` (для проведения вычислений нажмите <F9>). Когда в ячейках столбца B отображаются все единицы, в D1 содержится `SOLUTION FOUND`.

`=ЕСЛИ(СУММ(B1:B15) <> 15; "CALC AGAIN"; "SOLUTION FOUND")`

Для генерирования нового набора случайных чисел выберите любую ячейку из столбца B, затем нажмите клавишу <F2> для ее редактирования и далее нажмите клавишу <Enter>. Количество вычислений зависит от:

- ♦ установок параметров итерации на вкладке Вычисления диалогового окна Параметры. Если установить большее число итераций, шанс найти 15 уникальных чисел будет выше;
- ♦ соотношения количества требуемых чисел и количества возможных значений. В данном примере ищется 15 чисел из 30 значений. Меньшее количество вычислений потребуется, например, если искать 15 значений среди 100 возможных.

Решение рекурсивного уравнения

К рекурсивным уравнениям относятся уравнения, в которых переменная содержится с обеих сторон знака равенства. Ниже приведены примеры рекурсивных уравнений:

$$x=1/(x+1)$$

$$x=\cos(x)$$

$$x=\text{SQRT}(x+5)$$

$$x=2^{(1/x)}$$

$$x=5+(1/x)$$

Для решения рекурсивного уравнения убедитесь, что флажок итерации установлен. Затем преобразуйте уравнение в формулу со ссылкой на саму себя. Для решения первого уравнения введите в ячейку A1 следующую формулу:

$$=1/(A1+1)$$

Формула сходится к числу 0,618033988749895, что является значением x , удовлетворяющим уравнению.

В некоторых случаях данная техника не работает. Давайте, например, рассмотрим следующее уравнение:

$$x=5+(1/x)$$

Если в ячейку A1 ввести приведенную ниже формулу, будет возвращен признак ошибки #ДЕЛ/0! вследствие того, что итерации начинаются с нуля (и деление на ноль приводит к возникновению ошибки):

$$=5+1/A1$$

Для решения данного типа уравнений необходимо использовать две ячейки. Вот пошаговое описание данной методики:

1. Введите любое ненулевое значение в ячейке A1.

2. В ячейке A2 введите следующую формулу:

$$=5+1/A1$$

3. В ячейке A1 введите следующую формулу:

$$=A2$$

В обеих ячейках A1 и A2 будет отображено значение 5,19258235429625, что является решением данного уравнения. Обратите внимание, что ввод ненулевого значения на первом шаге обеспечивает ненулевое начало для рекурсии. После замены данного значения на формулу (шаг 3) исходное значение ячейки A1 продолжает выступать в качестве начального значения для формулы в ячейке A2.



Вследствие установленного способа выполнения вычислений в Excel ячейка с исходным значением должна находиться левее или выше самой формулы.

На рис. 16.8 представлен рабочий лист, на котором вычисляется несколько рекурсивных уравнений. Обратите внимание, что уравнения в строках 5 и 6 требуют наличия ячейки с исходным значением. Формулы в столбце E используют значения из столбца C для осуществления проверки результата. Например формула в ячейке E2 имеет такой вид:

$$=1/(C2+1)$$

	A	B	C	D	E	F
1	Equation	Seed	Circular Ref Formula		Check	
2	$x = 1/(x+1)$		0,618033989		0,61803399	
3	$x = \cos(x)$		0,739085133		0,73908513	
4	$x = \sqrt{X+5}$		2,791287847		2,79128785	
5	$x = 2^{\wedge}(1/x)$	1,559610469	1,559610469		1,55961047	
6	$x = 5 + (1/x)$	5,192582404	5,192582404		5,1925824	
7						
8						
9						

Рис. 16.8. В данной рабочей книге циклические ссылки используются для решения нескольких рекурсивных уравнений

Решение системы уравнений с помощью циклических ссылок

Иногда циклические ссылки могут использоваться для решения системы уравнений. Рассмотрим следующую систему уравнений:

$$\begin{aligned} 3x + 4y &= 8 \\ 3x + 8y &= 20 \end{aligned}$$

Требуется найти значения x и y , одновременно удовлетворяющих оба уравнения. Сначала необходимо записать уравнения в виде функций x и y . Первое уравнение в виде функции x будет иметь следующий вид:

$$x = (8 - 4y) / 3$$

Второе уравнение представленное в виде функции y приобретет следующий вид:

$$y = (20 - 3x) / 8$$

Как показано на рис. 16.9, ячейке B5 присвоено имя X а ячейке B6 — Y. Формулы в данных ячейках соответствуют приведенным выше уравнениям. Формула в ячейке B5 имеет следующий вид:

$$= (8 - (4 * Y)) / 3$$

В ячейке B6 (Y)записана формула:

$$= (20 - (3 * X)) / 8$$

На рисунке также представлены графики двух данных уравнений. Пересечение двух кривых дает значения X и Y удовлетворяющее решение системы уравнений.

Еще о циклических ссылках

Использование циклических ссылок в реальной жизни описана во врезке "Еще о циклических ссылках" ранее в данной главе.

Обратите внимание на описанную выше циклическую ссылку. Ячейка X ссылается на ячейку Y, а ячейка Y в свою очередь ссылается на ячейку X. Значения в этих ячейках сходятся к решению:

$$\begin{aligned} X &= -1,333 \\ Y &= 3,000 \end{aligned}$$

Использование циклических ссылок для решения систем уравнений больше интересно в демонстративных целях, чем в практических. Вы можете обнаружить, что многие итерационные вычисления никогда не сходятся. Другими словами, результат успешного

пересчета никогда не будет являться решением. Для примера давайте рассмотрим приведенную ниже систему уравнений. Решение для нее существует, однако его нахождение с помощью циклических ссылок невозможно.

$$x = 4 - y / 2$$

$$x = 3 + 2x$$

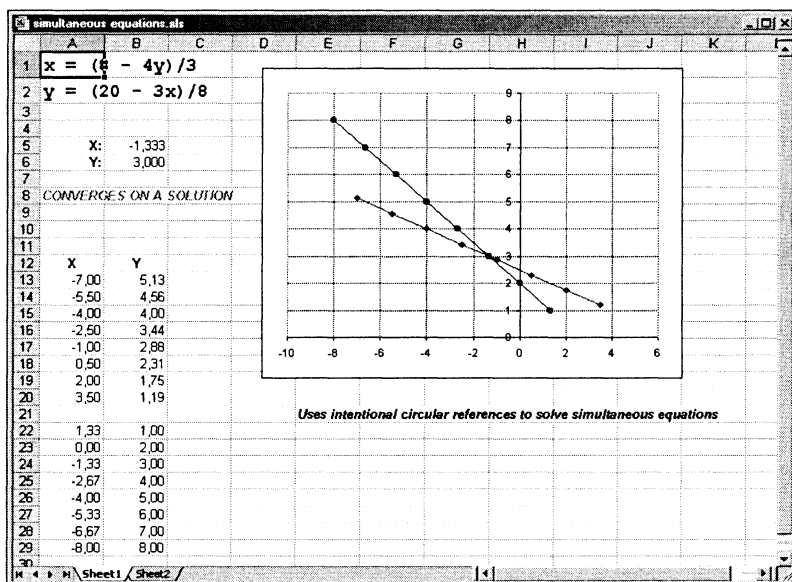


Рис. 16.9. На данном рабочем листе решена система двух уравнений



Лучшим способом решения систем уравнений в Excel является использование матриц. Примеры их использования приводятся в главе 10.

Возможные проблемы, связанные с использованием циклических ссылок

Хотя применение умышленных циклических ссылок может оказаться полезным, использование данного средства может таить в себе некоторые проблемы. Лучшее, что можно посоветовать, это использовать циклические ссылки с осторожностью и пониманием того, как они работают.

Для того чтобы воспользоваться возможностями умышленных циклических ссылок, необходимо, чтобы средство вычисления итераций было включено. В этом случае Excel не будет выводить предупреждения о наличии циклических ссылок. Поэтому существует риск создания случайных циклических ссылок, о наличии которых пользователю даже не будет известно.

Количество итераций, указанное в поле Максимальное число итераций применяется ко всем формулам в рабочей книге, а не только к тем, в которых используются циклические ссылки. Если рабочая книга содержит большое число сложных формул, это может привести к

существенному замедлению работы. Поэтому при использовании умышленных циклических ссылок стремитесь к максимальной простоте рабочих листов.

Иногда может возникать необходимость отправки рабочих книг, использующих циклические ссылки, другим пользователям. Если при открытии рабочей книги функция вычисления итераций не включена, Excel отобразит сообщение об ошибке вследствие наличия циклических ссылок, что может озадачить большую часть пользователей.

Резюме

В данной главе рассмотрено, как Excel работает с циклическими ссылками. Хотя в большинстве случаев наличие циклических ссылок обозначает ошибку, иногда их использование можно применить в практических целях. Для работы с ними необходимо включить функцию вычисления итераций.

В следующей главе рассмотрено, как формулы расширяют возможности по построению диаграмм..

Методы построения диаграмм

В этой главе...

- ◆ Представление данных на диаграммах
- ◆ Интерактивное отображение данных
- ◆ Создание экстравагантных проектов
- ◆ Диаграммы тенденций
- ◆ Полезные методы построения диаграмм
- ◆ Резюме

Excel поддерживает более 100 различных типов диаграмм; и почти все элементы и параметры диаграмм задаются пользователем. В данной главе, предполагая, что читатели знакомы со средствами построения диаграмм Excel, продемонстрированы некоторые полезные приемы создания диаграмм, в большей части из которых используются формулы.

Представление данных на диаграммах

Диаграммы используются главным образом для наглядного представления таблиц данных. Отображение данных с помощью хорошо построенной диаграммы может сделать их более понятными. Так как диаграммы представляют собой рисунок, их использование особенно полезно для понимания назначения больших последовательностей данных, а также взаимосвязи разных значений. Построение диаграмм может помочь в нахождении тенденций и анализе закономерностей, которые нельзя обнаружить при обычном изучении массивов данных.

Диаграммы строятся на основе данных, находящихся на рабочем листе. Они вводятся непосредственно, или получаются в результате вычисления формул. Обычно используемые для построения диаграмм данные находятся на одном листе, однако это не является строгим правилом. Диаграмма может использовать данные любого количества рабочих листов или даже разных рабочих книг.

Знакомство с формулой РЯД

Диаграмма состоит из одной или нескольких рядов данных, каждый из которых представляется на диаграмме линией, столбцом или сектором. Для каждого ряда диаграмма использует формулу РЯД. При выборе на диаграмме ряда данных, эта формула отображается в строке формул. Формула РЯД не является формулой в обычном понимании этого слова, ее нельзя записать в ячейке рабочего листа, а также использовать в функции рабочего листа. В то же время имеется возможность изменения аргументов формулы РЯД. Данная формула имеет следующий синтаксис:

=РЯД (имя ; подписи_категорий ; значения ; порядок)

В формуле РЯД использованы следующие аргументы:

- ◆ *имя*. (Не обязательно.) Имя, используемое в легенде. Если диаграмма содержит лишь один ряд данных, данный аргумент используется в качестве заголовка.
- ◆ *подписи_категорий*. (Не обязательно.) Диапазон, содержащий подписи осей категории. Если этот аргумент опущен, Excel использует последовательные целые числа, начиная с 1.
- ◆ *значения*. (Обязательно.) Диапазон, содержащий значения.
- ◆ *порядок*. (Обязательно.) Целое число, определяющее порядок построения ряда (используется только тогда, когда диаграмма содержит более одного ряда).

Ссылки на диапазон, используемые в формуле РЯД, всегда абсолютны и всегда включают имя листа. Например:

```
=РЯД(Лист1!$B$1; ; Лист1!$B$2 : $B$7 ; 1)
```

Ссылка может указывать и на разрывный диапазон. В этом случае каждый диапазон разделяется точками с запятыми, а весь аргумент заключается в скобки. В следующей формуле РЯД набор значений состоит из диапазонов В2 : В3 и В5 : В7:

```
=РЯД( ; ; (Лист1!$B$2 : $B$3 ; Лист1!$B$5 : $B$7) ; 1)
```

Хотя формула РЯД может ссылаться на данные в других рабочих листах, все данные ряда должны находиться на одном листе. Следующая формула РЯД недействительна, так как в ней присутствуют ссылки на данные двух разных рабочих листов:

```
=РЯД( ; ; (Лист1!$B$2 ; Лист2!$B$2) ; 1)
```

Использование имен в формуле РЯД

В формуле РЯД ссылки на диапазон могут быть заменены именами диапазонов. При этом Excel изменяет ссылку в формуле РЯД, включая в нее имя рабочей книги. Например, приведенная ниже формула РЯД использует диапазон с именем *MyData* (расположенный в рабочей книге с именем *budget.xls*). Excel добавил имя рабочей книги и восклицательный знак.

```
=РЯД(Лист1!$B$1; ; budget.xls!MyData ; 1)
```

Советы по построению диаграмм

- ◆ Ниже приведен ряд советов, которые могут оказаться полезными при построении диаграмм.
- ◆ Чтобы создать диаграмму единственным нажатием клавиши, выделите данные, которые хотите отобразить графически, и нажмите клавишу <F11>. В результате будет создан новый лист, содержащий диаграмму используемого по умолчанию типа.
- ◆ Размер диаграммы можно изменить в соответствии с размером окна с помощью команды Вид⇒По размеру окна. Если данная функция включена, размер диаграммы автоматически изменяется в соответствии с изменяемым размером окна рабочей книги (диаграмма всегда точно заполняет окно). В данном режиме вид диаграммы, с которой вы работаете, может не соответствовать виду диаграммы на бумаге.
- ◆ Если необходимо создать большое число диаграмм одного типа, изменение типа диаграммы, используемого по умолчанию, будет значительно эффективнее, нежели форматирование каждой из диаграмм по отдельности. После этого можно создать все диаграммы без указания их типа. Для того чтобы изменить тип диаграмм, используемых по умолчанию, выберите команду Диаграмма⇒Тип диаграммы и затем укажите необходимую опцию. Щелкните на кнопке Сделать стандартной. Можно также сохранить ее как пользовательский тип диаграммы с возможностью последующего использования. Для этого щелкните на вкладке Нестандартные и затем на кнопке Добавить.

- ◆ Для печати внедренной диаграммы на отдельной странице выделите ее и выберите команду Файл⇒Печать (или щелкните на кнопке Печать). Excel выведет на печать лишь диаграмму, сам рабочий лист распечатываться не будет.
- ◆ Если вы не хотите, чтобы определенная внедренная диаграмма выводилась на печать, щелкните на ней правой кнопкой мыши и выберите из контекстного меню команду Формат области диаграммы. В диалоговом окне Формат области диаграммы щелкните на вкладке Свойства и сбросьте флажок Выводить объект на печать.
- ◆ В некоторых случаях выделение элементов диаграмм с помощью мыши может быть довольно неудобным. Использование клавиатуры для этой цели в некоторых случаях значительно проще. Когда диаграмма активизирована, циклическое перемещение между всеми ее элементами осуществляется с помощью клавиш со стрелками вверх и вниз. Когда выделен ряд данных, при нажатии клавиш со стрелками вправо и влево осуществляется перемещение между отдельными значениями ряда.
- ◆ При выделении элемента диаграммы вы обнаружите, что многие из кнопок панелей инструментов, используемых обычно при форматировании рабочих листов, также применяются и к выбранным элементам диаграммы. Например, если выделить Область построения диаграммы, с помощью инструмента Цвет заливки панели инструментов Форматирование можно изменить ее цвет. Если выбрать элемент, содержащий текст, для изменения его цвета можно использовать инструмент Цвет текста.
- ◆ До версии Excel 97 при щелчке на внедренной диаграмме выделялся объект диаграммы. После этого можно было изменить его свойства. Для активизации диаграммы необходимо было щелкнуть на ней дважды. Начиная с версии Excel 97 щелчок на внедренной диаграмме активизирует диаграмму, содержащуюся внутри объекта диаграммы. После этого можно изменить свойства объекта диаграммы на вкладке Свойства диалогового окна Формат диаграммы. Для выделения самого объекта диаграммы необходимо при щелчке на диаграмме удерживать нажатой клавишу <Ctrl>. Выделение объекта диаграммы может потребоваться для изменения его имени в поле Имя.
- ◆ Можно удалить все ряды данных из диаграммы. Если это сделать, диаграмма становится пустой. Однако при этом она сохраняет свои установки. Таким образом, можно добавить на пустую диаграмму новые ряды данных и тем самым создать диаграмму, выглядящую подобно прежней.
- ◆ Для более полного управления положением диаграммы удерживайте нажатой клавишу <Ctrl>. При нажатии клавиш со стрелками диаграмма будет смещаться на один пиксель в указанном направлении.
- ◆ Чтобы создаваемая линия не прерывалась в точках, не содержащих информации, введите в пустых ячейках вашего диапазона формулу =ИД() .

Использование имен в формуле ряда имеет одно существенное преимущество: при внесении изменения, относящегося к имени диапазона, диаграмма автоматически отображает новые данные. Давайте, например, предположим, что имя *MyData* в предыдущем примере относится к диапазону A1 : A20. На диаграмме будет отображено двадцать значений из указанного диапазона. Позже можно с помощью команды Вставка⇒Имя⇒Определить переопределить диапазон *MyData*, скажем, как A1 : A30. Теперь диаграмма будет отображать 30 значений, указанных в новом диапазоне *MyData* (редактирования диаграммы не потребуется).

Как упоминалось выше, формула РЯД не может использовать функции рабочего листа. Однако вы *можете* создать именованные формулы (использующие функции), которые применяются в формуле РЯД. Как можно будет увидеть из последующего материала, данная методика позволяет реализовать самые невероятные задачи.

Отделение ряда диаграммы от его данных

Обычно диаграмма Excel использует данные, хранимые в определенном диапазоне. Стоит изменить данные, и диаграмма автоматически обновится. Иногда может возникнуть необходимость “отсоединения” диаграммы от ее данных и создания *статической диаграммы*, которая всегда сохраняет свой вид. Например, если вы изображаете данные, полученные методом подбора, то можете сохранить диаграмму, представляющую определенную базисную линию так, чтобы ее можно было сравнивать с другими графиками. Существует два различных способа создания подобной диаграммы:

- ♦ *Вставка в виде рисунка.* Активизируйте диаграмму, а затем нажмите клавишу <Shift> и выберите команду Правка⇒Копировать рисунок (команда Копировать рисунок доступна лишь тогда, когда при открытии меню Правка нажата клавиша <Shift>). После этого нажмите клавишу <Shift> и выберите команду Правка⇒Вставить рисунок. Результатом данных действий будет вставка моментального снимка скопированной диаграммы.
- ♦ *Преобразование ссылки на диапазон в массив.* Щелкните на ряде диаграммы, а затем щелкните в строке формул для активизации формулы РЯД. Нажмите клавишу <F9> для преобразования диапазона в массив. Повторите это для каждого из рядов. Данная техника (в противоположность методу создания рисунка) сохраняет возможность дальнейшего редактирования диаграммы. Описанная процедура не работает в случае большого объема данных вследствие существования ограничения на длину формулы РЯД.

Создание ссылок на ячейки

Вы можете к различным элементам диаграммы добавить ссылки на ячейки. Добавление таких ссылок позволяет сделать диаграмму более динамичной. Динамические ссылки устанавливаются для заголовков диаграммы, подписей данных, дополнительного описательного текста и рисунков.

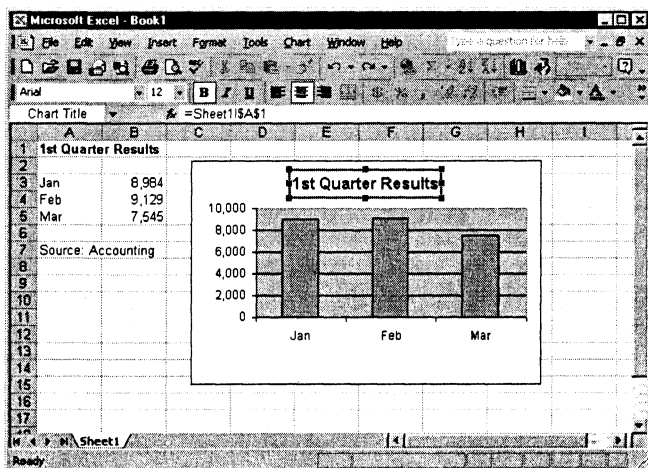


Рис. 17.1. Заголовок диаграммы связан с ячейкой A1

Добавление ссылок на заголовки

Обычно заголовки диаграммы (заголовок диаграммы, подпись оси) не связаны ни с какими ячейками. Другими словами, они содержат статический текст, который изменяется лишь при ручном редактировании. Однако вы можете создать ссылку так, что заголовок будет ссылаться на ячейку рабочего листа.

Перед созданием заголовка со ссылкой убедитесь, что диаграмма содержит нужный заголовок элемента диаграммы. Если на диаграмме нет нужного заголовка, его можно добавить в диалоговом окне **Параметры диаграммы** (для отображения этого диалогового окна выберите команду **Диаграмма**⇒**Параметры диаграммы**). Затем выберите заголовок и щелкните в строке формул. Введите знак равенства и затем щелкните на ячейке, содержащей текст заголовка. Результатом такой операции будет формула, содержащая ссылку на страницу и абсолютную ссылку на ячейку (например, =Лист1!\$A\$1). Нажмите клавишу <Enter> для присоединения формулы к заголовку диаграммы. На рис. 17.1 показана диаграмма, в которой заголовок диаграммы связан с ячейкой A1.

Добавление ссылок к подписям данных

Вероятно, вам известно, что Excel позволяет подписать каждую точку данных на диаграмме. Это можно сделать на вкладке **Подписи данных** диалогового окна **Формат ряда данных**. К сожалению, данное средство не обладает достаточной гибкостью. Скажем, вы не можете определить диапазон, содержащий подписи. Чтобы сделать это, щелкните один раз на любой из подписей для выделения всех подписей, затем щелкните еще один раз для выбора отдельной подписи данных. Когда выбрана отдельная подпись, можно добавлять любой текст. Или же можно определить ссылку на ячейку, щелкнув в строке формул и введя формулу ссылки (например, =Лист1!\$A\$1).



Пакет **Power Utility Pak** включает удобную утилиту, облегчающую добавление подписей данных на диаграмму путем указания диапазона рабочего листа; несмотря на частое возникновение потребности в данном средстве, разработчики Microsoft не позаботились о его добавлении в саму программу.

Добавление текстовых ссылок

Вам может понадобиться отобразить на диаграмме определенный текст (вроде пояснительной записки), хранящейся в любой из ячеек. Сделать это можно весьма просто. Сначала активизируйте диаграмму. Затем щелкните в строке формул, введите знак равенства и щелкните на ячейке, содержащей текст. Нажмите клавишу <Enter>. Excel создаст в центре диаграммы текстовое поле (рис. 17.2). Данное поле можно расположить в любом месте и применить к нему любой тип форматирования.

Чтобы добавить несвязанное текстовое поле, просто выделите диаграмму, введите текст в строке формул и нажмите клавишу <Enter>.



Добавление на диаграмму объектов вроде текстового поля может вызвать неполадки. Например, возможны ситуации, когда последовательность операций с диаграммой (таких как удаление осей или легенды) может привести к тому, что объекты перестают отображаться. Несмотря на то, что о данной ошибке давно известно, разработчики Microsoft до сих пор ее не устранили. Для получения лучшего результата добавляйте текстовое поле лишь после того, как все остальные изменения на диаграмме уже проведены.

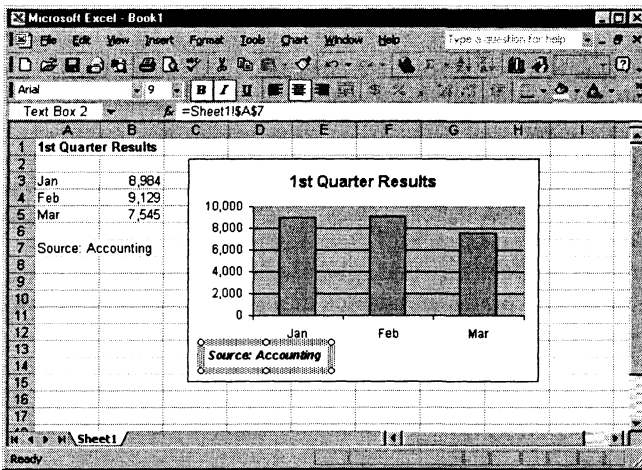


Рис. 17.2. Текстовое поле, связанное с ячейкой

Добавление ссылок на рисунки

Excel обладает возможностью отображения внутри диаграммы таблицы данных. Данным средством можно воспользоваться на третьем шаге мастера диаграмм. Таблицу данных также можно добавить в существующую диаграмму на вкладке Таблица данных диалогового окна Параметры диаграммы. Данное средство отображает таблицу со значениями, используемыми для построения диаграммы. Иногда это весьма удобно, хотя гибкости данной функции часто недостает. Например, имеются серьезные ограничения на использование параметров форматирования, средства управления положением таблицы данных тоже неактивны (она всегда располагается под диаграммой). Связанный рисунок диапазона данных, на основе которых построена диаграмма, является хорошей альтернативой таблицы данных (пример приведен на рис. 17.3).

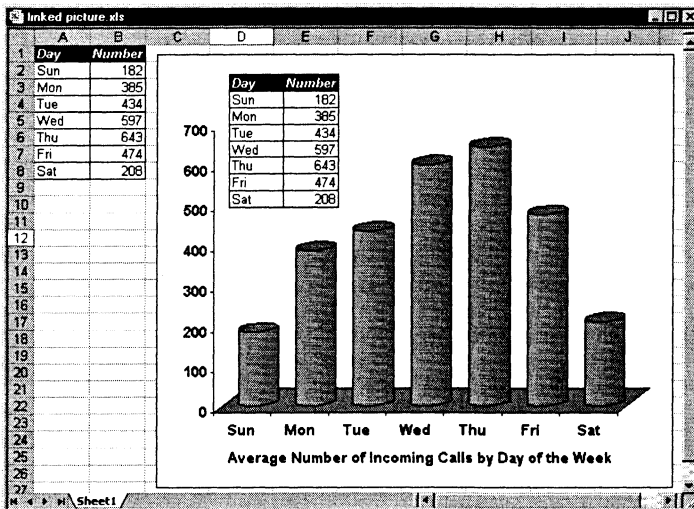


Рис. 17.3. Данная диаграмма содержит связанный рисунок диапазона A1 : B8

Чтобы создать на диаграмме связанный рисунок, сначала создайте диаграмму стандартным способом. После этого выполните следующие действия:

1. Выделите диапазон, который вы хотите включить в диаграмму.
2. Выберите команду **Правка**⇒**Копировать**.
3. Активизируйте диаграмму.
4. Нажмите клавишу <Shift> и затем выберите команду **Правка**⇒**Вставить рисунок**. При этом будет вставлен *несвязанный* рисунок диапазона.
5. Для создания связи выберите рисунок, а затем введите в строке формул ссылку на диапазон. Это можно сделать легко: ввести знак равенства и повторно выделить диапазон.

Теперь рисунок содержит активную ссылку на диапазон. Если изменить значения или форматирование ячеек, эти изменения будут отображены в связанном рисунке. Данная методика также применяется к листам диаграмм.

График выполнения работ

Вероятно, вы знакомы с гистограммами, используемым для отображения процентного соотношения выполнения проекта. Создание подобной диаграммы в Excel осуществляется довольно просто. Прием состоит в создании диаграммы, использующей в качестве ряда данных единственную ячейку (в ней содержится значение, показывающее процент выполнения плана).

На рис. 17.4 представлен рабочий лист, созданный для отслеживания ежедневного прогресса в реализации проекта: увеличение числа потребителей на 1000 за 15-дневный период. Ячейка B18 содержит запланированное значение, а в ячейке B19 записана простая формула суммирования:

=СУММ(B2 : B16)

Формула в ячейке B21 вычисляет процент выполнения плана:

=B19 / B18

При вводе новых данных в столбце B формула отображает текущий результат.

Для создания диаграммы выберите ячейку B21 и щелкните на кнопке **Мастер диаграмм**. Проследите за тем, чтобы перед ячейкой B21 располагалась пустая строка. В случае ее отсутствия в Excel для построения диаграммы используется не одна ячейка, а целый ряд данных. Когда ячейка B21 изолирована от других данных, мастер диаграмм использует единственную ячейку. На первом этапе работы мастера укажите тип диаграммы **Гистограмма** и первый ее подтип (**Гистограмма с накоплением**). Дважды щелкните на кнопке **Далее** и на третьем этапе проведите некоторые изменения: добавьте **Название диаграммы** (вкладка **Заголовки**), удалите ось (x) категорий (вкладка **Оси**), удалите легенду (вкладка **Легенда**) и включите в подписи значения (вкладка **Подписи данных**). Для создания диаграммы щелкните на кнопке **Готово**.

После этого выполните некоторые дополнительные настройки. Дважды щелкните на гистограмме для отображения диалогового окна **Формат ряда данных**. Щелкните на вкладке **Параметры** и установите значение параметра **Ширина зазора** равным 0 (в результате этого действия гистограмма будет занимать полную ширину области построения). Также можно изменить используемую в гистограмме заливку. Это можно сделать на вкладке **Вид**. В приведенном примере использована градиентная заливка. Затем дважды щелкните на вертикальной оси для вызова диалогового окна **Формат оси**. На вкладке **Шкала** установите минимальное значение равным 0, а максимальное значение 1.

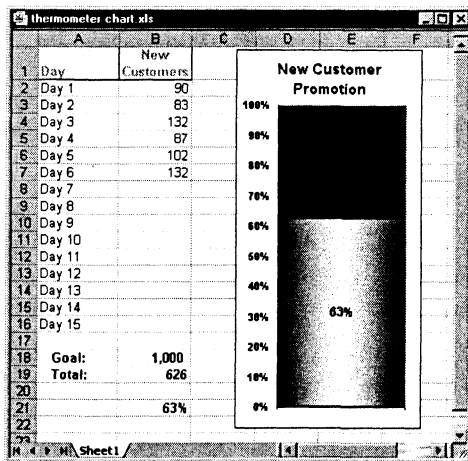


Рис. 17.4. Данная диаграмма отображает процент выполнения намеченного проекта

Кроме того, можно провести дополнительные косметические изменения по своему усмотрению. Например, можно изменить ширину диаграммы для того, чтобы она приобрела вид термометра, а также изменить шрифты, цвета и так далее.

Создание диаграммы Гантта

Диаграмма Гантта показывает время, необходимое для выполнения каждой отдельной задачи проекта. На рис. 17.5 показаны данные, которые использованы для построения простой диаграммы Гантта, представленной на рис. 17.6. Создание диаграммы данного типа в Excel не является сложной задачей, но все же для этого требуется проведение определенной подготовительной работы.

Рис. 17.5. Данные, используемые в диаграмме Гантта

Для создания данной диаграммы выполните следующие действия:

1. Введите данные, как это показано на рис. 17.5. В ячейке D2 находится формула, которая скопирована в расположенные ниже строки:

$$=B2+C2-1$$

2. С помощью Мастера диаграмм создайте линейчатую диаграмму на основе диапазона A2:C13. Используйте второй подтип, обозначенный как *линейчатая диаграмма с накоплением*.
3. На втором этапе мастера диаграмм выберите переключатель столбцах. Обратите внимание на то, что Excel для подписей осей ошибочно использует два первых столбца.

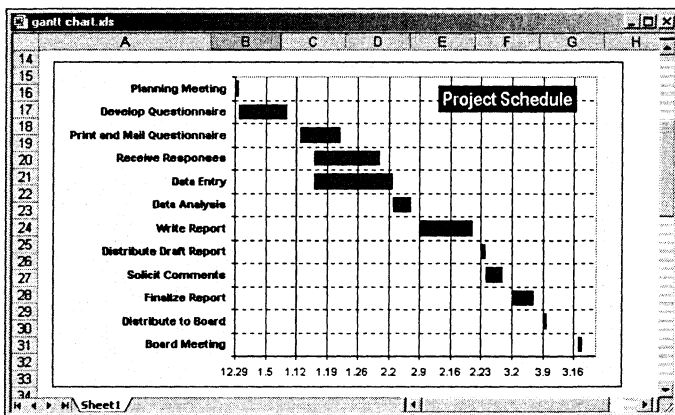


Рис. 17.6. Диаграмма Гантта строится на основе линейчатой диаграммы

4. На втором этапе мастера диаграмм щелкните на вкладке **Ряд** и добавьте новые ряды данных. Укажите ряды диаграмм следующим образом:
 Ряд1: B2 : B13
 Ряд2: C2 : C13
 Подписи оси X: A2 : A13
5. На третьем этапе мастера диаграмм удалите легенду, а затем щелкните на кнопке **Готово** для создания встроенной диаграммы.
6. Измените высоту диаграммы таким образом, чтобы стали видны все названия осей. Этого же можно добиться за счет использования шрифта меньшего размера.
7. Откройте диалоговое окно **Формат оси** для горизонтальной оси. Установите минимальное и максимальное значения в соответствии с самой ранней и самой поздней датами (даты можно вводить в полях минимальное значение и максимальное значение). Вы также можете изменить формат даты для подписей оси.
8. Откройте диалоговое окно **Формат оси** для вертикальной оси и щелкните на вкладке **Шкала**. Установите флажок **обратный порядок значений**, а также опцию **Пересечение с осью X (категорий)** в максимальном значении.
9. Выберите первый ряд данных и откройте диалоговое окно **Формат ряда данных**. На вкладке **Вид** в списках установите **невидимую границу** и **прозрачную заливку**. Это сделает невидимым первый ряд данных.
10. Примените дополнительное форматирование по своему усмотрению.

Создание гистограммы сравнения

Затратив самую малость творческих усилий, можно создавать диаграммы, построить которые с помощью Excel, казалось бы, просто невозможно. Примером может служить приведенная на рис. 17.7 гистограмма сравнения. Такие диаграммы, известные также как *пирамиды заселенности*, часто используются для отображения данных о популяции видов животных в определенных регионах планеты.

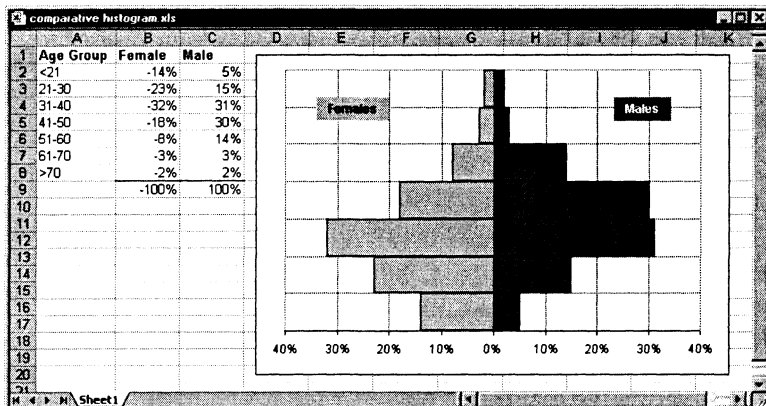


Рис. 17.7. Для построения данной диаграммы требуется знание дополнительных методов

Для создания диаграммы выполните следующие действия:

1. Введите данные, как показано на рис. 17.7. Обратите внимание, что значения для женских особей популяции указаны со знаком “минус”.
2. Выделите диапазон A1 : C8 и создайте двумерную линейчатую диаграмму. Используйте подтип, обозначенный как *нормированная линейчатая диаграмма*.
3. Для горизонтальной оси примените следующий пользовательский формат:
0%; 0%; 0%

При использовании данного форматирования удаляется знак минус перед процентными значениями.

4. Выберите вертикальную ось и откройте диалоговое окно *Формат оси*. Щелкните на вкладке *Вид* и уберите все деления. В группе меток делений выберите переключатель *внизу*. При этом ось размещается по центру диаграммы, а метки отображаются слева от нее.
5. Выберите любой из рядов данных и затем откройте диалоговое окно *Формат ряда данных*. Установите значение в поле *Перекрывание* равным 100, а в поле *Ширина зазора* — 0.
6. Удалите легенду.
7. Добавьте на диаграмму вместо легенды два текстовых поля (*Female* и *Male*).
8. Примените дополнительное форматирование по своему усмотрению.

Создание диаграммы разброса данных

Графики разброса данных часто используются для отображения итоговых результатов. На рис. 17.8 показана диаграмма разброса, созданная для четырех наборов данных. Строки данных находятся в столбцах от А до D. Диапазон G2 : J7, используемый в диаграмме, содержит формулы обобщения данных. В табл. 17.1 показаны формулы, записанные в столбце G (которые скопированы в три столбца вправо).

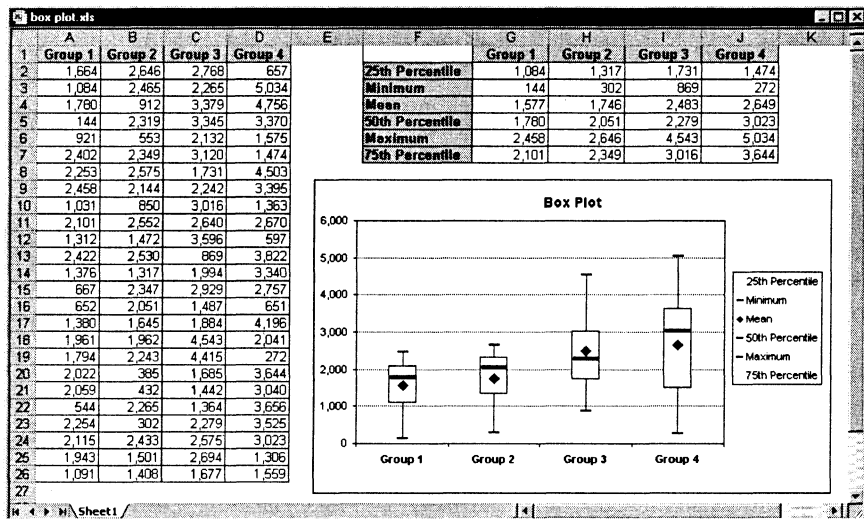


Рис. 17.8. На данном графике обобщены данные столбцов от А до D

Таблица 17.1. Формулы, используемые для создания диаграммы разброса данных

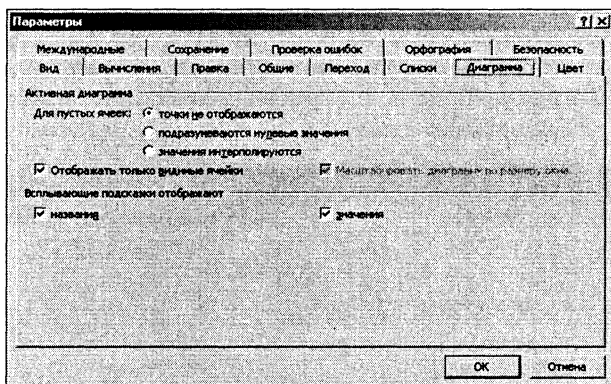
Ячейка	Результат	Формула
G2	25 процентов:	=КВАРТИЛЬ (A2 : A26 ; 1)
G3	Минимум	МИН (A2 : A26)
G4	Среднее значение:	СРЗНАЧ (A2 : A26)
G5	50 процентов	=КВАРТИЛЬ (A2 : A26 ; 2)
G6	Максимум	МАКС (A2 : A26)
G7	75 процентов	=КВАРТИЛЬ (A2 : A26 ; 3)

Управление отсутствующими данными?

В ряде случаев в данных, на основе которых строится диаграмма, может отсутствовать одна или несколько точек. Excel предлагает несколько вариантов решения ситуации с недостающими данными. В диалоговых окнах Формат ряда данных и Параметры диаграммы отсутствуют средства управления отображением отсутствующих данных. Необходимо выделить диаграмму и выбрать команду Сервис→Параметры, а затем щелкнуть на показанной ниже вкладке Диаграмма.

Данные установки применяются лишь к активной диаграмме. При открытии диалогового окна Параметры диаграмма должна быть активна, в противном случае настройка параметров будет недоступна. Это замечательный пример того, как настройка параметров может выполняться в самом неожиданном месте.

Выбранные настройки применяются ко всей диаграмме. Нельзя выбрать различные варианты для разных рядов той же диаграммы. Ниже приведено описание переключателей группы Активная диаграмма:



- ◆ не отображаются: отсутствующие данные игнорируются, в результате чего диаграмма терпит разрыв;
- ◆ подразумеваются нулевые значения: отсутствующие данные расцениваются как нулевые значения;
- ◆ значения интерполируются: значения для отсутствующих точек вычисляются на основе данных, находящихся по обе стороны от них. Данный вариант доступен лишь для графиков.

Для создания графика разброса данных выполните следующие действия:

1. Выделите диапазон F1 : J7.
2. Щелкните на кнопке Мастер диаграмм.
3. В первом окне мастера диаграмм выберите тип диаграммы График и четвертый подтип (График с маркерами). Щелкните на кнопке Далее.
4. На втором этапе мастера диаграмм выберите переключатель строках. Щелкните на кнопке Готово для создания диаграммы.
5. Активизируйте первый ряд данных (25 процентов), откройте диалоговое окно Формат ряда данных и щелкните на вкладке Вид. В группе Линия выберите переключатель отсутствует, в группе Маркер выберите переключатель отсутствует. Щелкните на вкладке Параметры и установите флажки Минимум-максимум и Открытие-закрытие. При желании перенастройте цвета.
6. Активизируйте второй ряд данных (минимум), откройте диалоговое окно Формат ряда данных и щелкните на вкладке Вид. В группе Линия выберите переключатель отсутствует, из списка типа маркера выберите маркер в виде горизонтальной черты. При желании перенастройте цвета.
7. Активизируйте третий ряд данных (среднее значение), откройте диалоговое окно Формат ряда данных и щелкните на вкладке Вид. В группе Линия выберите переключатель отсутствует, из списка тип маркера выберите маркер в виде ромба. При желании настройте цвета.
8. Активизируйте четвертый ряд данных (50 процентов), откройте диалоговое окно Формат ряда данных и щелкните на вкладке Вид. В группе Линия выберите переключатель отсутствует, из списка тип маркера выберите маркер в виде горизонтальной черты. При желании перенастройте цвета.

9. Активизируйте пятый ряд данных (максимум), откройте диалоговое окно **Формат** ряда данных и щелкните на вкладке **Вид**. В группе **Линия** выберите переключатель **отсутствует**, из списка **тип маркера** выберите маркер в виде горизонтальной черты. При желании перенастройте цвета.
10. Активизируйте шестой ряд данных (75 процентов), откройте диалоговое окно **Формат** ряда данных и щелкните на вкладке **Вид**. В группе **Линия** выберите переключатель **отсутствует**, в группе **Маркер** выберите переключатель **отсутствует**. При желании перенастройте цвета.



После того как все эти операции выполнены, вы можете создать пользовательский тип диаграмм для упрощения процедуры построения новых диаграмм этого же типа. Выделите диаграмму и затем выберите команду **Диаграмма** ⇒ **Тип диаграммы**. Щелкните на вкладке **Нестандартные** и выберите переключатель **дополнительные**. Щелкните на кнопке **Добавить** и определите **Название** и **Описание** для вашей диаграммы.

Построение каждой n-ой точки данных

Обычно Excel не строит точки, находящиеся в скрытых строках или столбцах. Иногда данное свойство можно использовать в своих целях, так как таким образом можно легко управлять видом диаграмм.

Предположим, что в столбце имеется огромное количество данных, и вы хотите отобразить на диаграмме каждое десятое значение. Один из способов решения данной задачи состоит в совместном использовании средства **Автофильтр** и формулы. На рис. 17.9 представлен рабочий лист, на котором показана в действии работа инструмента **Автофильтр**. На диаграмме отображаются лишь значения, находящиеся в видимых (отфильтрованных) строках, а значения, содержащиеся в скрытых строках, игнорируются.

Ячейка A1 содержит значение 10. Значение в данной ячейке определяет, какие из строк необходимо скрыть. Ячейки столбца B содержат идентичные формулы, использующие значение в ячейке A1. Например, формула в ячейке B3 имеет следующий вид:

=ОСТАТ(СТРОКА(); \$A\$1)

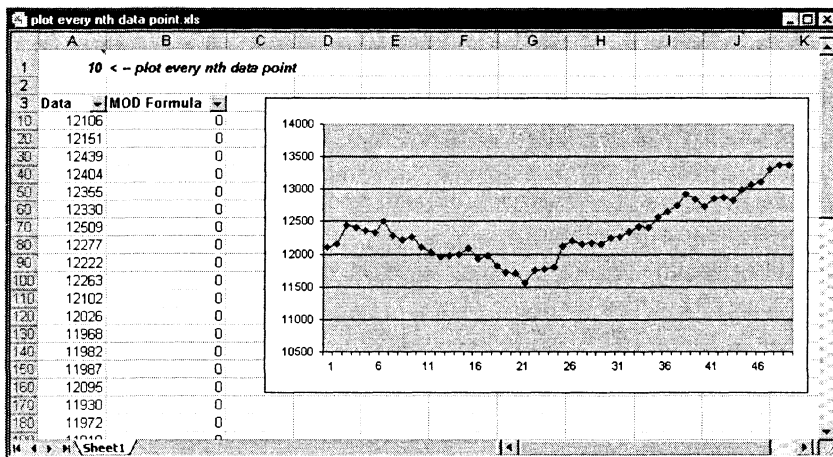


Рис. 17.9. На данной диаграмме отображается каждая десятая точка (как определено в A1) и с помощью средства **Автофильтр** игнорируются данные в скрытых строках

Формула использует функцию ОСТАТ для вычисления остатка деления номера строки (возвращаемого функцией СТРОКА) на значение в ячейке A1. В результате каждая n-ая ячейка (значение ячейки A1 определяет n) содержит 0. Далее для включения средства Автофильтр выберите команду Данные⇒Фильтр⇒Автофильтр. Настройте средство Автофильтр так, чтобы он отображал лишь те строки, для которых в столбце B находится значение 0. Примите во внимание, что в случае изменения значения в ячейке A1 необходимо повторно определить критерий для столбца B (строки не будут скрываться автоматически).

Предыдущая формула использует номер строки для определения тех ячеек, которые должны отображаться. Если нужно, чтобы на диаграмме всегда отображалась первая точка данных, воспользуйтесь следующей формулой, ссылающейся на ячейку (A4), содержащую первую точку данных:

=ОСТАТ(СТРОКА()-СТРОКА(\$A\$4);\$A\$1)



В некоторых случаях, вам может не понравиться то, что скрытые данные не отображаются на диаграмме. Для устранения такой ситуации активизируйте диаграмму и выберите команду Сервис⇒Параметры. В диалоговом окне Параметры щелкните на вкладке Диаграмма и сбросьте флажок Отображать только видимые ячейки.

Автоматическое обновление рядов данных

Изменить диапазон данных, используемых диаграммой нетрудно, однако в некоторых случаях вы можете захотеть, чтобы при вводе новых данных диаграмма обновлялась автоматически. Если, например, имеется диаграмма, отображающая ежедневные продажи, диапазон данных для нее должен, вероятно, изменяться каждый день при добавлении новых данных. В данном подразделе описано то, как заставить Excel всегда при введении новых данных на рабочий лист обновлять диапазон данных диаграммы.

Для того чтобы заставить Excel автоматически обновлять диаграмму при добавлении новых данных, необходимо выполнить следующие действия:

1. Создайте рабочий лист, показанный на рис. 17.10.
2. Выберите команду Вставка⇒Имя⇒Присвоить для вызова диалогового окна Присвоение имени. В поле Имя введите Data, а в поле Формула введите формулу следующего вида:

=СМЕЩ(Лист1!\$A\$2;0;0;СЧЕТЗ(Лист1!\$A:\$A)-1)

3. Щелкните на кнопке Добавить. Обратите внимание, что функция СМЕЩ ссылается на первую точку данных (A2) и использует функцию СЧЕТЗ для определения количества точек данных в столбце. Так как в первой строке столбца A находится заголовок, из полученного числа вычитается 1.
4. В поле Имя введите Sales, а в поле Формула введите формулу следующего вида:

=СМЕЩ(Лист1!\$B\$2;0;0;СЧЕТЗ(Лист1!\$B:\$B)-1)

5. Затем щелкните на кнопках Добавить и ОК для закрытия диалогового окна.
6. Активизируйте диаграмму и выберите ряды данных.
7. Ссылки на диапазоны замените именами, определенными на втором и четвертом этапах. Формула должна иметь следующий вид:

=РЯД(;Лист1!Data;Лист1!Sales;1)

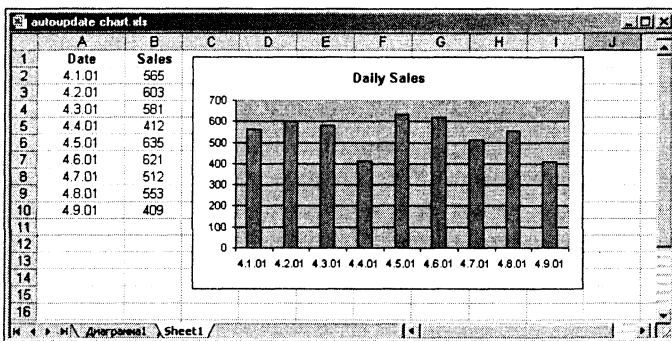


Рис. 17.10. Данная диаграмма обновляется автоматически при добавлении новых данных в столбцы А и В

После выполнения описанных действий, данная диаграмма будет автоматически обновляться при добавлении новых данных в столбцах А и В.



При использовании данной методики при построении диаграмм для своих данных проследите за тем, чтобы функция СМЕЩ ссылалась на первую точку данных, а аргумент функции СЧЕТЗ указывал на весь столбец данных. Кроме того, если используемый для данных столбец будет содержать и другие записи, функция СЧЕТЗ будет возвращать неверное значение.

Отображение n последних точек данных

Существует техника, позволяющая сделать так, чтобы на диаграмме отображались лишь свежие данные. Например, можно создать диаграмму, которая всегда отображает данные последних 12 месяцев (рис. 17.11).

Ниже приведено описание метода создания диаграммы, показанной на рис. 17.11.

1. Создайте рабочий лист, подобный тому, что показан на рис. 17.11.
2. Выберите команду Вставка⇒Имя⇒Присвоить для вызова диалогового окна Присвоение имени. В поле Имя введите Data, а в поле Формула введите формулу следующего вида:

$$=СМЕЩ(Лист1!A1;СЧЕТЗ(Лист1!$A:$A)-12;0;12;0)$$
3. Щелкните на кнопке Добавить. Обратите внимание, что функция СМЕЩ ссылается на ячейку A1 (не являющуюся ячейкой с первым месяцем).
4. В поле Имя введите Sales, а в поле Формула введите формулу следующего вида:

$$=СМЕЩ(Лист1!B1;СЧЕТЗ(Лист1!$B:$B)-12;0;12;1)$$
5. Щелкните на кнопках Добавить и ОК для закрытия диалогового окна.
6. Активируйте диаграмму и выберите ряды данных.
7. Ссылки на диапазоны замените именами, определенными на втором и четвертом этапах. Формула должна выглядеть следующим образом:

$$=РЯД(;Лист1!Data;Лист1!Sales;1)$$

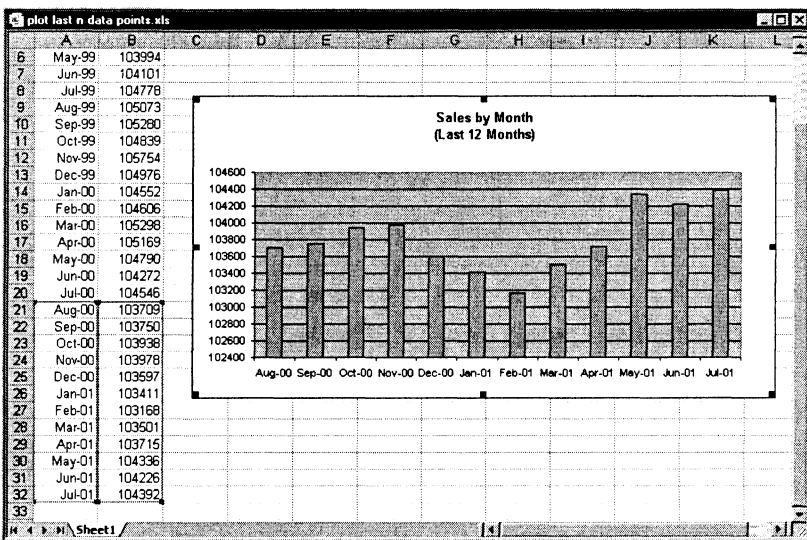


Рис. 17.11. Данная диаграмма отображает 12 последних точек данных



Чтобы построить диаграмму для другого количества точек, измените формулы, записанные на втором и четвертом этапах. В обоих случаях замените значение 12 на другое.

Интерактивное отображение данных

В данном разделе описываются две методики, которые могут использоваться для извлечения максимума пользы из одной диаграммы. Вы увидите, как пользователь может определять данные, отображаемые на диаграмме либо в результате выбора строки, либо указания элемента из раскрывающегося списка.

Построение диаграммы на основе выбора строки

На рис. 17.12 показана диаграмма, отображающая данные, находящиеся в той строке, в которой выделена ячейка. После перемещения маркера выделения ячейки нажмите клавишу <F9> для отображения данных строки, соответствующей его новому положению.

Диаграмма использует две именованных формулы, каждая из которых имеет смешанные ссылки (ссылка на столбцы абсолютна, на строки — относительна). При создании следующих имен предполагается, что ячейка A3 была активной. Формула *ChartTitle* определена так:

=СМЕЩ(\$A3;0;0)

Формула *ChartData* определена так:

=СМЕЩ(\$A3;0;1;5)

Формула РЯД для ряда данных диаграммы использует эти две именованные формулы. Формула РЯД имеет следующий вид:

=РЯД(Лист1!ChartTitle;Лист1!\$B\$2:\$F\$2;Лист1!ChartData;1)

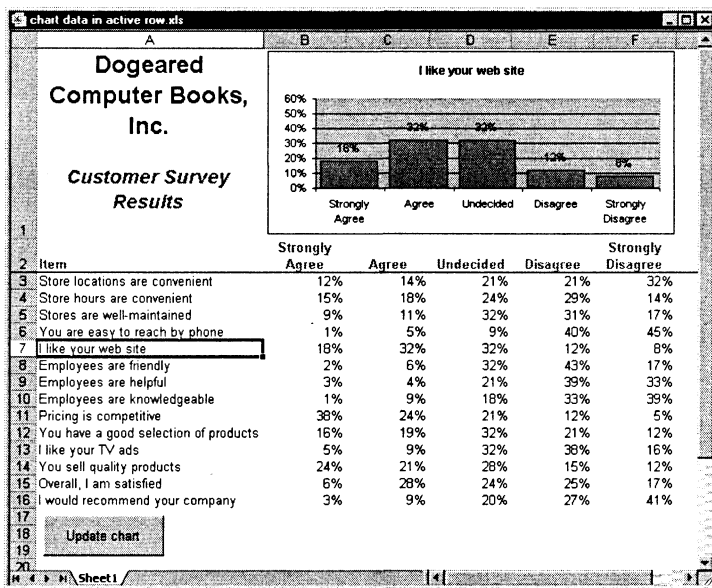


Рис. 17.12. При нажатии клавиши <F9> отображаются данные, содержащиеся в той строке, в которой выделена ячейка

При пересчете рабочего листа именованные формулы обновляются в соответствии с активной ячейкой.

Рабочий лист содержит кнопку, запускающую простой макрос VBA, определяющий находится ли маркер выделения ячейки в строке, содержащей данные (другими словами, в строках с 3 по 16). Если это так, лист вычисляется. Если нет, ничего не происходит. Ниже приведен листинг макроса:

```
Sub UpdateChart()
    If ActiveCell.Row > 2 And ActiveCell.Row < 17 Then _
        ActiveSheet.Calculate
End Sub
```

Выбор данных из списка

На рис. 17.13 показана диаграмма, отображающая данные, определяемые с помощью раскрывающегося списка. Диаграмма использует данные из диапазона B1:E2, однако содержимое ячеек данного диапазона определяется месяцем, выбранным в раскрывающемся списке. В диапазоне A6:D17 содержатся представляемые графически данные за указанный месяц, а формулы в диапазоне B1:E2 отображают данные, соответствующие значению, приведенному в ячейке A2. Например, когда ячейка A2 содержит значение 4, диаграмма отображает данные за апрель (четвертый месяц).

В ячейке B2 содержится следующая формула:

=ИНДЕКС (A6 : A17 ; \$A\$2)

Данная формула скопирована в диапазон C2 : E2.

Ключевой момент — сделать так, чтобы в раскрывающемся списке отображались названия месяцев, а номер месяца помещался в ячейку A2. Чтобы создать раскрывающийся список, выполните следующие действия:

1. Выберите команду Вид⇒Панели инструментов⇒Формы для отображения панели Формы.
2. На панели Формы щелкните на элементе, обозначенном как Поле со списком, и перетащите его на рабочий лист для создания элемента управления.

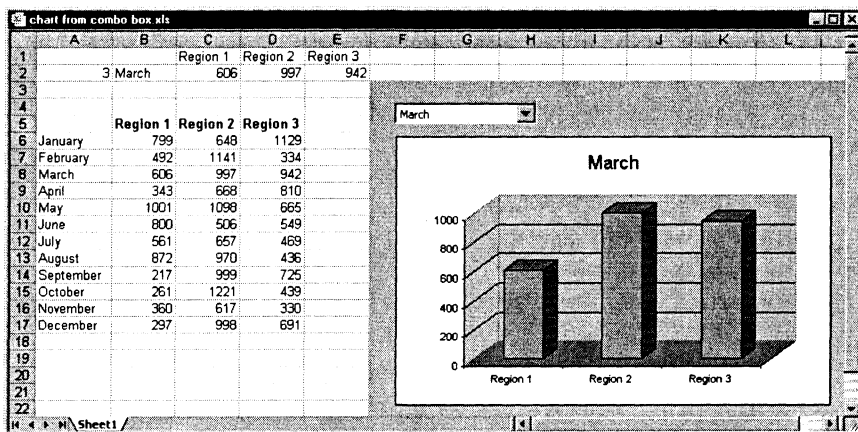


Рис. 17.13. Выбор данных для построения диаграммы на основе выбора в списке

3. Дважды щелкните на элементе Поле со списком для отображения диалогового окна Формат элемента управления.
4. Щелкните на вкладке Элемент управления данного диалогового окна.
5. В поле Формировать список по диапазону определите диапазон A6:A17, а в поле Связь с ячейкой укажите ячейку A2.

После этого в раскрывающемся списке будут отображаться названия месяцев, а номер выбранного месяца будет помещаться в ячейку A2. Формулы в строке 2 будут представлять данные, отображаемые на диаграмме.

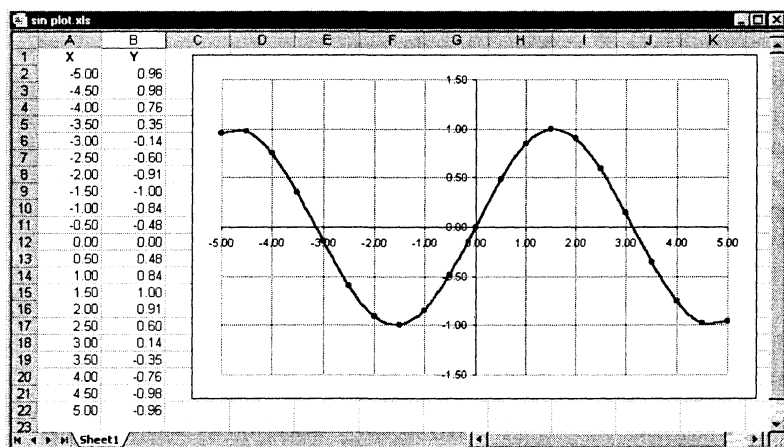


Рис. 17.14. На данной диаграмме построен график функции SIN(x)

Построение графиков функций с одной переменной

Средство построения диаграмм позволяет создавать графики различных математических и тригонометрических функций. Для примера на рис. 17.14 показан график функции SIN для переменной x (выраженной в радианах) в диапазоне значений от -5 до $+5$ с шагом $0,5$. Функция имеет следующий вид:

$$y = \text{SIN}(x)$$

Приведенная диаграмма относится к типу графиков с маркерами, в которой значения x представлены в столбце А, а значения y — в столбце В. Каждая пара значений x и y представляют собой точку данных на диаграмме, сами же точки соединены линией.

Создание графиков

Построение графиков базируется на использовании двух диапазонов данных: одного — для значений x , другого — для значений y . Последние генерируются формулой.



При построении диаграмм функций убедитесь, что используется график с маркерами точек. Для любого другого типа диаграмм Excel всегда использует одинаковое расстояние между значениями оси x .

Лучший способ построения графиков функций

Стефан Буллен (Stephen Bullen), весьма неординарный специалист по использованию Excel, разработал свою собственную методику, которой теперь можете воспользоваться и вы. Она позволяет автоматически строить графики функций или формул, исключая генерирование дополнительных значений на рабочем листе! Это одно из наиболее впечатляющих применений Excel, которое мне когда-либо приходилось видеть, и я благодарен Стефану, позволившему использовать данный материал в этой книге.

На примере, приведенном на рис. 17.15, для 25 значений x в диапазоне от -5 до $+5$ был построен график следующей функции:

$$y = (x^3) * (x^2)$$

В методе Стефана используются две именованные функции: X и Y . В данной диаграмме формула РЯД использует эти имена и имеет следующий вид:

=РЯД(; Лист1!X; Лист1!Y; 1)

Для того чтобы построить такой график:

1. В ячейке В7 введите формулу *в виде текста*.
2. В ячейке С10 введите начальное значение переменной x .
3. В ячейке С11 введите конечное значение для переменной x .
4. В ячейке С12 укажите число используемых при построении точек.

Итак, как она работает? Начнем с анализа формулы X , генерирующей значения x для ряда диаграммы:

= $\$C\$10 + (\text{СТРОКА}(\text{СМЕЩ}(\$B\$1; 0; 0; \$C\$12; 1)) - 1) * (\$C\$11 - \$C\$10) / (\$C\$12 - 1)$

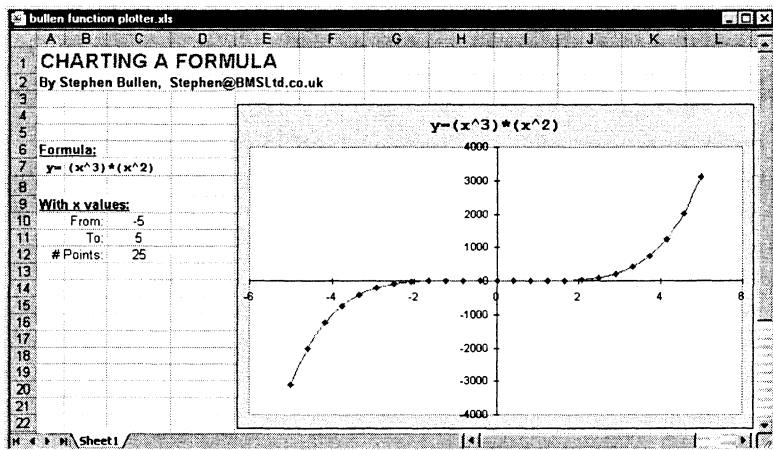


Рис. 17.15 Построение графика функции с помощью метода, разработанного Стефаном Булленом



Для проверки именованных формул в данной книге используйте команду Excel Вставка⇒Имя⇒Присвоить.

В данной формуле функция СМЕЩ используется для генерирования массива из n значений, где n определяется ячейкой C12. Массив начинается со значения в ячейке C10 и заканчивается значением в ячейке C11. Приведенное ниже выражение вычисляет прирост между двумя последовательными значениями x (вычисление производится путем вычитания начального значения из последнего и деления на количество точек минус 1):

$$= (\$C\$11 - \$C\$10) / (\$C\$12 - 1)$$

Например, если значения x начинаются с -3 и заканчиваются 3 , и имеется пять точек данных, будет создан следующий массив:

$$\{-3; -1,5; 0; 1,5; 3\}$$

Для генерирования массива значений y для диаграммы по формуле Y используется функция EVALUATE:

$$= \text{EVALUATE} (\$B\$7 + x * 0)$$

EVALUATE является макрофункцией XLM и не может быть использована в формулах рабочего листа. Данная функция, по существу, оценивает строчное выражение и возвращает результат. Хотя функцию EVALUATE нельзя использовать в функциях рабочего листа, ее можно использовать в имени.

Предположим, ячейка B7 содержит строку SIN(x), а значения x начинаются с -3 . Формула Y для первого значения будет иметь следующий вид:

$$= \text{EVALUATE} (\$B\$7 + x * 0)$$

Упрощенный аргумент функции: SIN(X) + $X * 0$



Часть формулы $+x*0$ указывает на то, что ее результат будет числовым.

Формула EVALUATE принимает первое значение (-3) из массива, генерируемого формулой X, оценивает следующее значение и возвращает результат как первое значение у:

$$\text{SIN}(-3) - 3 * 0$$

Рабочая книга содержит еще одну формулу в ячейке G8, скрытой диаграммой. Название диаграммы связано с данной ячейкой, содержащей следующую формулу:

$$\text{A7\&B7\&ТЕХТ(ТДАТА() ; " ")}$$

Эта формула объединяет ячейки A7 и B7 и использует функцию ТДАТА для принудительного обновления диаграммы при изменении любой из данных ячеек.

Построение графиков функций с двумя переменными

В предыдущем разделе было описано построение графиков функций с одной переменной. Вы, например, можете теперь построить график следующей функции для различных значений x:

$$y = x^2$$

Вы также имеете возможность построения графиков функций с двумя переменными. Следующая функция, например, вычисляет значение z для различных значений двух переменных (x и y):

$$z = \text{SIN}(x) * \text{COS}(y)$$

На рис. 17.16 показана поверхность, описываемая значением z для значений x, изменяющихся в диапазоне значений от -3,0 до 0 и значений y, изменяющихся в диапазоне значений от 2,0 до 5,0. Для обеих переменных используется шаг изменения значения, равный 0,15.

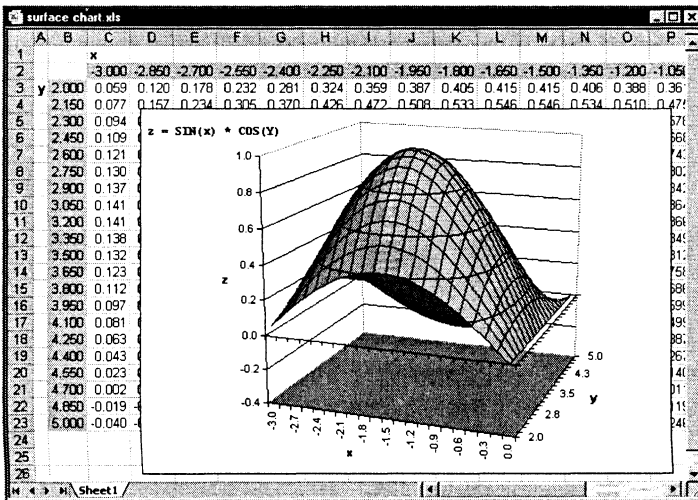


Рис. 17.16. Использование поверхностной диаграммы для графического отображения функции двух переменных

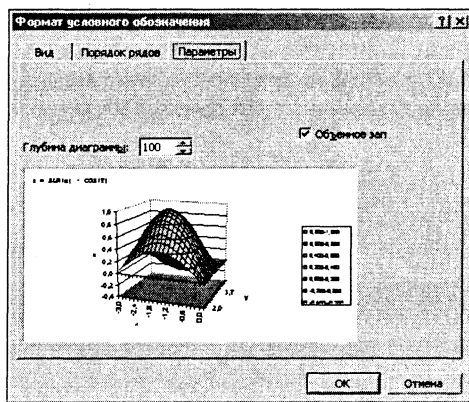
Если вам приходилось работать с диаграммами типа Поверхность, вы, наверняка, заметили, что данному типу диаграмм присущи некоторые серьезные ограничения. В идеале, хотелось бы иметь диаграмму поверхности в пространстве XYZ, для которой можно вносить различные значения X, Y и Z. К сожалению, данный тип диаграмм Excel не поддерживает.

Поверхностные диаграммы в Excel по существу отображают трехмерный вид чего-то, выглядящего как кусок резины, натянутый на трехмерную гистограмму. Пример, приведенный на рис. 17.16 содержит 21 ряд данных (отвечающий значениям y), каждый из которых содержит 21 точку данных (отвечающих значениям x).

“Секретные” приемы форматирования поверхностных диаграмм

Возможно, вы обратили внимание, что Excel не позволяет выделить отдельный ряд данных поверхностной диаграммы. Вследствие этого применять форматирования, обычно доступные в диалоговом окне Формат ряда данных, вы не сможете.

Некоторые методы форматирования поверхностных диаграмм все же можно использовать, но для того, чтобы добраться до нужного диалогового окна — Формат условного обозначения (см. приведенный рисунок), потребуется немного побродить лабиринтами настроек Excel. Чтобы открыть данное диалоговое окно, убедитесь, что диаграмма содержит легенду. Затем щелкните на легенде для ее выделения, после чего щелкните на любой из подписей легенды (цветном маркере, находящемся слева от элемента легенды). Дважды щелкните на выбранном маркере и вы окажетесь в диалоговом окне Формат условного обозначения.



- ◆ Используйте вкладку Вид для изменения цвета маркера легенды; при этом также изменяется цвет соответствующего ряда данных. Если вы хотите, чтобы диаграмма поверхности была одноцветной, нужно изменить каждый из маркеров.
- ◆ Используйте вкладку Параметры для изменения глубины диаграммы. Глубина может быть изменена для *любого* выбранного маркера диаграммы.
- ◆ На вкладке Параметры можно также применить к диаграмме трехмерную тень (Объемное зап). Опять-таки, данная установка влияет на всю диаграмму, а не только на ряд данных, соответствующий выбранному маркеру диаграммы.

При создании выразительной трехмерной поверхностной диаграммы необходимо начать с двухмерного диапазона, оставив пустой верхнюю левую ячейку. Верхняя строка должна содержать увеличивающиеся или уменьшающиеся значения x с постоянной разницей между каждым из них. Левый столбец должен содержать увеличивающиеся или уменьшающиеся значения y с постоянной разницей между каждым из них. Значения z заполняют оставшиеся ячейки в соответствии с парами значений x и y . В качестве исходных данных для диаграммы выберите полный диапазон.

Создание экстравагантных проектов

На рис. 17.17 приведен пример поверхностной диаграммы, отображающей “типоциклоидные” кривые, построенные на основе случайных значений.

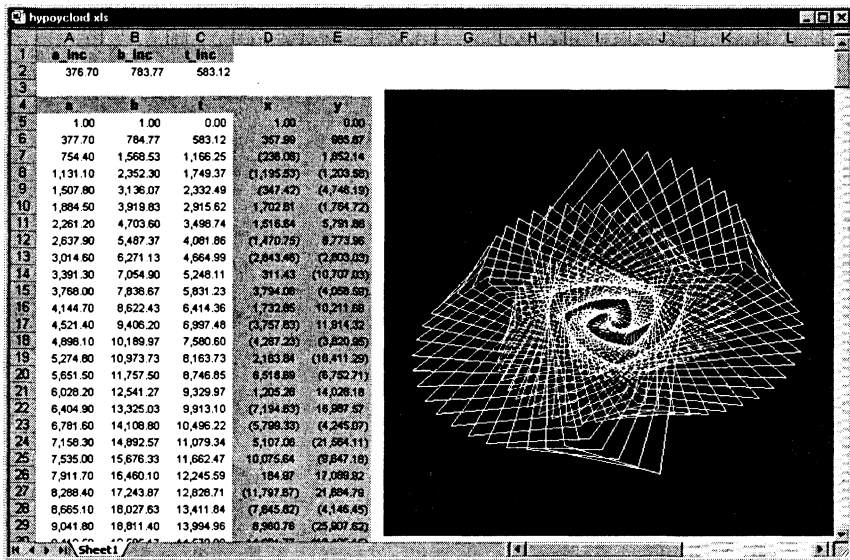


Рис. 17.17. Гипоциклоидные кривые

Диаграмма использует данные в столбцах D и E (диапазоны x и y). В данных столбцах находятся формулы, зависящие от данных, содержащихся в столбцах от A до C. В первом столбце (столбец D) находится формула:

$$= (A5 - B5) * \text{COS}(C5) + B5 * \text{COS}((A5 / B5 - 1) * C5)$$

Формула во втором столбце (столбец E) имеет следующий вид:

$$= (A5 - B5) * \text{SIN}(C5) + B5 * \text{SIN}((A5 / B5 - 1) * C5)$$

При нажатии клавиши <F9> выполняется пересчет листа; при этом генерируются новые возрастающие значения (случайные) для столбцов от A до C и создается новое изображение на диаграмме. Разнообразие (и красота) диаграмм, генерируемых данной формулой, не оставит вас равнодушными.

Диаграммы тенденций

В некоторых диаграммах для более наглядного описания данных может потребоваться построение графика тенденции. График тенденции отображает общие тенденции в имеющихся данных. В некоторых случаях с помощью тенденций могут предсказываться будущие данные. Один ряд данных может иметь несколько тенденций.

Добавление тенденции на диаграмму производится в Excel довольно просто. Можно ожидать, что доступ к данному средству будет осуществляться в диалоговом окне Формат ряда данных, однако это не так. Вы будете использовать диалоговое окно Линия тренда, показанное на

рис. 17.18, доступ к которому осуществляется с помощью команды **Диаграмма**⇒**Добавить** линию тренда. Данная команда доступна лишь тогда, когда ряд дынных выделен.

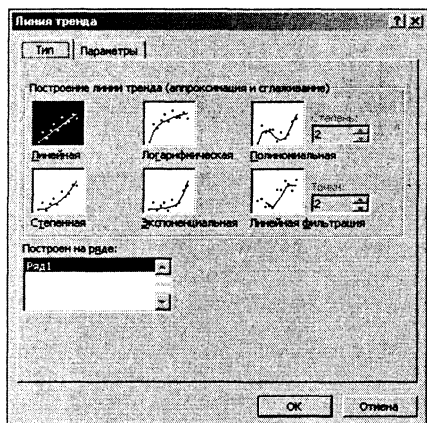


Рис. 17.18. В диалоговом окне **Линия тренда** предлагается несколько типов автоматических созданных тенденций

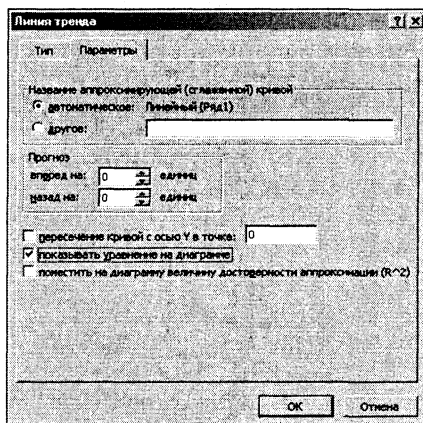


Рис. 17.19. Вкладка **Параметры** диалогового окна **Линия тренда**

То, какой тип тенденции выбрать, зависит от конкретных данных. Чаще всего используется линейный тип, однако для описания некоторых данных эффективнее может оказаться использование какого-то другого типа. При щелчке на вкладке **Параметры** диалогового окна **Линия тренда** вы получаете доступ к управлению параметрами, показанными на рис. 17.19.

Вкладка **Параметры** позволяет определить название, отображаемое в легенде, а также число периодов, на которое выполняется предсказание. Дополнительные параметры позволяют указать пересечение кривой с осью Y, представить на диаграмме уравнение, используемое для построения графика тенденции, а также выбрать, будет ли отображаться на диаграмме величина достоверности аппроксимации.

Вставляемая Excel линия тенденции может выглядеть как ряд данных, хотя, на самом деле, таковой не является. Это новый элемент диаграммы с названием, подобным **Линейный**. Безусловно, для нее не существует формулы **РЯД**. Для изменения форматирования или параметров линии тенденции необходимо дважды щелкнуть на ней.

Линейные тенденции

На рис. 17.20 показаны две диаграммы. Диаграмма, находящаяся слева, отображает ряд данных без графика тенденций. Как можно заметить, анализируется линейная зависимость изменения данных во времени. Справа изображена та же диаграмма, но уже с графиком тенденции, показывающей ход изменения данных.

Во второй диаграмме использованы опции отображения уравнения и величины достоверности аппроксимации. В данном примере уравнение имеет вид:

$$y = 53,194x + 514,93$$

Величина достоверности аппроксимации составляет 0,6748.



Для отображения большего или меньшего числа десятичных знаков выберите в формуле и области аппроксимации щелкните на кнопке **Увеличение разрядности** или **Уменьшение разрядности** панели инструментов **Форматирование**.

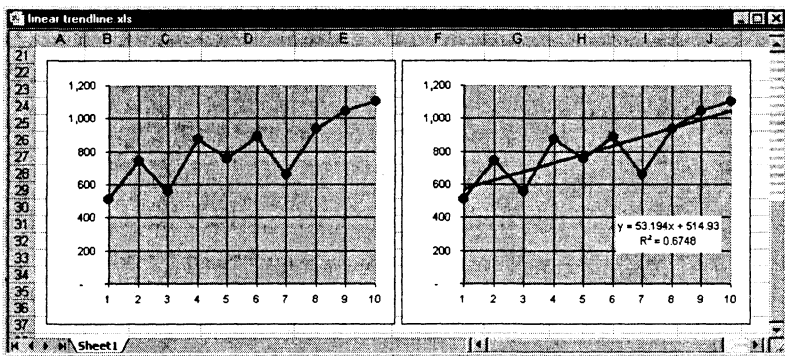


Рис. 17.20. Диаграмма до (левая диаграмма) и после (правая диаграмма) добавления на нее графика тенденции

Что обозначают все эти числа? Прямая линия описывается уравнением следующего вида:

$$y = mx + b$$

С помощью данного уравнения для каждого значения x (в данном случае это столбец В) можно получить расчетное значение y (значение на графике тенденции). Переменная m представляет угол наклона прямой, а b является точкой пересечения прямой с осью Y . Например, для февраля значение x равно 2, а значение y — 743. Расчетное значение, полученное с помощью следующей формулы, составило 621,318:

$$= (53,194 * 2) + 514,93$$

Величина достоверности аппроксимации, называемая еще *коэффициентом соответствия*, может меняться в пределах от 0 до 1. Данное значение показывает, на сколько близки расчетные данные к реальным. Линия тренда наиболее достоверна, когда величина достоверности аппроксимации равна или близка к 1.

Вычисление угла наклона и точки пересечения с осью Y

Как вы знаете, Excel может отображать на диаграмме уравнение графика тенденции. Данное уравнение описывает угол наклона (m) и точку пересечения с осью Y (b). Вы можете вычислить данные величины самостоятельно с помощью формулы, использующей функцию ЛИНЕЙН.

На рис. 17.21 показаны 10 точек данных (значения x находятся в столбце В, значения y — в столбце С).

1	Month	X	Actual Y				m	b
2	Jan	1	512				53.19394	514.9333
3	Feb	2	743					
4	Mar	3	559					
5	Apr	4	875					
6	May	5	755					
7	Jun	6	890					
8	Jul	7	663					
9	Aug	8	934					
10	Sep	9	1 042					
11	Oct	10	1 102					
12	Nov	11						
13	Dec	12						

Рис. 17.21. Использование функции ЛИНЕЙН для вычисления угла наклона и точки пересечения с осью Y

Приведенная ниже формула является формулой массива, отображающей свой результат в двух ячейках:

`{=ЛИНЕЙН(C2:C11;B2:В11)}`

Чтобы ввести данную формулу, начните с выделения двух ячеек (в данном примере G2:H2). Затем введите формулу (без скобок) и нажмите сочетание клавиш <Ctrl+Shift+Enter>. В ячейке G2 будет отображен угол наклона, а в ячейке H2 — точка пересечения с осью Y.

Нахождение расчетных значений

После того как определены угол наклона и точка пересечения с осью Y, можно найти расчетное значение y для произвольного значения x. На рис. 17.22 показан результат. В ячейке E2 содержится формула, которая была скопирована вниз по столбцу:

`=(B2*G2)+H2`

Month	X	Actual Y	Predicted Y	Predicted Y	Forecasted	m	b
Jan	1	512	568,127	568,1273		53,19394	514,9333
Feb	2	743	621,321	621,3212			
Mar	3	559	674,515	674,5152			
Apr	4	875	727,709	727,7091			
May	5	755	780,903	780,9030			
Jun	6	890	834,097	834,0970			
Jul	7	663	887,291	887,2909			
Aug	8	934	940,485	940,4848			
Sep	9	1 042	993,679	993,6788			
Oct	10	1 102	1 046,873	1 046,8727			
Nov	11			1 100,0667			
Dec	12			1 153,2606			

Рис. 17.22. Столбец D содержит формулы, находящие расчетные значения y

Вычисленные значения в столбце E используются для построения линейного графика тенденции. Расчетные значения можно найти и без вычисления угла наклона и точки пересечения с осью Y. Тот же результат может быть получен с помощью формулы массива, использующей функцию ТЕНДЕНЦИЯ. Выделите диапазон D2:D11, после чего введите приведенную ниже формулу (без скобок) и нажмите сочетание клавиш <Ctrl+Shift+Enter>:

`{=ТЕНДЕНЦИЯ(C2:C11;B2:В11)}`

Линейное предсказание

Если диаграмма имеет тенденцию, Excel может предсказать и отобразить на диаграмме дополнительные точки. Данная возможность реализуется на вкладке Параметры диалогового окна Формат линии тренда (или на вкладке Параметры диалогового окна Линия тренда). Необходимо просто указать число предсказываемых точек. На рис. 17.23 представлена диаграмма, предсказывающая результат двух точек.

Если известны угол наклона и точка пересечения с осью Y (см. в одном из предыдущих подразделов “Вычисление угла наклона и точки пересечения с осью Y”), можно сделать прогноз для любых значений x. Например, чтобы вычислить значение y, при котором x=11 (ноябрь), воспользуйтесь следующей формулой:

$y = (53,194 * 11) + 514,93$

Можно предсказать значения с помощью функции ПРЕДСКАЗ. Вот пример формулы, прогнозирующей значение для ноября (т.е. для x=11), используя известные значения x и y:

`=ПРЕДСКАЗ(11;C2:C11;B2:В11)`

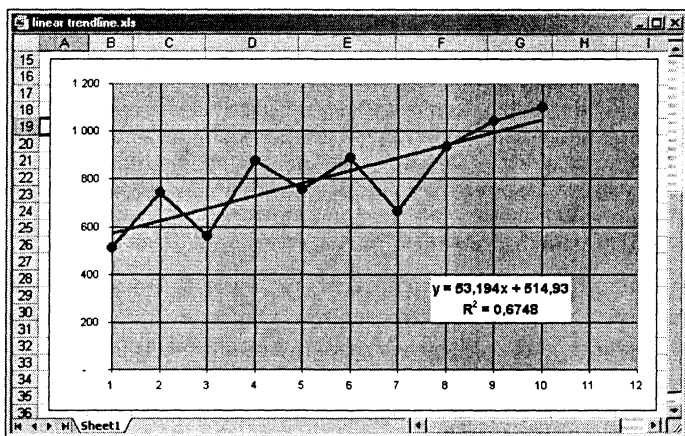


Рис. 17.23. Использование тенденции для предсказания двух дополнительных точек

Вычисление величины достоверности аппроксимации

Точность предсказываемых значений определяется тем, насколько хорошо линейная тенденция соответствует реальным данным. Величина достоверности аппроксимации показывает степень соответствия. Чем ближе данное значение к 1, тем выше соответствие и тем выше точность предсказания. Другими словами величину достоверности аппроксимации можно рассматривать как отношение расхождения значений y к расхождению значений x .

Как говорилось ранее, можно указать Excel отображать величину достоверности аппроксимации на диаграмме. Можно также посчитать данное значение самостоятельно с помощью функции КВПИРСОН. Приведенная ниже формула вычисляет величину достоверности аппроксимации для значений x , находящихся в диапазоне B1 : B11, и для значений y диапазона C1 : C11.

=КВПИРСОН (B1 : B11 ; C1 : C11)



Вычисление величины достоверности аппроксимации с помощью функции КВПИРСОН правомерно лишь в случае линейной тенденции.

Нелинейные тенденции

Аппроксимацией кривой называется процесс создания математического описания зависимости для нелинейного диапазона данных (экстраполяция) или произведение оценки полученных точек данных (интерполяция). Наряду с линейными тенденциями Excel может строить тенденции следующих типов:

- ◆ Логарифмическая — используется в случае, когда данные имеют высокую скорость возрастания или спада, а затем стабилизируются.
- ◆ Степенная — используется, когда данные увеличиваются или спадают с постоянной скоростью. Данные не могут содержать нулевых или отрицательных значений.
- ◆ Экспоненциальная — используется, когда значения увеличиваются или уменьшаются со все большей скоростью. Данные не могут содержать нулевых или отрицательных значений.

- ◆ Полиномиальная — используется в случае, когда данные неустойчивые. В зависимости от количества колебаний значений данных можно выбрать порядок полинома (от 2 до 6).



На вкладке Тип диалогового окна Линия тренда предоставляется также возможность выбрать тенденцию Линейная фильтрация, которая не является настоящей тенденцией. Данная возможность, однако, используется при сглаживании “зашумленных” данных. При выборе данной опции предоставляется возможность указания количества точек данных, которые будут включаться в каждое усреднение. Например, если выбрать 5 точек, Excel будет усреднять каждую группу из пяти точек данных.

Ранее в данной главе было описано, как найти угол наклона и точки пересечения с осью Y для линейного уравнения, описывающего линейную тенденцию. Нелинейные тенденции также имеют свои уравнения, о чем сказано в следующих подразделах.

Логарифмическая тенденция

Логарифмическая тенденция описывается уравнением следующего вида:

$$y = (c * \text{LN}(x)) - b$$

На рис. 17.24 показана диаграмма с добавленной логарифмической тенденцией. Одна формула массива, записанная в ячейках E2 : F2, вычисляет значения для переменных c и b:

{=ЛИНЕЙН(C2 : C11 ; LN(B2 : B11)) }

В столбце C содержатся расчетные значения y для каждого значения x, полученные на основе вычисленных значений b и c. Например, в ячейке C2 содержится следующая формула:

$$= (\$E\$2 * \text{LN}(A2)) + \$F\$2$$

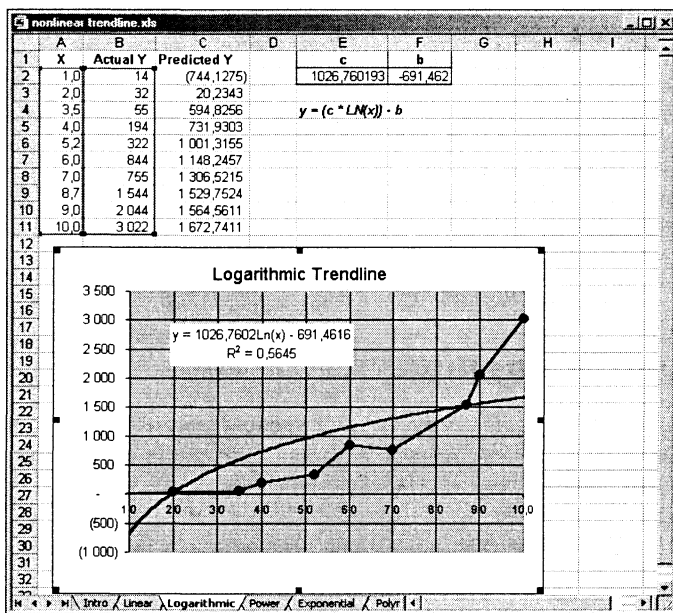


Рис. 17.24. Диаграмма, отображающая логарифмическую тенденцию

Как видно, логарифмическая тенденция не является хорошей аппроксимацией для имеющихся данных. Величина достоверности аппроксимации низкая, и график тенденции не совпадает с диаграммой данных.

Степенная тенденция

Степенная тенденция описывается уравнением следующего вида:

$$y = c * x^b$$

На рис. 17.25 показана диаграмма с добавленной степенной тенденцией. Первый элемент в двухъячейчной формуле массива, записанной в диапазоне E2:F2, вычисляет значения переменной b:

```
{=ЛИНЕЙН(LN(B2:B11);LN(A2:A11)); ;ИСТИНА)}
```

Значение переменной c вычисляется с помощью формулы, находящейся в ячейке F3:
=EXP(F2)

В столбце C содержатся расчетные значения y для каждого значения x, полученные на основе вычисленных значений b и c. Например, в ячейке C2 содержится следующая формула:

```
=(F$3*(A2^$E$2))
```

Экспоненциальная тенденция

Экспоненциальная тенденция описывается уравнением следующего вида:

$$y = c * \text{EXP}(b * x)$$

На рис. 17.26 показана диаграмма с добавленной экспоненциальной тенденцией. Первый элемент в двухъячейчной формуле массива, записанной в ячейках F2:G2, вычисляет значения для переменной b:

```
{=ЛИНЕЙН(LN(B2:B11);A2:A11)}
```

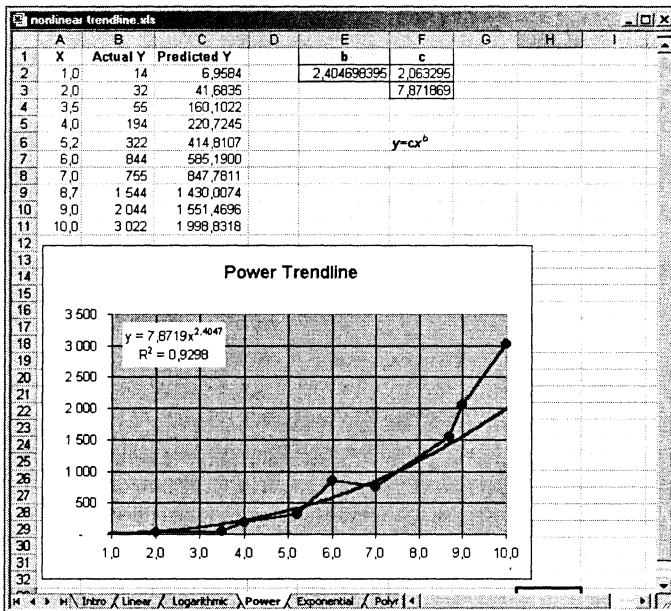


Рис. 17.25. Диаграмма, отображающая степенную тенденцию

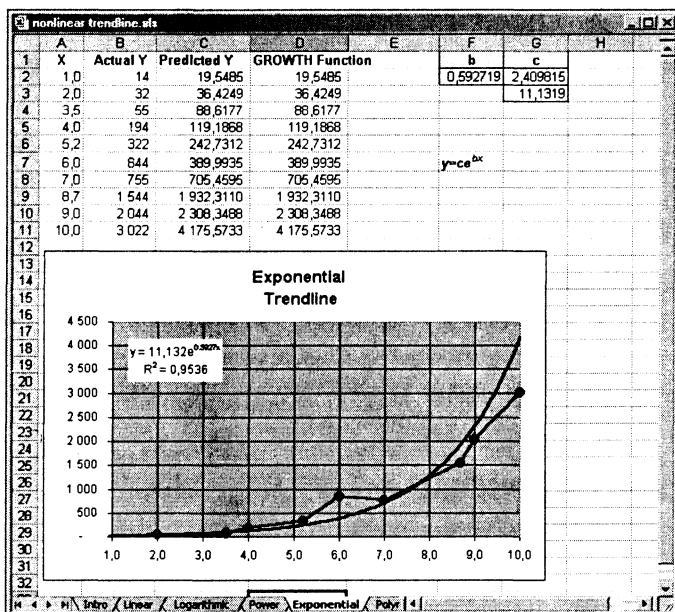


Рис. 17.26. Диаграмма, отображающая экспоненциальную тенденцию

Значение переменной c вычисляется с помощью формулы, находящейся в ячейке G3:
 $=\text{EXP}(G2)$

В столбце C содержатся расчетные значения y для каждого значения x , полученные на основе вычисленных значений b и c . Например, в ячейке C2 содержится следующая формула:
 $=\$G\$3*\text{EXP}(\$F\$2*A2)$

В столбце D в формуле массива используется функция РОСТ для генерирования расчетного значения y . Формула массива, записанная в ячейках D2:D10, имеет следующий вид:
 $\{\text{РОСТ}(B2:B11;A2:A11)\}$

Линейная фильтрация

При выборе тенденции Линейная фильтрация требуется также указать порядок полинома (изменяющийся от 2 до 6). Вид уравнения для данного типа тенденции зависит от порядка полинома. Ниже приведено уравнение для полиномиальной тенденции третьего порядка:

$$y = (c3 * x^3) + (c2 * x^2) + (c1 * x^1) + b$$

Обратите внимание на то, что имеется три коэффициента c (по одному для каждого порядка).

На рис. 17.27 показана диаграмма с добавленной полиномиальной тенденцией третьего порядка. Четырехэлементная формула массива, записанная в ячейках F2:I2, вычисляет значения каждого из трех коэффициентов c и коэффициента b :

$$\{=\text{ЛИНЕЙН}((B2:B11);(A2:A11)\wedge\{1;2;3\})\}$$

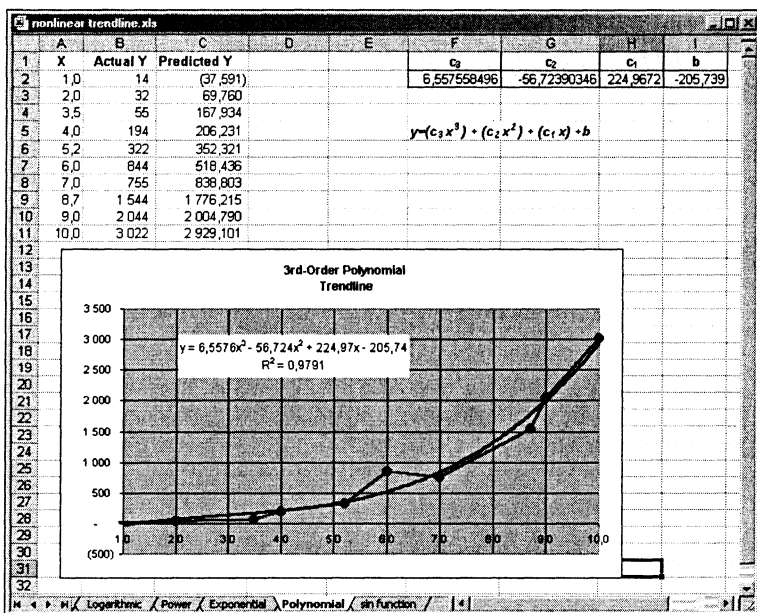


Рис. 17.27. Диаграмма, отображающая полиномиальную тенденцию

В столбце С содержатся расчетные значения y для каждого значения x , полученные на основе вычисленных коэффициента b и трех коэффициентов c . Например, в ячейке С2 содержится следующая формула:

$$= (\$F\$2 * A2^3) + (\$G\$2 * A2^2) + (\$H\$2 * A2) + \$I\$2$$

Полезные методы построения диаграмм

В данном разделе описан ряд полезных приемов построения диаграмм, которые собраны мною за несколько лет работы. Здесь вы найдете советы по сохранению нескольких диаграмм на одном листе, просмотру встроенных диаграмм в окне, изменению значений рабочего листа путем перетаскивания точек данных на диаграмме, а также по анимации диаграмм.

Сохранение нескольких диаграмм на одном листе

Большинство пользователей Excel согласится с утверждением, что лист содержит только одну диаграмму. Обычно так оно и есть. Однако существует возможность хранения на одном листе нескольких диаграмм. Фактически, Excel предоставляет возможность сделать это напрямую. Если выделить внедренную диаграмму, а затем выбрать команду **Диаграмма** → **Размещение**, будет открыто диалоговое окно **Размещение диаграммы**. Если выбрать переключатель **Именованная** и в качестве места расположения выбрать существующий лист диаграмм, то вы увидите окно сообщения, показанное на рис. 17.28. Щелкните на кнопке **ОК** и диаграмма появится на листе поверх уже существующей на нем.

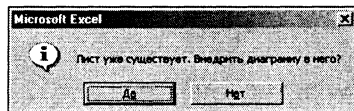


Рис. 17.28. Excel позволяет переместить внедренную диаграмму на существующий лист диаграмм

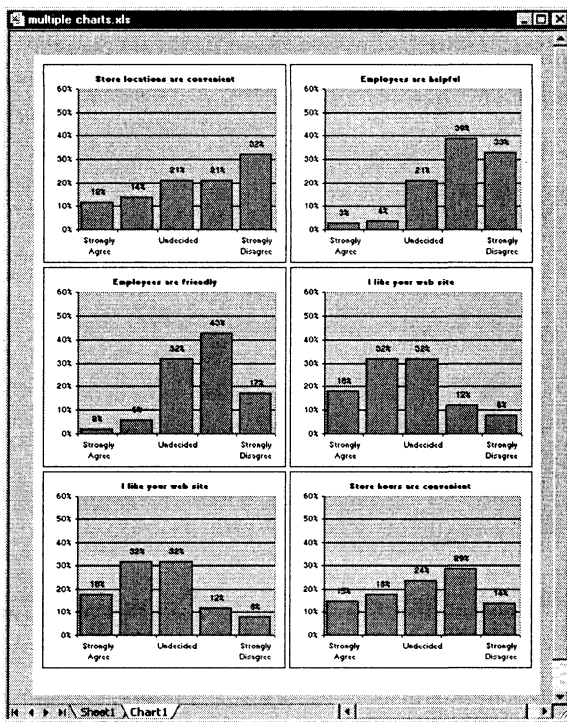


Рис. 17.29. Данный лист диаграмм содержит шесть вложенных диаграмм

Обычно встроенные диаграммы перемещают на *пустой* лист диаграмм. Для создания пустого листа диаграмм выберите пустую ячейку и нажмите клавишу <F11>. Или же можно выбрать на листе диаграмм область диаграммы и нажать клавишу .

При сохранении нескольких диаграмм на одном листе диаграмм у вас появляется возможность автоматического масштабирования диаграмм в соответствии с размером окна с помощью команды Вид⇒По размеру окна. На рис. 17.29 показан пример листа диаграмм, содержащий шесть вложенных диаграмм.

Просмотр вложенной диаграммы в окне

Когда вы активизируете вложенную диаграмму, она, на самом деле, находится в окне, которое в обычном состоянии *невидимо*. Для просмотра встроенной диаграммы в отдельном окне щелкните на ней правой кнопкой мыши и выберите из контекстного меню команду *Окно диаграммы*. Вложенная диаграмма остается на рабочем листе, но при этом она также отображается в плавающем окне. Данное окно можно перемещать, а также можно изменять его размер, однако его нельзя максимизировать. Если переместить окно, можно увидеть, что вложенная диаграмма осталась на своем месте. При активизировании любого другого окна окно диаграммы снова становится невидимым.

Изменение значений рабочего листа путем перетаскивания точек данных

В Excel имеется одна замечательная возможность, которая, правда, может таить в себе определенную опасность, заключающуюся в том, что данные рабочего листа можно изменить путем перетаскивания точек данных на двухмерной кривой, гистограмме, линейчатой, точечной или пузырьковой диаграмме.

Вот как это делается. Выберите в ряду диаграммы отдельную точку данных (не ряд в целом) и перетащите ее в направлении желаемого изменения значения. При перетаскивании выбранной точки относящееся к ней значение будет изменено на рабочем листе в соответствии с ее новым положением на диаграмме.

Если значение перетаскиваемой точки данных является результатом вычисления формулы, будет отображено диалоговое окно Подбор параметра. С помощью данного диалогового окна укажите ячейку, которую Excel должен изменить для того, чтобы формула возвращала тот результат, который был показан на диаграмме. Данная техника используется в случаях, когда вид диаграммы известен, а необходимо найти значения, для которых данная диаграмма может быть получена.



Очевидно, что использование данной методики может оказаться опасным, вследствие того, что можно непреднамеренно изменить значения, которые вы, на самом деле, изменять не собирались. Поэтому используйте данное средство исключительно на свой страх и риск.

Использование анимированных диаграмм

Большинство людей не подозревают о том, что Excel может создавать простую анимацию, используя формы и диаграммы (для анимации требуется макрос). Давайте рассмотрим двухмерный график, показанный на рис. 17.30.

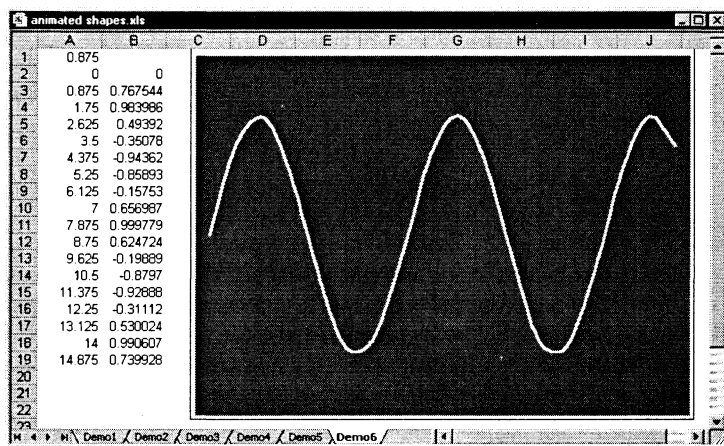


Рис. 17.30. Простая процедура VBA, превращающая диаграмму в интерактивный анимированный ролик

Значения x (столбец А) зависят от значения в ячейке А1. Значение в каждой строке определяется как сумма предыдущей строки и значения в ячейке А1. В столбце В содержится

формула, вычисляющая синус соответствующего значения в столбце А. Следующая простая процедура создает интересную анимацию. Она просто изменяет значение в ячейке А1, что приводит к изменению значений в диапазонах х и у.

```
Sub AnimatedChart()  
    Range("A1") = 0  
    For i = 1 To 150  
        Range("A1") = Range("A1") + 0,035  
    Next i  
    Range("A1") = 0  
End Sub
```

Создание “измерительной” диаграммы

На рис. 17.31 показано что-то, что можно назвать новым типом диаграмм, напоминающим измерительный прибор. На самом деле, это обычная круговая диаграмма, в которой один сегмент скрыт. Скрытый сегмент занимает 50 процентов диаграммы, а скрыта она за счет выбора прозрачной заливки и использования невидимой границы.

Круговая диаграмма использует значения в диапазоне А1:А3. В ячейке А1 содержится значение 1, и она представляет скрытый сегмент. В ячейке А2 находится значение, которое будет отображено на приборе. В ячейке А3 содержится простая формула:

=1-A2

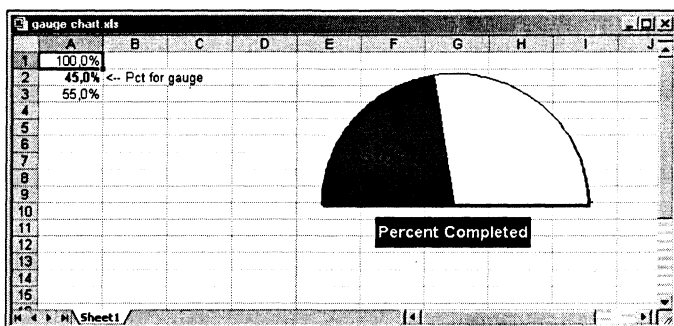


Рис. 17.31. Скрыв один сегмент, можно создать измерительную диаграмму

Создание диаграммы часов

На рис. 17.32 показан график, отформатированный так, что имеет вид часов. Он не только *выглядит* как часы, он и функционирует как часы. На самом деле, тяжело придумать причину, по которой кому-то захотелось бы создать на рабочем листе часы, подобные этим. Но создание такой книги можно воспринимать как вызов для настоящего изыскателя.

Диаграмма использует четыре ряда данных: один для часовой стрелки, один для минутной, один для секундной и последний для отображения чисел. По последнему ряду данных строится окружность с 12 точками. Числа созданы из введенных вручную подписей (см. врезку “Построение окружности”).

Формулы, перечисленные в табл. 17.2, вычисляют ряды данных для стрелок часов (в диапазоне G4:L4 содержатся не формулы, а нулевые значения).

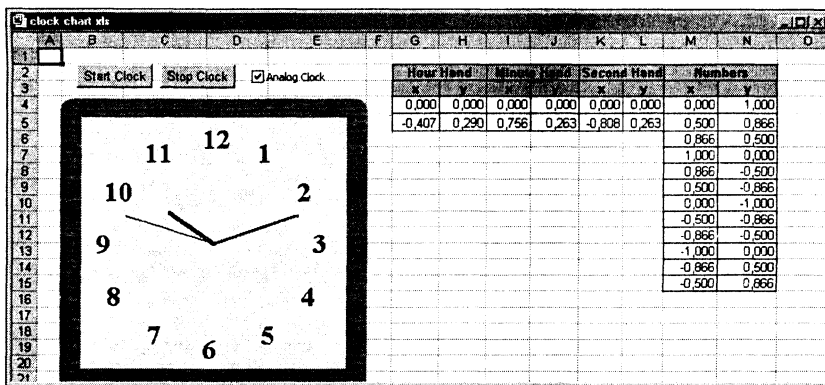


Рис. 17.32. Полнофункциональные часы, являющиеся, на самом деле, видоизмененным графиком

В данной рабочей книге используется простая VBA-процедура, пересчитывающая рабочий лист каждую секунду.

Таблица 17.1. Формулы, используемые для создания

Ячейка	Описание	Формула
G5	Начало часовой стрелки	=0,5*SIN((ЧАС(ТДАТА())+(МИНУТЫ(ТДАТА())/60))*(2*ПИ()/12))
H5	Конец часовой стрелки	=0,5*COS((ЧАС(ТДАТА())+(МИНУТЫ(ТДАТА())/60))*(2*ПИ()/12))
I5	Начало минутной стрелки	=0,8*SIN((МИНУТЫ(ТДАТА())+(СЕКУНДЫ(ТДАТА())/60))*(2*ПИ()/60))
J5	Конец минутной стрелки	=0,8*COS((МИНУТЫ(ТДАТА())+(СЕКУНДЫ(ТДАТА())/60))*(2*ПИ()/60))
K5	Начало секундной стрелки	=0,85*SIN(СЕКУНДЫ(ТДАТА())/60)*(2*ПИ()/60))
L5	Конец секундной стрелки	=0,85*COS(СЕКУНДЫ(ТДАТА())/60)*(2*ПИ()/60))

В дополнение к диаграмме часов рабочая книга содержит текстовое поле, в котором с помощью функции ТДАТА() отображается время, как показано на рис. 17.33. Данное поле, которое обычно скрыто, можно отобразить, сбросив флажок Аналоговые часы. В зависимости от состояния флажка связанная с ним простая VBA-процедура скрывает или отображает диаграмму.

При изучении данной рабочей книги примите во внимание следующие факторы:

- ♦ Объект диаграммы, названный *ClockChart*, скрывает диапазон *DigitalClock*, используемый для отображения времени в цифровом виде.

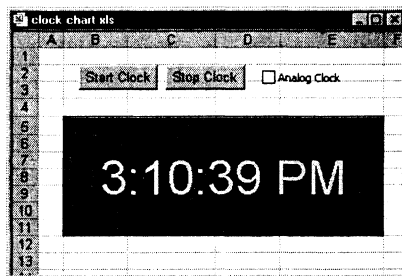
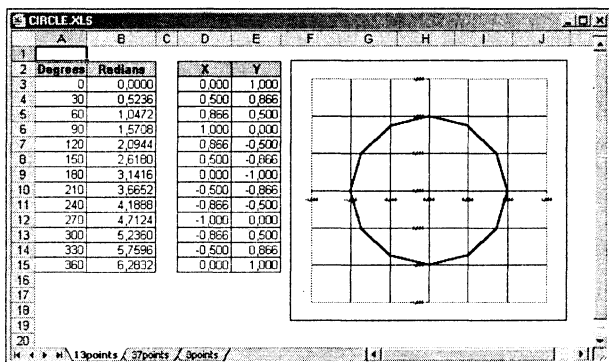


Рис. 17.33. Отобразить цифровые часы на рабочем листе проще, но не так интересно

- ◆ Две имеющиеся на рабочем листе кнопки взяты из панели инструментов **Формы**, и каждой из них назначена VBA-процедура (StartClock и StopClock).
- ◆ Находящийся на рабочем листе флажок (cbClockType) взят из панели инструментов **Формы**, а не **Элементы управления**. При щелчке на данном объекте выполняется процедура cbClockType_Click, которая переключает свойство Visible объекта ChartObject на противоположное значение. Когда данный объект скрыт, отображаются цифровые часы.
- ◆ Диаграмма является графиком с четырьмя рядами данных. Данные ряды представляют часовую, минутную и секундные стрелки, а также 12 чисел.

Построение окружности

Можно построить диаграмму (график), изображающую идеальную окружность. Для этого необходимо иметь два диапазона, один для значений x , другой для значений y . Число точек данных в наборе определяет *гладкость* окружности. Для получения гладкой линии можно также воспользоваться функцией *Сглаженная линия* диалогового окна *Формат ряда данных* (вкладка *Вид*).



В показанном примере (содержащемся на Web-узле) для создания окружности используется 13 точек данных. Если вы желаете работать с градусами, сгенерируйте ряд данных, подобный тому, что показан в столбце A. Ряд начинается с 0, а последовательные значения увеличиваются на 30 градусов. Если используются радианы (столбец B), ряд начинается с 0, а приращение составляет $\pi/6$.

Диапазоны, используемые в диаграмме, находятся в столбцах D и E. Если вы работаете с градусами, формула в ячейке D3 имеет следующий вид:

=SIN(РАДИАНЫ(A3))

В ячейке E3 находится формула:

=COS(РАДИАНЫ(A3))

Если вы работаете с радианами, в ячейке D3 используйте следующую формулу:

=SIN(A3)

А в ячейке E3:

=COS(A3)

Формулы в ячейках D3 и E3 просто копируются в находящиеся ниже строки.

Для того чтобы построить окружность на основе большего числа данных, необходимо уменьшить величину приращения значений в столбце A и B (или C и D, если вы работаете с радианами). Последнее значение должно быть тем же, что находится в строке 15. В градусах, приращение будет равно 360, деленное на число точек данных минус 1. В радианах, приращение будет равно 2π , деленное на число точек данных минус 1.

- ◆ Процедура UpdateClock выполняется при щелчке на кнопке Start Clock. Данная процедура определяет, какие из часов отображаются и производит соответствующее обновление.
- ◆ Процедура UpdateClock использует метод OnTime объекта Application. Данный метод позволяет выполнять процедуру в определенное время. Прежде чем процедура UpdateClock завершит свою работу, она вызывает новое событие OnTime, которое происходит через одну секунду. Другими словами, процедура UpdateClock вызывается каждую секунду.
- ◆ Процедура UpdateClock использует определенные тригонометрические вычисления для определения угла отображения стрелок на часах.

Рисование с помощью графиков

Последний пример не имеет абсолютно никакого практического применения, однако он может показаться вам довольно интересным или даже занимательным. Рабочий лист состоит из внедренной диаграммы (графика) и ряда элементов управления. (Данные элементы управления не являются компонентами ActiveX, они взяты из панели инструментов Формы).

При щелчке на одной из кнопок со стрелкой рисуется линия, длина которой определяется значением шага, задаваемом с помощью элемента прокрутки. После небольшой практики (и при определенной выдержке) вы сможете создавать простые чертежи. На рис. 17.34 представлен возможный рисунок.

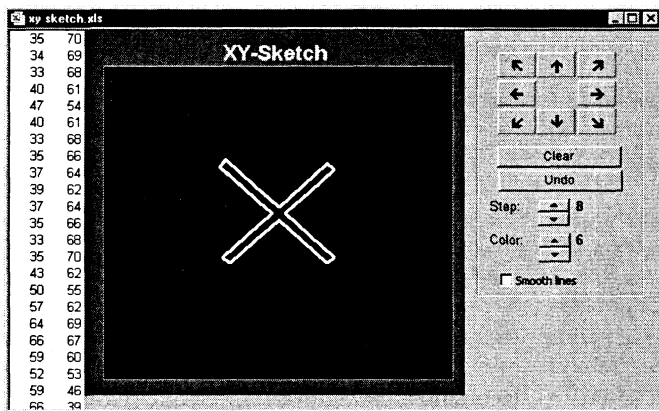


Рис. 17.34. Данный чертеж на самом деле является внедренной диаграммой (двухмерным графиком)

При щелчке на кнопке со стрелкой выполняется макрос, добавляющий в диапазон два значения: значение x и значение y . После этого он переопределяет два имени диапазонов (X range и Y Range), используемых в формуле РЯД диаграммы. Особенно полезна кнопка многоуровневой отмены действия. При щелчке на данной кнопке просто уничтожаются два последних значения из диапазонов и затем переопределяются их имена. В добавок ко всему имеется возможность изменения цвета линий, а также отображения сглаженной линии.

Резюме

В представленной главе была рассмотрена формула РЯД, используемая в диаграммах, а также приведен ряд примеров нестандартных диаграмм, которые могут быть созданы с помощью Excel. В главе также рассмотрены различные тенденции и описаны методы построения графиков функций. Было представлено ряд полезных способов построения диаграмм и описаны методики, которые несомненно будут использованы при построении собственных диаграмм.

В следующей главе приведена методика применения формул при управлении сводными таблицами.

Глава 18

Сводные таблицы

В этой главе...

- ◆ О сводных таблицах
- ◆ Создание сводной таблицы
- ◆ Группирование элементов сводной таблицы
- ◆ Создание вычисляемых полей или вычисляемых элементов
- ◆ Резюме

Сводные таблицы Excel, можно сказать, являются наиболее технологически изощренным компонентом программы. Кому-то может показаться, что данная глава лишняя в этой книге. В конце концов, сводные таблицы делают свою работу и без использования всяких формул. В этом-то и вся соль. Если вы еще не открыли для себя мощь сводных таблиц, то данная глава покажет, как сводные таблицы могут стать прекрасной альтернативой создания многих сложных формул.

О сводных таблицах

Сводная таблица является, по существу, динамическим отчетом, генерируемым на основе базы данных. База данных может находиться как на рабочем листе, так и во внешнем файле. Сводные таблицы могут оказать помощь в создании “удобоваримого” представления бесконечного числа строк и столбцов.

С помощью данного средства можно, например, создавать частотные распределения и таблицы массивов данных разных размерностей. В дополнение вы можете отображать промежуточные суммы с любой желаемой степенью детализации. Пожалуй самым прогрессивным аспектом сводных таблиц является их интерактивность. После того как сводная таблица создана, можно переупорядочить информацию практически любым мыслимым образом, можно даже вставить специальные формулы, которые будут производить новые вычисления. Можно также выполнять постформатирование (например, можно скомбинировать промежуточные итоги по северным и западным областям).

Сводные таблицы являются действительно уникальным инструментом. Заголовки строк и столбцов таблицы можно перемещать вокруг массива данных, создавая таким образом новые представления сводных данных.

Один небольшой недостаток использования сводных таблиц заключается в том, что они не обновляются автоматически при изменении исходных данных, как это происходит в случае создания отчетного доклада с использованием формул. Однако это не является серьезной проблемой, так как для того чтобы сводная таблица использовала последние измененные данные, необходимо всего лишь щелкнуть на единственной кнопке.

Пример сводной таблицы

Лучший способ понять идеологию сводных таблиц — увидеть хоть одну из них своими глазами. Начнем с рис. 18.1, на котором представлены данные, используемые для создания сводной таблицы.

	A	B	C	D	E	F	G
1	Date	Amount	AcctType	OpenedBy	Branch	Customer	
2	01.09.2000	340	Checking	New Accts	Central	Existing	
3	01.09.2000	15 769	CD	Teller	Westside	Existing	
4	01.09.2000	15 276	CD	New Accts	North County	Existing	
5	01.09.2000	12 000	CD	New Accts	Westside	Existing	
6	01.09.2000	5 000	CD	New Accts	North County	Existing	
7	01.09.2000	7 000	Savings	New Accts	North County	New	
8	01.09.2000	5 000	Savings	New Accts	Westside	Existing	
9	01.09.2000	4 623	Savings	New Accts	North County	Existing	
10	01.09.2000	5 879	Checking	New Accts	Central	Existing	
11	01.09.2000	3 171	Checking	New Accts	Westside	Existing	
12	01.09.2000	4 000	Savings	New Accts	Central	Existing	
13	01.09.2000	5 000	Checking	New Accts	Central	Existing	
14	01.09.2000	16 000	CD	New Accts	Central	New	
15	01.09.2000	50 000	Savings	New Accts	Central	Existing	
16	01.09.2000	13 636	CD	New Accts	North County	Existing	
17	04.09.2000	50 000	CD	New Accts	North County	New	
18	04.09.2000	15 000	CD	New Accts	Westside	New	
19	04.09.2000	13 000	CD	New Accts	North County	New	
20	04.09.2000	13 000	CD	New Accts	Central	Existing	
21	04.09.2000	3 000	Checking	New Accts	Central	Existing	
22	04.09.2000	2 878	Savings	New Accts	North County	Existing	
23	04.09.2000	13 519	CD	New Accts	Central	New	
24	04.09.2000	4 000	Checking	New Accts	Central	Existing	
25	04.09.2000	3 075	Checking	New Accts	Westside	Existing	
26	04.09.2000	4 000	Checking	New Accts	North County	Existing	
27	04.09.2000	6 000	Savings	New Accts	Central	Existing	
28	04.09.2000	65 000	Savings	New Accts	Westside	Existing	

Рис. 18.1. Данная база данных используется для создания сводной таблицы

База данных состоит из ежедневно обновляемой информации о счетах по трем филиалам банка. Она содержит 350 записей, отслеживающих следующую информацию:

- ♦ дата открытия каждого из счетов;
- ♦ размер открытого счета;
- ♦ тип счета;
- ♦ владелец счета;
- ♦ филиал, в котором открыт счет;
- ♦ открытие счета новым или старым клиентом;

База данных банковских счетов содержит массу информации. Однако из данных, представленных в данной форме, понять можно не много. Для того чтобы из данных можно было извлечь больше полезной информации, их нужно определенным образом организовать. Такое обобщение данных является по существу процессом выбора ответов на ряд вопросов. Вот некоторые из вопросов, которые интересны банковскому руководству:

- ♦ Какова общая сумма депозита в каждом из филиалов, и какова доля каждого из типов счетов?
- ♦ Сколько счетов было открыто в каждом из филиалов, и какова доля каждого из типов счетов?
- ♦ Каково распределение различных типов счетов?

- ◆ Какие из типов счетов открываются чаще всего?
- ◆ Как соотносятся результаты всех филиалов банка?
- ◆ Какой из филиалов открыл большее число счетов для новых клиентов?

Безусловно, для получения ответов на все эти вопросы можно написать соответствующие формулы. Однако во многих случаях использование сводных таблиц является более удачным решением. Создать сводную таблицу можно за считанные секунды, не написав при этом ни единой формулы.

На рис. 18.2 показана сводная таблица, созданная из базы данных, представленных на рис. 18.1. В данном примере показана величина вложений, разбитых по филиалам и типам счетов. Это одно из десятков возможных обобщений имеющихся данных.

	Central	North County	Westside	Общий итог
CD	859438	830139	344962	2034539
Checking	208208	92225	90597	391030
IRA	63380	134374	10000	207754
Savings	332349	152607	154000	638956
Общий итог	1463375	1209345	599559	3272279

Рис. 18.2 Простая сводная таблица

На рис. 18.3 показана другая сводная таблица, созданная из той же базы данных. В ней используется поле страницы для объекта Customer. В данном случае в таблице отображены данные лишь для старых клиентов (пользователь также может выбрать из списка поля варианты New или (Все)). Обратите внимание на изменение ориентации таблицы; филиалы находятся в строках, а типы счетов в столбцах. Это еще одна демонстрация гибкости сводных таблиц.

Терминология, используемая при работе со сводными таблицами

Прежде чем приступить к урокам мастерства обращения со сводными таблицами необходимо ознакомиться с используемой при работе с ними терминологией. Приведенная ниже иллюстрация может упростить понимание того, о чем идет речь.

- ◆ Поле столбца — это поле, имеющее в сводной таблице ориентацию столбца. Каждый элемент в сводной таблице занимает столбец. На приведенном рисунке, Customer является полем столбца с двумя элементами (Existing и New). В сводной таблице можно использовать вложенные поля.
- ◆ Область данных — это ячейки сводной таблицы, содержащие итоговые данные. Excel предлагает возможность обобщения данных несколькими способами (суммирование, среднее значение, счет и так далее). На представленном рисунке поле данных включает ячейки C5 : E20.
- ◆ Общая сумма — это строка или столбец, отображающий итоги для всех ячеек в строке или столбце. То, как будет вычисляться общая сумма (для столбцов, строки, или как для строк, так и для столбцов, или ни для строк, ни для столбцов), задается пользователем. В сводной таблице, приведенной на рисунке, общая сумма найдена и для строк, и для столбцов.
- ◆ Группа — это набор элементов, рассматриваемых как один элемент. Элементы можно группировать и вручную, и автоматически (например, можно сгруппировать даты в месяцы). Приведенная сводная таблица не содержит определенных групп.
- ◆ Элемент — это элемент поля, выступающий в сводной таблице в качестве заголовка строки или столбца. На приведенном рисунке Existing и New являются элементами поля

Customer. Поле Branch имеет 3 элемента: Central, North Country и Westside. Поле AcctType имеет четыре элемента: CD, Checking, IRA и Savings.

OpenedBy	Branch	AcctType	Existing	New	Общий итог
Central		CD	736269	123149	859438
		Checking	158980	49228	208208
		IRA	63360		63360
		Savings	261749	70600	332349
Central Итого			1220398	242977	1463375
North County		CD	677639	152500	830139
		Checking	72155	20070	92225
		IRA	125374	9000	134374
		Savings	113000	39607	152607
North County Итого			988168	221177	1209345
Westside		CD	273525	71437	344962
		Checking	83178	7419	90597
		IRA	10000		10000
		Savings	153500	500	154000
Westside Итого			520203	79356	599559
Общий итог			2728769	543510	3272279

- ◆ Поле страницы — это поле, имеющее в сводной таблице строчное представление, подобно грани трехмерного куба. На поле страницы может отображаться только один элемент или все элементы сразу. На приведенном рисунке OpenedBy является полем страницы, отображающим элемент New Accts.
- ◆ Обновление — это пересчет сводной таблицы после внесения изменений в исходные данные.
- ◆ Поле строки — это поле, имеющее в сводной таблице ориентацию строки. Каждый элемент в поле занимает строку. Поля строки можно объединять. На рисунке есть два поля строки: Branch и AcctType.
- ◆ Исходные данные — это данные, используемые для создания сводной таблицы. Могут находиться как на рабочем листе, так и во внешней базе данных.
- ◆ Промежуточные суммы — это строка или столбец, отображающий промежуточные суммы для части ячеек в строке или столбце сводной таблицы. В представленной таблице отображены промежуточные суммы для каждого филиала.

Customer	Branch	CD	Checking	IRA	Savings	Общий итог
Central		859438	208208	63360	332349	1463375
		830139	92225	134374	152607	1209345
		344962	90597	10000	154000	599559
		2034539	391030	207754	638956	3272279
North County Итого						
Westside Итого						
Общий итог						

Рис. 18.3. Сводная таблица, использующая поле страницы

Данные, используемые в сводной таблице

Для создания сводной таблицы могут использоваться не все данные. Обобщаемые данные упорядочены в виде базы данных. База данных может храниться как на рабочем листе (что называется списком), так и во внешнем файле базы данных. Хотя Excel способен генерировать сводные таблицы на основе любой базы данных, извлечь из нее пользу можно далеко не всегда.

Если говорить кратко, поля в таблице базы данных могут быть двух типов:

- ◆ Данные. Содержат исключения данных, для которых подводятся итоги. На рис. 18.1 поле Amount является полем данных.
- ◆ Категории. Описывают данные. На рис. 18.1 поля Date, AcctType, OpenedBy, Branch и Customer являются полями категорий, так как они описывают данные, содержащиеся в поле Amount.

Таблица базы данных может иметь любое количество полей данных и категорий. При создании сводной таблицы обычно ставится задача подведения итогов для одного или нескольких полей данных. Значения в полях категорий в сводной таблице представляются в виде строк, столбцов или страниц.

Однако существуют и исключения. Средство создания сводных таблиц Excel может оказаться полезным в случае баз данных, не содержащих настоящих числовых полей данных. Например, база данных, показанная на рис. 18.4 вообще не содержит числовых данных. Однако вы можете создать удобную сводную таблицу, которая вместо суммирования элементов выполняет их пересчет.

Обобщение информации в сводной таблице может выполняться методом, отличным от суммирования. Например, сводная таблица, показанная на рис. 18.5, построена в результате пересечения Month Born и Sex; в ячейках пересечения отображается количество соответствующих комбинаций для каждого месяца и пола.

Employee	Month Born	Sex	
1			
2	Anthony Taylor	July	Male
3	Charles S. Billings	February	Male
4	Christine Poundsworth	January	Female
5	Clark Bickerson	February	Male
6	Douglas Williams	March	Male
7	Janet Silberstein	April	Female
8	James Millen	May	Male
9	Jeffrey P. Jones	June	Male
10	Joan Morrison	July	Female
11	John T. Foster	August	Male
12	Kurt Kamichoff	January	Male
13	Michael Hayden	February	Male
14	Phyllis Todd	March	Female
15	Richard E. Card	April	Male
16	Rick Fogerty	May	Male
17	Robert H. Miller	June	Male
18	Stephen C. Carter	July	Male
19	Steven H. Katz	August	Male
20	Thomas F. Ahnett	February	Male

Рис. 18.4. Эта база данных не содержит числовых полей, однако она также может использоваться для создания сводной таблицы

Count of Employee	Sex	Grand Total		
Month Born	Female	Male	Grand Total	
5	January	1	3	4
6	February	0	6	6
7	March	3	1	4
8	April	1	3	4
9	May	1	4	5
10	June	1	4	5
11	July	1	5	6
12	August	1	4	5
13	September	0	5	5
14	October	1	4	5
15	November	1	4	5
16	December	1	5	6
17	Grand Total	12	48	60

Рис. 18.5. Данная сводная таблица подводит итоги по нечисловым полям, отображая количество вместо суммы

Создание сводной таблицы

Создание сводной таблицы производится с помощью ряда шагов, которые представляет мастер сводных таблиц и диаграмм. Вызов данного мастера осуществляется командой Данные⇒Сводная таблица. После этого вам предстоит пройти несколько описанных ниже этапов.



Данное обсуждение предполагает, что вы используете Excel 2000, или более новую версию программы. В предыдущих версиях процедура немного отличается.

Шаг 1. Указание расположения данных

При выборе команды Данные⇒Сводная таблица отображается диалоговое окно, показанное на рис. 18.6.

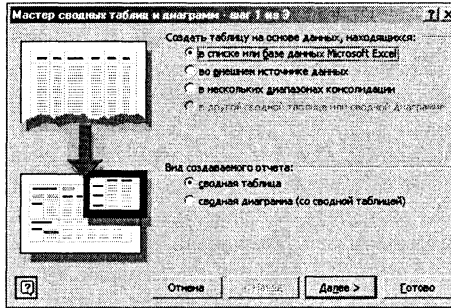


Рис. 18.6. Первое из трех диалоговых окон мастера сводных таблиц и диаграмм

На данном шаге определяется источник данных. Excel предоставляет довольно широкие возможности при выборе данных, которые могут использоваться для сводной таблицы (см. врезку “Источники данных для сводных таблиц”). В данном примере используется база данных в виде рабочего листа.



Отображаемые мастером диалоговые окна будут отличаться в зависимости от того, где находятся выбранные для анализа данные. В последующих подразделах приведены диалоговые окна, выводимые мастером для данных, находящихся в списке или базе данных Excel.

Источники данных для сводных таблиц

Данные, используемые в сводных таблицах, берутся из различных источников, включая базы данных или списки Excel, внешние источники данных, некоторые табличные диапазоны и другие сводные таблицы. Здесь приводится описание каждого из источников.

Базы данных или списки Excel

Обычно анализируемые данные хранятся в базе данных (или списке) рабочего листа. Для таких баз данных имеется ограничение на размер: 65535 записей и 256 полей. Работу с базой данных такого размера нельзя назвать эффективной (более того, для этого может оказаться недостаточно памяти). Первая строка в базе данных должна содержать имена полей. Других правил не существует. Данные могут быть значениями, текстом или формулами.

Внешние источники данных

При работе с данными из внешней базы данных для их извлечения необходимо использовать отдельное приложение Query. Вы можете использовать файлы dBASE, данные SQL Server и другие данные, поддерживаемые вашей системой. На втором этапе мастера сводных таблиц и диаграмм необходимо указать источник данных. Необходимо отметить, что начиная с версии Excel 2000 имеется возможность создания сводной таблицы из базы данных OLAP (On-Line Analytical Processing — Аналитическая обработка в реальном времени).

Несколько объединенных диапазонов

Сводная таблица также может быть создана из нескольких таблиц. Данная процедура эквивалентна объединению данных в таблицах. При создании сводной таблицы, для объединения информации в таблицах, вы получаете возможность использования всех доступных инструментов сводной таблицы.

Другие сводные таблицы

Excel позволяет создать сводную таблицу на основе уже существующей. Здесь имеет место некоторая неточность. Создаваемая сводная таблица основывается на **данных**, используемых первой сводной таблицей (а не на самой сводной таблице). Если в рабочей книге нет сводных таблиц, данная опция деактивизирована, то есть ее нельзя выбрать. Если требуется создание нескольких сводных таблиц на основе одних и тех же данных, с точки зрения использования памяти, вариант, когда создается одна таблица, которая затем используется в качестве источника для последующих сводных таблиц, более эффективен.

Шаг 2. Определение данных

Для перехода к следующему шагу мастера щелкните на кнопке **Далее**. На втором этапе мастера сводных таблиц и диаграмм необходимо указать данные. Помните, что вид данного диалогового окна зависит от выбора, сделанного в первом диалоговом окне; на рис. 18.7 показан вид диалогового окна для случая, когда на первом этапе выбран вариант в списке или базе данных Microsoft Excel.

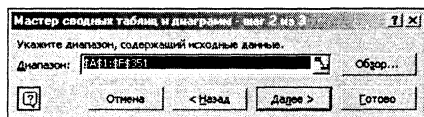


Рис. 18.7. На втором этапе необходимо указать диапазон данных

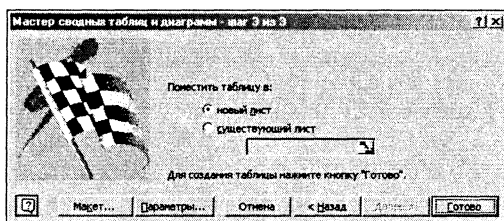


Рис. 18.8. На третьем этапе указывается расположение сводной таблицы

Если до выбора команды **Данные**⇨**Сводная таблица** маркер выделения ячейки находился в пределах базы данных рабочего листа, Excel на втором этапе мастера сводных таблиц и диаграмм определяет диапазон базы данных автоматически.

Для того чтобы открыть другую рабочую книгу и выбрать диапазон из нее, щелкните на кнопке **Обзор**. Для перехода к третьему шагу щелкните на кнопке **Далее**.



Если исходный диапазон для сводной таблицы расценивается как **база данных**, вы имеете возможность добавления новых данных в диапазон с помощью встроенной формы данных Excel. Именованный диапазон будет автоматически расширяться для включения новых записей.

Шаг 3. Завершение создания сводной таблицы

На рис. 18.8 показано диалоговое окно последнего шага мастера сводных таблиц и диаграмм. На данном шаге требуется указать расположение сводной таблицы.

При выборе переключателя **новый лист** Excel добавляет новый рабочий лист для помещения на него сводной таблицы. При выборе переключателя **существующий лист** сводная таблица размещается на текущем рабочем листе (вы можете указать расположение начальной ячейки).

На данном этапе можно выбрать некоторые параметры, определяющие вид сводной таблицы; для этого необходимо щелкнуть на кнопке **Параметры** (см. врезку “Параметры сводной таблицы”). Данные параметры можно перенастроить в любой момент после создания таблицы, так что большой необходимости в немедленной их установке нет.

Реальный макет сводной таблицы можно настроить с помощью одного из двух методов:

- ◆ Щелкнуть на кнопке **Макет** на третьем этапе мастера сводных таблиц и диаграмм. Открывшееся диалоговое окно затем можно использовать для макетирования сводной таблицы.
- ◆ Щелкнуть на кнопке **Готово** для создания пустой сводной таблицы. После этого макет сводной таблицы можно создавать с помощью панели инструментов **Список полей** сводной таблицы.

В последующих подразделах будут описаны оба метода.

Использование для макетирования сводной таблицы диалогового окна

При щелчке на кнопке **Макет** на последнем шаге мастера открывается диалоговое окно, показанное на рис. 18.9. Поля базы данных представляются кнопками, расположенными вдоль правого края диалогового окна. Просто перетащите кнопки на соответствующие области диаграммы сводной таблицы (находящейся в центре диалогового окна).

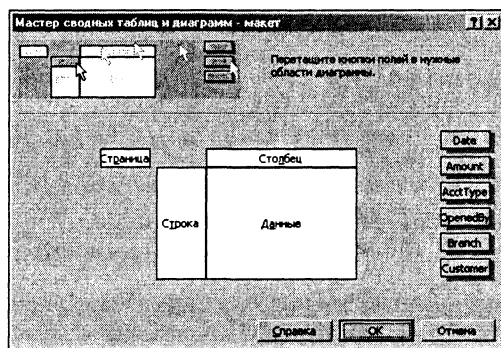


Рис. 18.9. Создание макета таблицы



Во всех версиях до Excel 2000 данное диалоговое окно появлялось на третьем этапе мастера. Это был единственный способ макетирования сводной таблицы.

Диаграмма сводной таблицы имеет четыре области:

- ◆ **Страница.** Значения в данной области выступают в качестве элементов страницы в сводной таблице.
- ◆ **Строка.** Значения в данной области выступают в качестве элементов строки в сводной таблице.
- ◆ **Данные.** В данной области подводятся итоговые для сводной таблицы поля.
- ◆ **Столбец.** Значения в данной области выступают в качестве элементов столбца в сводной таблице.

Можно перетащить в эти области любое желаемое количество кнопок полей, использовать все кнопки не обязательно. Все неиспользованные поля просто не будут отображаться в сводной таблице.

При перетаскивании кнопки поля в область данных мастер сводных таблиц и диаграмм применяет к данным функцию суммирования, если они числовые, и функцию счета, если данные нечисловые.

При макетировании сводной таблицы можно производить настройку полей, дважды щелкнув для этого на кнопке поля. Можно, например, указать способ (функцию) подведения итогов. Можно также определить, какие элементы поля следует скрыть или опустить. Если вы перетащили кнопку поля не туда, куда нужно, просто стащите ее за пределы диаграммы сводной таблицы. Следует отметить, что настройка полей возможна в любое время после создания сводной таблицы; это еще будет продемонстрировано ниже в данной главе.

На рис. 18.10 показано, какой вид принимает диалоговое окно после перетаскивания нескольких кнопок полей на диаграмму сводной таблицы. Данная сводная таблица показывает сумму поля Amount, разбитую по категориям AcctType (строки) и Customer (столбцы). В дополнение, поле Branch выступает и в качестве элемента страницы. Щелкните на кнопке ОК для возвращения к диалоговому окну Мастер сводных таблиц и диаграмм - Шаг 3 из 3.

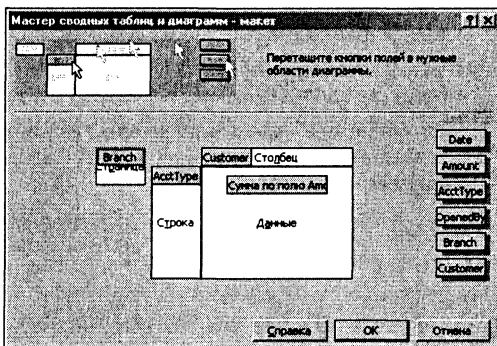


Рис. 18.10. Макет сводной таблицы после перетаскивания кнопок полей на диаграмму сводной таблицы

Использование для макетирования сводной таблицы панели инструментов Список полей сводной таблицы

Кому-то может больше нравиться создавать макет сводной таблиц прямо на рабочем листе с помощью панели инструментов Список полей сводной таблицы. Данная методика очень схожа с только что описанной, поскольку процесс макетирования также заключается в перетаскивании полей. Только в данном случае поля перетаскиваются с панели инструментов на рабочий лист.



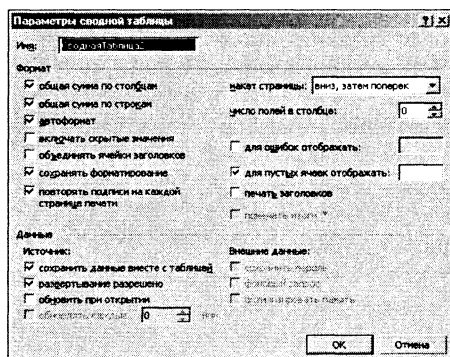
Данная методика не может применяться в версиях программы до Excel 2000. Примите также во внимание, что в Excel 2000 нет панели инструментов Список полей сводной таблицы; поля отображаются в виде кнопок на панели инструментов Сводные таблицы.

Параметры сводной таблицы

Excel предоставляет множество возможностей определения того, как сводная таблица будет выглядеть и работать. Для доступа к ее параметрам необходимо щелкнуть на кнопке Параметры на последнем шаге мастера сводных таблиц и диаграмм для отображения диалогового окна Параметры сводной таблицы. Работа с данным диалоговым окном возможна также и после создания сводной таблицы. Щелкните правой кнопкой мыши на любой ячейке сводной таблицы и затем выберите из контекстного меню команду Параметры таблицы. На приведенном рисунке показан вид диалогового окна Параметры сводной таблицы. Ниже приведен список изменяемых параметров:

- ♦ Имя. Сводной таблице присваивается имя. Excel сама предлагает имена в форме СводнаяТаблица 1, СводнаяТаблица 2 и так далее.

- ♦ Общая сумма по столбцам. Установите данный флажок, если хотите, чтобы Excel вычисляла общую сумму для элементов, отображаемых в столбцах.
- ♦ Общая сумма по строкам. Установите данный флажок, если хотите, чтобы Excel вычисляла общую сумму для элементов, отображаемых в строках.
- ♦ Автоформат. Установите данный флажок, если хотите, чтобы Excel применила к сводной таблице один из типов автоматического форматирования.
- ♦ Включать скрытые значения. Установите данный флажок, если хотите чтобы Excel включала скрытые элементы, находящиеся на странице, в промежуточные суммы.
- ♦ Объединять ячейки заголовков. Установите данный флажок, если хотите чтобы Excel объединяла ячейки внешней строки и столбца заголовков. Это позволяет сделать таблицу более читабельной.



- ♦ Сохранять форматирование. Установите данный флажок, если хотите чтобы Excel при обновлении сводной таблицы сохраняла все примененное вами форматирование.
- ♦ Повторять подписи на каждой странице печати. Установите данный флажок для того, чтобы расположить подписи строк на каждой странице распечатанного отчета сводной таблицы.
- ♦ Помечать итоги *. Доступна, лишь когда сводная таблица создается из источника данных OLAP. Если данный флажок установлен, после каждой промежуточной и общей суммы отображается звездочка для индикации того, что данные значения включают как отображаемые, так и все скрытые элементы.
- ♦ Макет страницы. Можно указать порядок, в котором будут отображаться поля страницы.
- ♦ Число полей в столбце. Можно указать число полей страницы, отображаемых в одной строке страницы.
- ♦ Для ошибок отображать. Можно указать значение, которое будет отображаться для ячеек сводной таблицы, отображающих ошибку.
- ♦ Для пустых ячеек отображать. Можно указать значение, которое будет показано для пустых ячеек сводной таблицы.
- ♦ Печать заголовков. Установите данный флажок для того, чтобы расположить подписи столбцов на каждой странице распечатанного отчета сводной таблицы.
- ♦ Сохранять данные вместе с таблицей. Если данный флажок установлен, Excel сохраняет дополнительную копию данных (называемую *кэшем сводной таблицы*), позволяющую более быстро пересчитывать таблицы при изменении макета. Если размер памяти велик, данный флажок необходимо сбросить (что немного замедлит обновление).
- ♦ Развертывание разрешено. Если данный флажок установлен, можно дважды щелкнуть на ячейке в сводной таблице для просмотра элементов, составляющих итоговое значение.
- ♦ Обновлять при открытии. Если флажок установлен, сводная таблица обновляется при каждом открытии рабочей книги.

- ◆ Обновлять каждые x минут. Если вы подключены к внешней базе данных, можно задать интервал времени, через который сводная таблица в открытой рабочей книге будет обновляться.
- ◆ Сохранить пароль. Если используется внешняя база данных, для которой требуется пароль, его можно сохранить как часть запроса, что избавит вас от необходимости его повторного ввода.
- ◆ Фоновый запрос. Если флажок установлен, Excel выполняет запрос базы данных в фоновом режиме, что позволяет вам не прерывать свою работу.
- ◆ Оптимизировать память. Данный параметр уменьшает объем памяти, используемой при обновлении запроса внешней базы данных.

Завершите работу с первыми двумя шагами мастера сводных таблиц и диаграмм. При желании настройте параметры сводной таблицы с помощью кнопки **Параметры**, находящейся в третьем диалоговом окне мастера. Однако не обращайтесь на кнопку **Макет**. Выберите место расположения сводной таблицы и щелкните на кнопке **Готово**. Excel отобразит шаблон сводной таблицы, подобный тому, что приведен на рис. 18.11. Шаблон содержит подсказки относительно того, куда следует помещать различные типы полей.

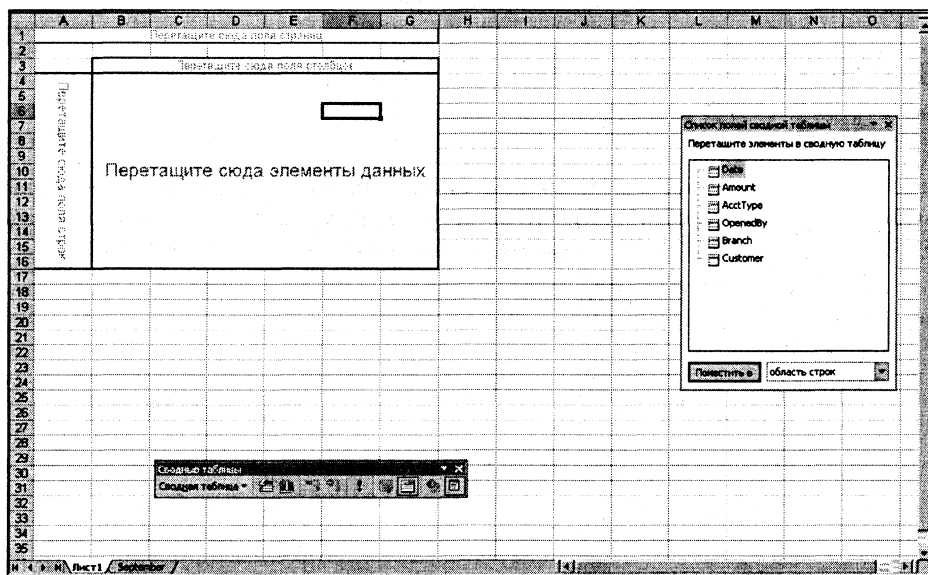


Рис. 18.11. Используйте панель инструментов **Список полей сводной таблицы** для перетаскивания полей на отображаемый Excel шаблон сводной таблицы

Перетащите и опустите поля с панели инструментов **Список полей сводной таблицы** на шаблон. Или же выберите имя поля, выберите для него место из раскрывающегося списка и щелкните на кнопке **Поместить в**. Excel продолжает обновление сводной таблицы при добавлении и удалении полей. По этой причине данный метод проще использовать, если элементы *данных* перетаскивать в конце. Другими словами, расположите элементы полей, а затем укажите данные для подведения итогов.

Если вы допустите ошибку, просто вытащите поле за пределы шаблона и опустите его на лист, — Excel удалит его из сводной таблицы. Все поля, даже после их использования, остаются на панели инструментов **Список полей сводной таблицы**.

Конечный продукт

На рис. 18.12 показан результат выполнения данного примера. Обратите внимание, что поля страницы отображаются в виде раскрывающегося списка. Можно также выбрать элемент с именем (Все); при этом будут отображены все данные.

Branch	Amount	Customer	
CD	1687453	347086	2034539
Checking	314313	76717	391030
IRA	196754	9000	207754
Savings	528249	110707	638956
Общий итог	2728769	543510	3272279

Рис. 18.12. Сводная таблица, созданная с помощью мастера сводных таблиц и диаграмм

Группирование элементов сводной таблицы

Одним из самых полезных свойств сводной таблицы является возможность комбинирования элементов в группы. Для группирования объектов выделите их, щелкните на них правой кнопкой мыши и выберите из контекстного меню команду Группа и структура⇒Группировать.

Date	Sales
01.06.2000	1344
02.06.2000	1189
05.06.2000	1023
06.06.2000	998
07.06.2000	1384
08.06.2000	1156
09.06.2000	1185
12.06.2000	1256
13.06.2000	1030
14.06.2000	641
15.06.2000	1475
16.06.2000	792
19.06.2000	1200
20.06.2000	1187
21.06.2000	532
22.06.2000	787
23.06.2000	1193
26.06.2000	1233
27.06.2000	675
28.06.2000	691
29.06.2000	1448
30.06.2000	959
03.07.2000	1301
04.07.2000	555
05.07.2000	1027
06.07.2000	812
07.07.2000	1160
10.07.2000	698

Рис. 18.13. Сводная таблица может использоваться для подведения итогов продаж по месяцам

Date	Total
6/1/2000	1344
6/2/2000	1189
6/5/2000	1023
6/6/2000	998
6/7/2000	1384
6/8/2000	1156
6/9/2000	1185
6/12/2000	1256
6/13/2000	1030
6/14/2000	641
6/15/2000	1475
6/16/2000	792
6/19/2000	1200
6/20/2000	1187
6/21/2000	532
6/22/2000	787
6/23/2000	1193
6/26/2000	1233

Рис. 18.14. Сводная таблица до группирования по месяцам

Если поле содержит даты, Excel способна создать группы автоматически. На рис. 18.13 показана простая таблица базы данных с двумя полями: Date и Sales. Данная таблица содержит 370 записей и покрывает диапазон времени с 1 июня 2000 года по 1 ноября 2001. Цель состоит в том, чтобы подытожить информацию о продажах по месяцам.

На рис. 18.14 показана сводная таблица, созданная для дат. Нет ничего неожиданного в том, что она имеет тот же вид, что входные данные, так как даты не сгруппированы. Для группирования элементов по месяцам щелкните правой кнопкой мыши на заголовке Date и выберите команду Группа и структура → Группировать. Будет открыто диалоговое окно Группирование, показанное на рис. 18.15.

Копирование сводной таблицы

Сводная таблица является специальным типом объектов и управлять ею с помощью обычных средств нельзя. Например, вы не можете вставить в сводную таблицу новую строку или формулу. Если вы хотите работать со сводной таблицей как с обычной, сделайте ее копию.

Для того чтобы скопировать сводную таблицу, выделите ее и выберите команду Правка → Копировать. Затем активизируйте новый рабочий лист и выберите команду Правка → Специальная вставка. Выберите переключатель значения и щелкните на кнопке ОК. Содержимое сводной таблицы будет скопировано в новое место, где с ним можно будет делать все, что вам захочется. При необходимости (для копирования форматирования из сводной таблицы) можно также повторно воспользоваться Правка → Специальная вставка и выбрать переключатель форматы.

Данная техника может оказаться полезной в случае, когда вы хотите создать стандартную диаграмму. Если попытаться создать диаграмму из сводной таблицы, Excel создаст сводную диаграмму, содержащую кнопки полей. Иногда вам может потребоваться создать обычную диаграмму.

Примите во внимание, что скопированная информация больше не связана с исходными данными. Если последние будут изменены, в скопированной сводной таблице данные изменения не отобразятся.

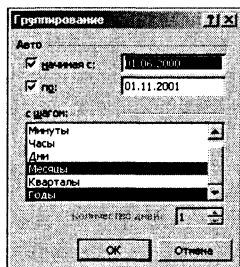


Рис. 18.15. Диалоговое окно Группирование используется для группирования элементов в сводной таблице

Sum of Sales		
Years	Date	Total
2000	Jun	23378
	Jul	21354
	Aug	21913
	Sep	22188
	Oct	22067
	Nov	21198
2001	Dec	21166
	Jan	22739
	Feb	21842
	Mar	24554
	Apr	20875
	May	23814
	Jun	19311
	Jul	20738
	Aug	23226
	Sep	20596
	Oct	21911
Grand Total		372890

Рис. 18.16. Сводная таблица после группирования по месяцам



В версиях до Excel 2002 рассматриваемая команда контекстного меню имеет другое название.

Из списка диалогового окна выберите элементы Месяцы и Годы, убедитесь также, что начальные и конечные даты указаны верно. Щелкните на кнопке ОК. Элементы поля Date сводной таблицы будут сгруппированы по годам и месяцам (см. рис. 18.16).



Если в диалоговом окне Группирование выбрать лишь элемент Месяцы, месяцы, относящиеся к разным годам, будут скомбинированы вместе. Например, элемент Июнь будет отображать продажи за данный месяц для 2000 и 2001 года.

Создание вычисляемых полей или вычисляемых элементов

После создания сводной таблицы можно создать два типа *формул* для дальнейшего анализа.

- ◆ Вычисляемое поле. Новое поле, создаваемое из других полей в сводной таблице. Вычисляемое поле должно находиться в области данных сводной таблицы (оно не может использоваться в области страницы, строки или столбца).
- ◆ Вычисляемый элемент. Такой элемент использует содержимое других элементов в пределах поля сводной таблицы. Вычисляемый элемент должен находиться в области страницы, строки или столбца (он не может использоваться в области данных).

Формулы, используемые для создания вычисляемых полей и элементов, не являются стандартными формулами Excel. Иначе говоря, вы не вводите формулы в ячейки. Вместо этого формулы вводятся в диалоговом окне и хранятся вместе с данными сводной таблицы.



Начиная с Excel 2000 имеется возможность использования в качестве источника для сводной таблицы базы данных OLAP. Однако в этом случае вычисляемые поля или элементы не могут быть созданы.

В примерах, приведенных в данном разделе, используется таблица базы данных рабочего листа, представленная на рис. 18.17. Таблица имеет пять полей и 48 записей. Каждая запись представляет информацию о месячных продажах каждого из торговых представителей. Например, Аму, являющаяся торговым представителем в северных областях, продала за январь 239 единиц продукции на общую сумму \$23 049.

На рис. 18.18 показана простая сводная таблица, созданная на основе этих данных. В ней отображены продажи, разбитые по месяцам и торговым представителям.

В представленных примерах продемонстрировано следующее:

- ◆ Создание вычисляемого поля для определения средней стоимости единицы продукции.
- ◆ Создание вычисляемого элемента для нахождения общей суммы по кварталам.

Создание в сводной таблице вычисляемого поля

Так как сводная таблица является особым типом диапазона данных, внутри такой таблицы нельзя вставлять строки или столбцы. Это означает, что вы не можете вставить формулы, ко-

торы бы производили бы расчеты на основе данных, содержащихся в сводной таблице. Для этой цели вы все же можете создавать вычисляемые поля. Вычисляемое поле состоит из вычислений, в которых использованы другие поля.

	A	B	C	D	E
1	SalesRep	Region	Month	Sales	Units Sold
2	Amy	North	Jan	\$23 040	239
3	Amy	North	Feb	\$24 131	79
4	Amy	North	Mar	\$24 646	71
5	Amy	North	Apr	\$22 047	71
6	Amy	North	May	\$24 971	157
7	Amy	North	Jun	\$24 218	92
8	Amy	North	Jul	\$25 735	175
9	Amy	North	Aug	\$23 638	87
10	Amy	North	Sep	\$25 749	567
11	Amy	North	Oct	\$24 437	95
12	Amy	North	Nov	\$25 955	706
13	Amy	North	Dec	\$25 899	180
14	Bob	North	Jan	\$20 024	103
15	Bob	North	Feb	\$23 822	267
16	Bob	North	Mar	\$24 854	96
17	Bob	North	Apr	\$22 836	74
18	Bob	North	May	\$25 320	231
19	Bob	North	Jun	\$24 733	164
20	Bob	North	Jul	\$21 184	68
21	Bob	North	Aug	\$23 174	114
22	Bob	North	Sep	\$25 999	84

Рис. 18.17. На основе этих данных продемонстрированы возможности вычисляемых полей и элементов

	A	B	C	D	E	F
1	Region	(Etc)				
2						
3	Сумма по полю Sales	SalesRep				
4	Month	Amy	Bob	Chuck	Doug	Общий итог
5	Apr	22047	22836	22058	29338	96261
6	Aug	23636	23174	21273	29506	97591
7	Dec	25899	23179	20583	28670	98331
8	Feb	24131	23822	23494	29953	101400
9	Jan	23040	20024	19886	26254	89214
10	Jul	25735	21184	23032	25044	94995
11	Jun	24218	24733	23965	23371	100287
12	Mar	24646	24854	21824	25041	96365
13	May	24971	25320	20280	25150	95721
14	Nov	25355	23449	19832	25953	95069
15	Oct	24437	22639	19625	27113	93814
16	Sep	25749	25999	21584	29061	102393
17	Общий итог	293866	281715	257436	328454	1161481

Рис. 18.18. Данная сводная таблица создана на основе данных, представленных на рис. 18.17

Вычисляемое поле обычно используется как средство отображения в сводной таблице новой информации. По существу оно выступает альтернативой создания нового поля данных в исходной базе данных. Вычисляемое поле не может использоваться в поле страницы, строки или столбца.

Для примера давайте предположим, что требуется вычислить среднюю стоимость единицы продукции. Данная величина может быть найдена путем деления значения поля Sales на значение поля Units Sold. Результат должен находиться в новом поле сводной таблицы (в вычисляемом поле).

Для создания вычисляемого поля, содержащего результат деления значений поля Sales на значения поля Units Sold, используйте следующую процедуру:

1. Поместите маркер выделения ячейки в любом месте сводной таблицы.
2. Из панели инструментов Сводные таблицы выберите команду Сводная таблица⇒Формулы⇒Вычисляемое поле. Будет отображено диалоговое окно Вставка вычисляемого поля.

- В поле **Имя** введите имя описательного характера, а в поле **Формула** укажите необходимую формулу (см. рис. 18.19). В формуле могут использоваться другие поля и функции рабочего листа. В данном примере имя вычисляемого поля — **Avg Unit Price**, а формула имеет вид:

=Sales/'Units Sold'

- Щелкните на кнопке **Добавить** для добавления этого нового поля.
- Щелкните на кнопке **ОК** для закрытия диалогового окна **Вставка вычисляемого поля**.

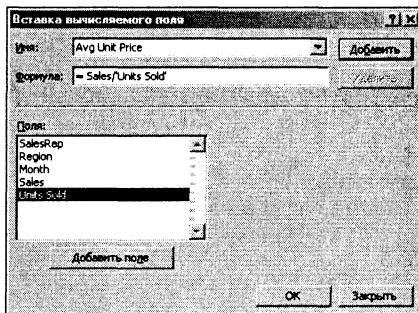


Рис. 18.19. Диалоговое окно **Вставка вычисляемого поля**

После того как вычисляемое поле создано, Excel добавляет его в область данных сводной таблицы. С ним можно обращаться так же, как и с любым другим полем, за одним исключением: оно не может быть перемещено в область страницы, строки или столбца (поле должно оставаться в области данных).

На рис. 18.20 показана сводная таблица после добавления в нее вычисляемого поля. В новом поле отображается средняя стоимость единицы продукции. Вычисляемое поле также становится доступным на панели инструментов **Список полей сводной таблицы** вместе с другими полями сводной таблицы.

1	Region	(Все)					
2							
3							
4	Month	Данные	SalesRep				Общий итог
5	Apr	Сумма по полю Sales	22047	22638	22058	29338	96261
6		Сумма по полю Avg Unit Price	\$311	\$309	\$230	\$132	\$208
7	Aug	Сумма по полю Sales	23638	23174	21273	29506	97591
8		Сумма по полю Avg Unit Price	\$272	\$203	\$28	\$286	\$91
9	Dec	Сумма по полю Sales	25899	23179	20583	26670	98331
10		Сумма по полю Avg Unit Price	\$144	\$50	\$116	\$145	\$96
11	Feb	Сумма по полю Sales	24131	23822	23494	29953	101400
12		Сумма по полю Avg Unit Price	\$305	\$89	\$159	\$35	\$75
13	Jan	Сумма по полю Sales	23040	20024	19886	26264	89214
14		Сумма по полю Avg Unit Price	\$96	\$194	\$209	\$295	\$169
15	Jul	Сумма по полю Sales	25735	21184	23032	25044	94995
16		Сумма по полю Avg Unit Price	\$147	\$312	\$149	\$305	\$198
17	Jun	Сумма по полю Sales	24218	24733	23965	27371	100287
18		Сумма по полю Avg Unit Price	\$263	\$151	\$32	\$288	\$90
19	Mar	Сумма по полю Sales	24646	24854	21824	25041	96365
20		Сумма по полю Avg Unit Price	\$347	\$259	\$263	\$291	\$287
21	May	Сумма по полю Sales	24971	25320	20280	25150	95721
22		Сумма по полю Avg Unit Price	\$159	\$110	\$45	\$104	\$88
23	Nov	Сумма по полю Sales	25355	23949	19832	25953	95089
24		Сумма по полю Avg Unit Price	\$36	\$220	\$283	\$320	\$98
25	Oct	Сумма по полю Sales	24437	22639	19625	27113	93814
26		Сумма по полю Avg Unit Price	\$257	\$87	\$236	\$226	\$168
27	Sep	Сумма по полю Sales	25749	25999	21584	29061	102393
28		Сумма по полю Avg Unit Price	\$46	\$310	\$189	\$199	\$114
29	Итого	Сумма по полю Sales	293866	281715	257436	328464	1161481
30		Сумма по полю Avg Unit Price	\$117	\$138	\$86	\$142	\$118
31							

Рис. 18.20. В данной сводной таблице используется вычисляемое поле



Формулу можно записать вручную с помощью клавиатуры или путем двойного щелчка на элементе списка **Поля**. При двойном щелчке элемент переносится в поле **Формула**. Так как в имени поля **Units Sold** есть пробел, Excel добавляет вокруг него одиночные кавычки.



Применяемые формулы могут использовать и функции рабочего листа, однако последние не могут ссылаться на ячейки или именованные диапазоны.

Вставка в сводную таблицу вычисляемого элемента

В предыдущем подразделе было описано, как создать вычисляемое поле. Excel также

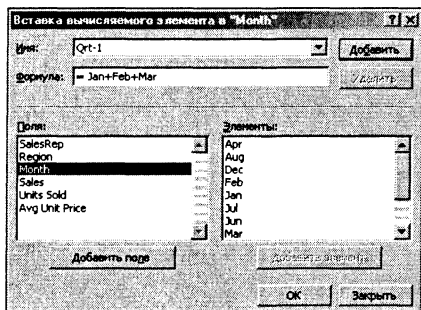


Рис. 18.21 Диалоговое окно Вставка вычисляемого элемента

позволяет создать для поля сводной таблицы *вычисляемый элемент*. В примере предыдущего подраздела используется поле с именем Month, состоящее из текстовых строк. Можно создать вычисляемый объект (например, с именем Qtr-1), отображающий сумму месяцев Jan, Feb и Mar.

Это же можно сделать с помощью группирования элементов, однако при этом отдельные месяцы оказываются скрытыми, и отображается лишь сумма для группы. Создание вычисляемого объекта для квартальных итогов более удобно, так как отображается и сумма, и данные по отдельным месяцам.

Чтобы создать вычисляемый элемент для суммирования данных за январь, февраль и март, выполните следующие действия:

1. Поместите маркер выделения ячейки в область строки, столбца или страницы сводной таблицы, содержащую элемент, который будет вычислен. В данном примере маркер должен находиться в области Month.
2. На панели инструментов Сводные таблицы выберите команду Сводная таблица⇒Формулы⇒Вычисляемый объект. Excel отобразит диалоговое окно Вставка вычисляемого элемента.
3. В поле *Имя* введите имя для нового элемента, а в поле *Формула* укажите необходимую формулу (см. рис. 18.21). В формуле могут использоваться элементы из других полей, но не могут применяться функции рабочего листа. В данном примере имя нового объекта — Qrt-1, а формула имеет вид:
 $=\text{Jan}+\text{Feb}+\text{Mar}$
4. Щелкните на кнопке *Добавить*.
5. Повторите пп. 3 и 4 для создания дополнительных вычисляемых объектов Qrt-2 ($=\text{Apr}+\text{May}+\text{Jun}$), Qrt-3 ($=\text{Jul}+\text{Aug}+\text{Sep}$) и Qrt-4 ($=\text{Oct}+\text{Nov}+\text{Dec}$).
6. Щелкните на кнопке *ОК* для закрытия диалогового окна.



Если в сводной таблице используется вычисляемый объект, во избежание двойного расчета может потребоваться отключить отображения общей суммы.

После завершения создания вычисляемых объектов они появляются в сводной таблице. На рис. 18.22 показана сводная таблица после того, как к ней добавлено четыре элемента кварталов. Вы можете изменить порядок объектов путем их перетаскивания.

1	Region	(Все)				
2						
3	Сумма по полю Sales	SalesRep				
4	Month	Amy	Bob	Chuck	Doug	Общий итог
5	Apr	22047	22638	22058	29336	96281
6	Aug	23638	23174	21273	29506	97591
7	Dec	25899	23179	20583	28670	98331
8	Feb	24131	23822	23494	29953	101400
9	Jan	23040	20024	19886	26264	89214
10	Jul	25735	21184	23032	25044	94995
11	Jun	24218	24733	23965	27371	100287
12	Mar	24646	24854	21824	25041	96365
13	May	24971	25320	20280	25150	95721
14	Nov	25355	23949	19832	25953	95089
15	Oct	24437	22639	19625	27113	93814
16	Sep	25749	25999	21584	29061	102393
17	Qrt-1	71817	68700	65204	81258	286979
18	Qrt-2	71236	72691	66303	81859	292289
19	Qrt-3	75122	70357	65889	83611	294979
20	Qrt-4	75691	69767	60040	81736	287234
21	Общий итог	587732	563430	514872	656928	2322962
22						
23						
24						
25						
26						

Рис. 18.22. В данной сводной таблице для отображения квартальных итогов использованы вычисляемые элементы



Вычисляемый объект отображается в сводной таблице лишь в том случае, когда отображается то поле, на котором он основан. Если вы удалите поле из категории строки или столбца в категорию данных, вычисляемый элемент отображаться не будет.

Резюме

Данная глава — это вступление в тему сводных таблиц. В ней показано, как создать сводную таблицу, сгруппировать элементы и как создать вычисляемые поля или объекты.

В следующей главе речь пойдет об условном форматировании и о проверке данных.

Условное форматирование и проверка вводимых значений

В этой главе...

- ◆ Условное форматирование
- ◆ Проверка вводимых значений
- ◆ Резюме

В данной главе рассматриваются два очень полезных средства Excel: условное форматирование и проверка вводимых значений. Может показаться, что данные вопросы имеют мало общего с формулами. Однако, как вы сможете увидеть из ниже изложенного материала, если взять формулы в рассмотрение, данные средства окажутся способными сотворить, действительно нечто потрясающее.



Средства условного форматирования и проверки достоверности данных были введены в Excel 97. Поэтому изложенный материал не принесет много пользы тем, кто использует более ранние версии программы.

Условное форматирование

Условное форматирование позволяет выполнять форматирование ячеек выборочно и автоматически, основываясь на их содержимом. Например, можно сделать так, чтобы все ячейки с отрицательными значениями в указанном диапазоне имели желтый фон. Если вы вводите в диапазоне новое значение или изменяете уже существующее, Excel проверяет его и применяет к нему правила условного форматирования. Если значение оказывается отрицательным, выполняется заливка фона ячейки. В противном случае форматирование не применяется.

Условное форматирование очень полезно при быстром поиске ячеек с ошибочными данными или ячеек особого типа. Вы можете применить специальное форматирование (например заливку красным цветом) для того, чтобы можно легко идентифицировать ячейки с определенным содержанием.

Полезно ли данное средство? Несомненно. Кроме того, стоит копнуть немного глубже, и вы узнаете о таких возможностях, о которых раньше не могли бы и мечтать. Ключом к ним, как вы увидите, является определение условий с помощью формул. В данном разделе приведено описание средства условного форматирования Excel, а также указаны некоторые ограничения и серьезные упущения.

Настройка условного форматирования

Чтобы применить условное форматирование к ячейке или диапазону, выполните следующие действия:

1. Выделите ячейку или диапазон.
2. Выберите команду **Формат**⇒**Условное форматирование**. Excel отобразит диалоговое окно **Условное форматирование**, показанное на рис. 19.1.

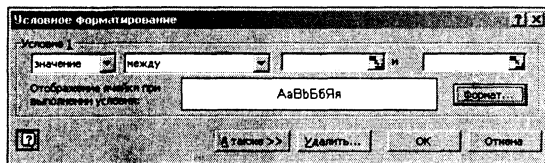


Рис. 19.1. Диалоговое окно **Условное форматирование**

3. Из раскрывающегося списка выберите либо значение **значение** (для простого условного форматирования), либо **формула** (для форматирования, базирующегося на использовании формул).
4. Определите условие (или введите формулу).
5. Щелкните на кнопке **Формат** и определите форматирование, которое необходимо применить, если заданное условие удовлетворяется.
6. Для того чтобы добавить дополнительные условия (еще одно или несколько), щелкните на кнопке **A** также и затем повторите пп. 3–5.
7. Щелкните на кнопке **ОК**.

После того как данные операции будут проделаны, ячейка или диапазон будут отформатированы в соответствии с установленными условием или условиями. Данное форматирование, несомненно, динамическое: если изменить содержимое ячейки, Excel снова оценит новое содержимое и в соответствии с результатом снимет или применит соответствующее форматирование.

Возможные типы форматирования

При щелчке на кнопке **Формат** диалогового окна **Условное форматирование** открывается диалоговое окно **Формат ячеек**, показанное на рис. 19.2. Это модифицированный вариант стандартного диалогового окна **Формат ячеек** — в нем нет вкладок **Число**, **Выравнивание** и **Защита**, зато есть кнопка **Очистить**. Вы можете определить любой из ниже перечисленных форматов:

- ◆ Начертание шрифта (обычное, полужирное или курсив)
- ◆ Подчеркивание шрифта
- ◆ Цвет шрифта
- ◆ Зачеркивание шрифта
- ◆ Отображение границы
- ◆ Стиль линии границы
- ◆ Цвет линии границы

- ◆ Цвет заливки ячейки
- ◆ Узор заливки ячейки

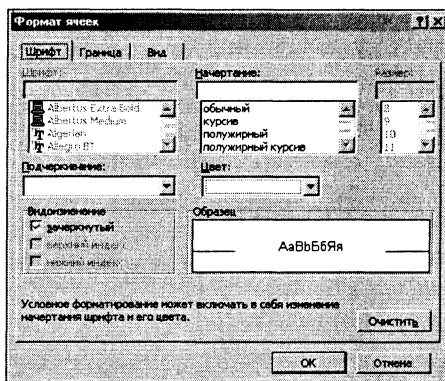


Рис. 19.2. Диалоговое окно *Формат ячеек*, используемое при условном форматировании

Обратите внимание, что нельзя задать шрифт или его размер; вероятно, причиной является то, что данные характеристики могут приводить к изменению высоты строки. Разработчики посчитали, что возникновение данного эффекта может не всегда однозначно восприниматься пользователями, и, кроме того, это может приводить к изменению областей разбивки рабочего листа на страницы при печати.



В диалоговом окне *Формат ячеек* доступны 56 цветов палитры рабочей книги. Если ни один из них вам не подходит, вы можете изменить цветовую палитру рабочей книги. Для этого выберите команду *Сервис*⇒*Параметры* и щелкните на вкладке *Цвет* диалогового окна *Параметры*. Выберите цвет; для того чтобы его изменить, щелкните на кнопке *Изменить*. Но будьте внимательны, так как изменение цвета может привести к изменению других цветовых параметров рабочей книги.



Диалоговое окно *Найти и заменить* в Excel 2002 позволяет найти в рабочем листе ячейки с определенным форматированием. Данное средство не обнаруживает ячеек, содержащих форматирование, установленное с помощью средства условного форматирования.

Определение условий

Список в левой части диалогового окна *Условное форматирование* позволяет выбрать одно из двух значений:

- ◆ значение — для простых условий;
- ◆ формула — для более сложных, основанных на формулах условий.

В приведенных ниже подразделах обсуждаются оба типа условий.

Простые условия

При выборе значения значение могут определяться условия следующих типов:

- ◆ между (указывается два значения);
- ◆ вне (указывается два значения);

- ◆ равно (указывается одно значение);
- ◆ не равно (указывается одно значение);
- ◆ больше (указывается одно значение);
- ◆ меньше (указывается одно значение);
- ◆ больше или равно (указывается одно значение);
- ◆ меньше или равно (указывается одно значение).

Можно либо напрямую ввести значение(я), либо указать ссылку на ячейку(и).

Условия, основанные на формулах

При выборе значения формула вы имеет возможность указания формулы. Это выполняется в результате определения содержащей ее ячейки или путем ввода формулы непосредственно в данном диалоговом окне (см. рис. 19.3).

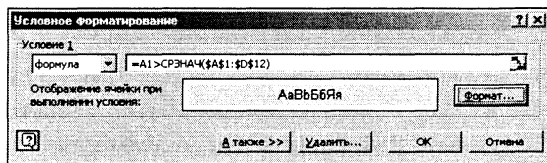


Рис. 19.3. Ввод формулы в диалоговом окне Условное форматирование



Формула должна быть логической, то есть возвращать значение ИСТИНА или ЛОЖЬ. Если формула возвращает ИСТИНА, условие удовлетворяется и условное форматирование применяется. Если возвращаемое значение ЛОЖЬ, условие не удовлетворяется и условное форматирование не применяется.

Как вы сможете увидеть на приводимых далее примерах, эффективность условного форматирования проявляется, когда формула вводится непосредственно в диалоговом окне Условное форматирование.

Если во введенной формуле имеется ссылка на ячейку, она считается относительной, с привязкой к левой верхней ячейке выбранного диапазона. Предположим, например, что вам нужно задать условное форматирование, применяющее заливку к пустым ячейкам в диапазоне В2 : В10. Выполните следующие действия:

1. Выделите диапазон В2 : В10.
2. Выберите команду **Формат**⇒**Условное форматирование**.
3. Из раскрывающегося списка выберите элемент **формула**.
4. В поле диалогового окна введите формулу
=В2=" "
5. Щелкните на кнопке **Формат** и выберите способ заливки.
6. Два раза щелкните на кнопке **ОК**.

Обратите внимание, что введенная формула содержит ссылку на левую верхнюю ячейку выбранного диапазона. Чтобы убедиться в том, что ссылка относительна, выберите ячейку В5 и посмотрите формулу условного форматирования для нее. Вы увидите, что для данной ячейки формула имеет вид:

=В5=" "

Изменение шрифта с помощью пользовательских числовых форматов

В некоторых случаях можно отказаться от использования условного форматирования в пользу пользовательского числового форматирования, позволяющего в зависимости от соответствия условиям изменять цвет шрифта. Например, приведенное ниже числовое форматирование отображает положительные значения черным цветом, отрицательные красным, а нулевые значения синим цветом:

[Черный] Основной; [Красный] - Основной; [Синий] Основной

Информацию о пользовательских числовых форматах можно почерпнуть из приложения В.

Обычно, при вводе формулы условного форматирования для диапазона ячеек, вы будете использовать ссылку на верхнюю левую ячейку выбранного диапазона. Исключением является случай, когда необходимо сослаться на определенную ячейку. Предположим, например, что вы выбрали диапазон A1:B20 и хотите применить форматирование ко всем ячейкам диапазона, значение которых превосходит значение в ячейке C1. Введите следующую формулу условного форматирования:

=A1>\$C\$1

В этом случае ссылка на ячейку C1 является абсолютной; она будет оставаться неизменной для всех остальных ячеек выбранного диапазона. Другими словами, формула условного форматирования для ячейки A2 будет иметь вид:

=A2>\$C\$1

Относительная ссылка изменилась, а абсолютная осталась такой же.

Работа с условными форматами

В данном подразделе приведена некоторая дополнительная, касающаяся условного форматирования, информация, которая может оказаться для вас полезной.

Множественные условия

Как отмечалось ранее, в диалоговом окне Условное форматирование, щелкая на кнопке A также, вы можете задать до трех условий. Можно, например, определить три следующих условия (и задать различное форматирование для каждого из них):

Значение меньше 0

Значение равно 0

Значение больше 0

В этом случае знак значения определяет применяемое форматирование.

Если ни одно из условий не удовлетворится, ячейка сохраняет свои существующие форматы. Если указаны множественные условия и удовлетворяется несколько из них, Excel применяет форматирование лишь для первого выполняющегося условия. Например, могут быть определены два условия следующего типа:

Значение больше 1 и меньше 12

Значение меньше 6

Ввод значения 4 удовлетворит оба условия. При этом к ячейке будет применен формат, определенный для первого условия.

Будьте осторожны при вставке

Очень важно помнить о том, что можно очень легко (слишком легко) уничтожить условное форматирование в ячейке или в диапазоне, вставив в ячейку скопированные данные.



Копирование ячейки и ее вставка в ячейку или диапазон, содержащий условное форматирование, приводит к уничтожению последнего в месте записи. При этом не выводится предупреждения. Безусловно, данная недоработка целиком на совести разработчиков Microsoft, однако при использовании условного форматирования вы должны не забывать о ней.

Копирование ячеек, содержащих условное форматирование

Информация об условном форматировании сохраняется вместе с ячейкой подобно информации об обычном форматировании. Это означает, что при копировании ячейки, содержащей условное форматирование, последнее тоже будет скопировано.



Для того чтобы скопировать только условные форматы, выделите ячейки, которые вы хотите отформатировать, и включите в их набор по крайней мере одну ячейку, имеющую условное форматирование, которое вы хотите скопировать. Выберите команду **Формат**⇒**Условное форматирование** и щелкните на кнопке **ОК**.

При вставке строк или столбцов внутри диапазона, содержащего условное форматирование, новые ячейки приобретают то же условное форматирование.

Удаление условного форматирования

Если удалить содержимое ячейки нажатием клавиши ****, условное форматирование (как и любое другое) не будет удалено. Для удаления всех условных форматов (а также любых других) выделите ячейки и выберите команду **Правка**⇒**Очистить**⇒**Форматы**.

Чтобы удалить только условное форматирование и оставить другие типы форматирования нетронутыми, необходимо использовать диалоговое окно **Условное форматирование**. Выделите ячейки, затем выберите команду **Формат**⇒**Условное форматирование**. Щелкните в диалоговом окне **Условное форматирование** на кнопке **Удалить**. Будет открыто диалоговое окно (рис. 19.4), позволяющее указать условия, которые вы хотите удалить. В данном диалоговом окне всегда содержатся флажки для трех условий, даже, если такое количество условий ранее не было определено.

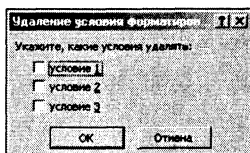


Рис. 19.4. Диалоговое окно **Удаление условия форматирования** используется для удаления одного или нескольких условий

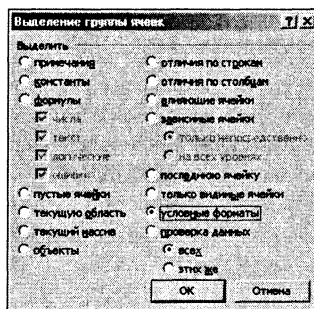


Рис. 19.5. С помощью диалогового окна **Переход** можно определить ячейки, содержащие условное форматирование



Удалить условное форматирование можно просто путем копирования ячейки, *не имеющей* условного форматирования, и ее вставки в нужную ячейку или диапазон. При этом, конечно же, также копируется значение (или формула) ячейки и все остальные форматы.

Обнаружение ячеек, содержащих условное форматирование

Посмотрев на ячейку, не всегда можно сказать, содержит она условное форматирование или нет. Для выделения таких ячеек вы можете воспользоваться диалоговым окном Excel Переход.

Для отображения диалогового окна Переход выберите команду Правка⇒Перейти (или нажмите клавишу <F5>). Щелкните на кнопке Выделить и затем выберите переключатель условные форматы (рис. 19.5). Для того чтобы выделить на рабочем листе все ячейки, содержащие условное форматирование, выберите переключатель всех. Если нужно выделить лишь ячейки с тем же условным форматированием, что и у активной ячейки, выберите переключатель этих же. Щелкните на кнопке ОК, и искомые ячейки будут выделены.

Использование ссылок на другие листы

Если введенная формула условного форматирования использует одну или несколько ссылок на другие листы, Excel выдаст сообщение об ошибке. Если необходимо сослаться на ячейку в другом листе, необходимо создать на нее ссылку в листе, содержащем условное форматирование. Например, если в формуле условного форматирования должна быть ссылка на ячейку A1 в рабочем листе Лист3, в ячейке активного листа может быть введена следующая формула:

=Лист3!A1

После этого в формуле условного форматирования используйте ссылку на эту ячейку.



Другой метод состоит в создании имени для ячейки (с помощью команды Вставка⇒Имя⇒Присвоить). После присвоения имени его можно использовать в диалоговом окне Условное форматирование в качестве ссылки на ячейку. При использовании данной методики именованная ячейка может находиться в любом рабочем листе рабочей книги.

Формулы условного форматирования

В данном подразделе приведен ряд примеров, демонстрирующих различные варианты использования условного форматирования. В каждом из данных примеров используется формула, введенная непосредственно в диалоговом окне Условное форматирование. Тип применяемого форматирования определяется удовлетворяемыми условиями.

Определение нечисловых данных

Следующая формула условного форматирования применяет форматирование к ячейке A1 лишь в том случае, если она содержит текст:

=ЕТЕКСТ (A1)

Чтобы данную формулу применить к диапазону, необходимо его предварительно выделить. В качестве аргумента функции ЕТЕКСТ должна использоваться верхняя левая ячейка диапазона.

Определение ячеек, содержащих значение выше среднего

Следующая формула условного форматирования применена к диапазону A1 : D12. Она задает форматирование всех ячеек данного диапазона, значение которых больше среднего значения диапазона (рис. 19.6):

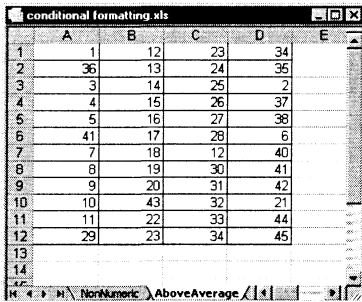
```
=A1>СРЗНАЧ ($A$1 : $D$12)
```

Обратите внимание, что ссылка на первую ячейку (A1) относительна, а в качестве аргументов формулы СРЗНАЧ используются абсолютные ссылки.

Нахождение дат, относящихся к определенному месяцу

Условно форматирование работает также и с датами. Следующая формула применяет форматирование лишь в том случае, когда дата относится к июню месяцу:

```
=МЕСЯЦ (A1) =6
```

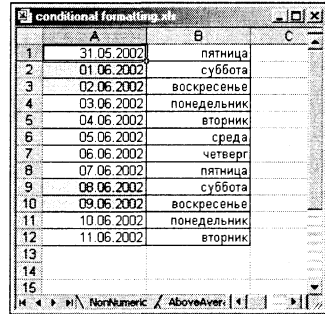


The screenshot shows a spreadsheet with columns A through E and rows 1 through 14. The data in the first 12 rows is as follows:

	A	B	C	D	E
1	1	12	23	34	
2	36	13	24	35	
3	3	14	25	2	
4	4	15	26	37	
5	5	16	27	38	
6	41	17	28	6	
7	7	18	12	40	
8	8	19	30	41	
9	9	20	31	42	
10	10	43	32	21	
11	11	22	33	44	
12	29	23	34	45	

Cells with values greater than the average (32.5) are highlighted in light blue.

Рис. 19.6. Использование условного форматирования для выделения цветом ячеек со значением, большим среднего по диапазону



The screenshot shows a spreadsheet with columns A through C and rows 1 through 15. The data in the first 12 rows is as follows:

	A	B	C
1	31.05.2002	пятница	
2	01.06.2002	суббота	
3	02.06.2002	воскресенье	
4	03.06.2002	понедельник	
5	04.06.2002	вторник	
6	05.06.2002	среда	
7	06.06.2002	четверг	
8	07.06.2002	пятница	
9	08.06.2002	суббота	
10	09.06.2002	воскресенье	
11	10.06.2002	понедельник	
12	11.06.2002	вторник	

Cells with dates in June (rows 11-12) are highlighted in light blue.

Рис. 19.7. Использование условного форматирования для выделения цветом дат, приходящихся на выходные дни

В данной формуле предполагается, что ячейка A1 находится в верхнем левом углу выбранного диапазона. Работа формулы основывается на использовании функции МЕСЯЦ, возвращающей номер месяца для выбранной даты.



Функция МЕСЯЦ не различает ячейки с датами и без дат. Другими словами, данная функция применяется ко всем ячейкам, даже если в них не содержится дата.

Определение текущей даты

Функция Excel СЕГОДНЯ возвращает текущую дату. Если в рабочем листе имеется ряд дат, с помощью условного форматирования вы можете определить значения, относящиеся к текущей дате. Приведенная ниже формула условного форматирования применяет форматирование лишь к ячейкам, содержащим текущую дату. Здесь предполагается, что перед вводом формулы условного форматирования был выбран диапазон, начинающийся с ячейки A1.

```
=A1=СЕГОДНЯ()
```

Определение дат выходных дней

Функция Excel ДЕНЬНЕД возвращает целое число, представляющее собой порядковый номер дня недели (1 — воскресенье, 2 — понедельник и так далее). Данная функция может использоваться в формуле условного форматирования для определения выходных дней. Следующая формула применяет форматирование к ячейкам, содержащим даты, приходящиеся на субботу или воскресенье.

=ИЛИ (ДЕНЬНЕД (A1) =7 ; ДЕНЬНЕД (A1) =1)

В данной формуле используется функция ИЛИ, возвращающая значение ИСТИНА, если функция ДЕНЬНЕД возвращает либо значение 7, либо 1. Вы можете заметить, что функция ДЕНЬНЕД возвращает значение 7, если ее аргументом является пустая ячейка. Поэтому, если в диапазоне есть пустые ячейки, формулу необходимо переписать следующим образом:

=ЕСЛИ (ЕПУСТО (A1) ; " " ; ИЛИ (ДЕНЬНЕД (A1) =7 ; ДЕНЬНЕД (A1) =1))

Скрытие ошибочных значений

Условное форматирование может использоваться для того, чтобы скрыть ошибочные значения в ячейках. В данном случае, скрытие содержимого ячейки состоит в присвоении шрифта цвета фона. Следующая формула применяет форматирование к ячейкам, если они возвращают ошибочное значение (например, #ДЕЛ/0!):

=ЕОШИБКА (A1)

Применяемое форматирование делает цвет заливки и шрифта одинаковыми.



Хотя методика применения одного цвета заливки и шрифта работает, данный способ не является лучшим с точки зрения отображения ошибочных значений. Ячейки, содержащие ссылки на ошибочную ячейку, тоже будут отображать ошибку, да и пользователь легко может сам изменить цвет фона. Во многих случаях лучшим решением будет использование функции ЕСЛИ, отображающей пустую строку, если формула возвращает ошибку. Следующая формула отображает пустую строку, если выражение B1/C1 приводит к возникновению ошибки:

=ЕСЛИ (ЕОШ (B1/C1) ; " " ; (B1/C1))



В Excel 2002 появилась новая возможность определения того, как ошибочные значения будут выводиться на печать. Ошибка может выводиться на печать как пустое место, черта или обозначаться символами #. Управление данным представлением осуществляется на вкладке Лист диалогового окна Параметры страницы.

Определение максимального значения в диапазоне

Функция Excel МАКС возвращает максимальное содержащееся в диапазоне значение. Если вы хотите, чтобы данное значение выделялось среди остальных, можете использовать формулу условного форматирования, подобную приведенной ниже:

=A1=МАКС (\$A\$1 : \$A\$30)

В данном случае условное форматирование применяется ко всем ячейкам в диапазоне A1:A30, а отформатированным всегда будет максимальное значение в нем. Данную формулу, конечно же, можно легко модифицировать для использования с функцией МИН (возвращающей минимальное значение в диапазоне).

Определение трех наибольших значений в диапазоне

Функция Excel **НАИБОЛЬШИЙ** возвращает n-ое наибольшее значение в диапазоне (n задается как второй аргумент). Следующая формула применяет форматирование к трем наибольшим значениям в диапазоне A1:A30: данная формула возвращает значение ИСТИНА для ячейки, значение которых больше или равно третьему наибольшему значению в диапазоне.

$=A1 >= \text{НАИБОЛЬШИЙ}(\$A\$1 : \$A\$30 ; 3)$

Создание чередующейся заливки строк

Приведенная ниже формула условного форматирования была применена к диапазону A1:D18 для создания чередующейся заливки строк (рис. 19.8). Данная формула весьма полезна для повышения читабельности электронных таблиц.

$=\text{ОСТАТ}(\text{СТРОКА}() ; 2) = 0$

	A	B	C	D	E
1	410	425	808	972	
2	506	568	329	792	
3	511	24	399	266	
4	202	271	849	546	
5	867	737	69	412	
6	913	497	115	261	
7	331	3	664	13	
8	363	606	604	409	
9	923	711	467	193	
10	93	371	751	92	
11	227	805	695	345	
12	32	426	459	261	
13	601	390	228	455	
14	962	649	84	134	
15	327	942	872	152	
16	92	377	756	165	
17	762	219	950	189	
18	778	665	645	504	

Рис. 19.8. Использование условного форматирования для применения форматирования к чередующимся строкам

	A	B	C	D	E	F	G	H
1	940	980	940	217	94	177	940	857
2	891	771	233	705	131	625	624	344
3	568	531	71	242	765	596	228	969
4	626	725	304	659	682	494	560	331
5	710	526	125	141	464	333	380	918
6	521	288	774	825	116	665	220	41
7	651	196	889	871	812	392	228	623
8	212	641	269	772	762	63	65	612
9	917	844	667	959	171	10	419	368
10	923	443	206	96	782	775	465	674
11	484	453	48	627	451	966	962	160
12	136	399	49	919	903	228	376	658
13	238	805	42	110	628	279	69	291
14	155	23	529	996	882	94	271	151
15	715	558	111	126	338	848	104	929
16	939	33	605	124	963	812	336	435
17	271	367	639	184	438	621	711	425
18	262	247	412	33	787	965	360	917
19	624	801	628	297	842	808	332	818
20	595	101	182	202	138	383	904	715
21	645	841	449	280	416	915	205	497
22	347	738	326	149	231	880	257	70

Рис. 19.9. Использование условного форматирования для создания эффекта шахматной доски

В данной формуле используются функция **СТРОКА** (возвращающая номер строки) и функция **ОСТАТ** (возвращающая остаток деления первого аргумента на второй). Для ячеек четных строк используется функция **ОСТАТ**, возвращающая значение 0, и к ячейке применяется форматирование. Для создания чередующейся заливки столбцов необходимо вместо функции **СТРОКА** использовать функцию **СТОЛБЕЦ**.

Для получения других типов заливки могут использоваться различные вариации данной формулы условного форматирования. Вот пример формулы условного форматирования, применяющей заливку к каждой третьей строке:

$=\text{ОСТАТ}(\text{СТРОКА}() ; 3) = 0$

Следующая формула условного форматирования создает чередующуюся заливку групп из четырех строк (четыре строки имеют заливку, четыре последующие — нет):

$=\text{ОСТАТ}(\text{ЦЕЛОЕ}((\text{СТРОКА}()) / 4 + 1 ; 2))$

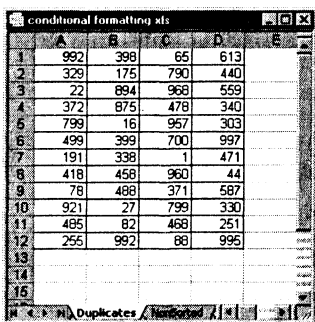
Вам нужно создать “шахматную” заливку, показанную на рис. 19.9? Вот формула, которая легко справится с этой задачей:

$=\text{ОСТАТ}(\text{СТРОКА}() ; 2) = \text{ОСТАТ}(\text{СТОЛБЕЦ}() ; 2)$

Определение повторяющихся значений в диапазоне

Иногда может потребоваться найти в диапазоне одинаковые значения (рис. 19.10). Для этого можно использовать формулу условного форматирования, подобную приведенной ниже. В данном примере форматирование применяется ко всем ячейкам, значение которых в диапазоне A1:D12 не уникально.

`=СЧЕТЕСЛИ(A1:D12;A1)>1`

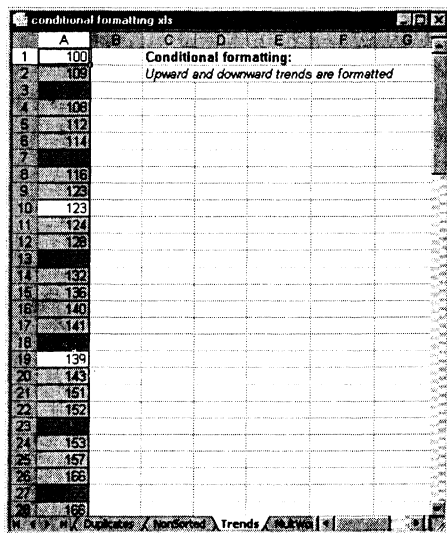


The screenshot shows a spreadsheet with columns A through E and rows 1 through 16. The data in columns A, B, C, and D is as follows:

	A	B	C	D
1	992	398	65	613
2	329	175	790	440
3	22	894	968	559
4	372	875	478	340
5	799	16	957	303
6	499	399	700	997
7	191	338	1	471
8	418	458	960	44
9	78	488	371	587
10	921	27	799	330
11	485	82	468	251
12	255	992	88	995
13				
14				
15				
16				

The status bar at the bottom indicates 'Duplicates' and 'Nonsorted'.

Рис. 19.10. Использование условного форматирования для определения повторяющихся значений в диапазоне



The screenshot shows a spreadsheet with columns A through G and rows 1 through 27. The data in column A is as follows:

	A
1	100
2	109
3	
4	108
5	112
6	114
7	
8	116
9	129
10	123
11	124
12	128
13	
14	132
15	136
16	140
17	141
18	
19	139
20	143
21	151
22	152
23	
24	153
25	157
26	166
27	168

The status bar at the bottom indicates 'Trends' and 'Nonsorted'.

Рис. 19.11. Использование условного форматирования для определения тенденций к повышению и понижению

Чтобы применить форматирование только к неповторяющимся значениям, используйте следующую формулу:

`=СЧЕТЕСЛИ(A1:D12;A1)=1`

Определение неупорядоченных значений в диапазоне

Если есть диапазон, состоящий из одного столбца, значения в котором должны располагаться в возрастающем порядке, с помощью формулы условного форматирования можно легко определить значения, нарушающие данный порядок. В примере предполагается, что вы упорядочиваете значения, начинающиеся в ячейке A1. Выделите диапазон значений, начинающийся с ячейки A2, и определите следующую формулу условного форматирования:

`=A2<A1`

Условное форматирование будет применено к любой ячейке, значение в которой меньше, чем значение в ячейке, находящейся над ней.

Определение тенденции (к повышению или понижению)

Иногда может оказаться очень полезно визуально определить в столбце данных тенденцию к повышению и понижению. В примере предполагается, что данные начинаются в ячейке A1. Выделите диапазон значений, начинающийся с ячейки A2, и определите два следующих условия:

`=A2>A1`

`=A2<A1`

Определите различные форматы для каждого из условий; это позволит задать характер тенденций без построения диаграммы. На рис. 19.11 показан пример использования такого форматирования.

Определение ячеек, содержащих более одного слова

Условное форматирование может использоваться при работе с текстом. К примеру, следующая формула условного форматирования использована для применения форматирования к ячейкам, содержащим больше, чем одно слово:

=ДЛСТР (СЖПРОБЕЛЫ (A1)) -ДЛСТР (ПОДСТАВИТЬ (A1 ; " " ; " ")) > 0

В данной формуле предполагается, что выбранный диапазон начинается с ячейки A1. Работа формулы основана на подсчете пробелов в ячейке (функция СЖПРОБЕЛЫ используется для удаления нескольких следующих один за другим пробелов). Если результат больше 1, формула возвращает значение ИСТИНА и условное форматирование применяется.

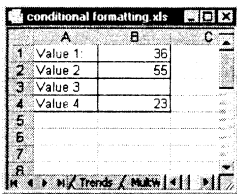
Определение ячеек, содержащих определенный символ

Следующая формула условного форматирования применяет форматирование к ячейкам (начиная с ячейки A1), содержащим букву "a" (записанную в верхнем или нижнем регистре):

=ДЛСТР (A1) -ДЛСТР (ПОДСТАВИТЬ (ПРОПИСН (A1) ; "a" ; " ")) > 0

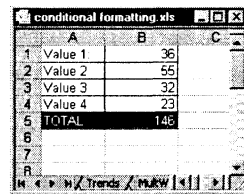
Отображение результата только при условии ввода всех данных

В приведенном примере условное форматирование используется для предотвращения отображения результата до ввода всех необходимых данных. Формула в ячейке B5 (рис. 19.12) вычисляет сумму четырех находящихся выше значений. Идея состоит в том, чтобы скрывать сумму до тех пор, пока не будут введены все четыре значения.



	A	B	C
1	Value 1	36	
2	Value 2	55	
3	Value 3	32	
4	Value 4	23	
5			
6			
7			
8			

Рис. 19.12. Условное форматирование скрывает содержимое ячеек A5:B5, пока не введены значения во все ячейки диапазона B1:B4



	A	B	C
1	Value 1	36	
2	Value 2	55	
3	Value 3	32	
4	Value 4	23	
5	TOTAL	146	
6			
7			
8			

Рис. 19.13. Содержимое ячеек A5:B5 становится видимым только тогда, когда все ячейки диапазона B1:B4 содержат данные

Выберите ячейки A5:B5 и отформатируйте их так, чтобы цвет фона и шрифта совпадали (например, сделайте цвет шрифта белым). Это сделает данные ячейки невидимыми. Не изменяя выделение введите следующую формулу условного форматирования:

=СЧЕТ (\$B\$1:\$B\$4) = 4

Формула возвратит значение ИСТИНА только тогда, когда все ячейки в диапазоне B1:B4 будут не пустыми. Задайте форматирование. Можно, например, сделать цвет фона черным. На рис. 19.13 показан результат, получаемый после введения всех необходимых данных.

Определение положительных изменений

На рис. 19.14 представлены результаты двух тестов группы студентов. Условное форматирование используется для выделения цветом строк, в которых результат студента во втором тесте выше, чем в первом.

Для диапазона A2:C12 используется следующая формула условного форматирования:

= \$C2 > \$B2

Обратите внимание, что в данной формуле используются ссылки смешанного типа. Часть ссылки, относящаяся к столбцу, абсолютна, а часть, относящаяся к строке, относительна.

	A	B	C	D
1	Student	Pre-test	Post-test	
2	Anna	89	93	
3	Bill	91	90	
4	Chris	75	81	
5	Darla	65	73	
6	Ernie	90	81	
7	Frank	93	100	
8	George	89	89	
9	Hilda	99	98	
10	Ishmael	54	69	
11	James	80	75	
12	Keith	89	88	
13				
14				
15				

Рис. 19.14. Использование условного форматирования для определения студентов, повысивших результат на повторном тесте

Использование пользовательских функций в формулах условного форматирования

Формулы условного форматирования могут работать также и с пользовательскими функциями рабочего листа, созданными с помощью VBA. В данном подразделе приведены некоторые примеры.



В части 5 приведен обзор VBA и описаны методы создания пользовательских функций рабочего листа.

Определение ячеек формулы

Странно, но в Excel нет функции, определяющей, содержит ли выбранная ячейка формулу или нет. В случаях, когда средств Excel оказывается недостаточно, на помощь приходит язык VBA. Приведенные ниже функции VBA используют свойство HasFormula. Функция, введенная в модуль VBA, возвращает значение ИСТИНА, если ячейка (определенная в качестве аргумента) содержит формулу; в противном случае возвращается значение ЛОЖЬ.

```
Function ISFORMULACELL(cell) As Boolean
    ISFORMULACELL = cell.HasFormula
End Function
```

После ввода такой функции в модуль VBA ее можно использовать в формулах рабочего листа. Например, следующая формула возвращает значение ИСТИНА, если ячейка A1 содержит формулу:

=ISFORMULACELL(A1)

Такая функция также может использоваться и в формуле условного форматирования. Например, в рабочем листе, представленном на рис. 19.15, условное форматирование используется для выделения цветом ячеек, содержащих формулу.



Найти формулу можно также и с помощью команды Правка ⇨ Перейти. При выборе данной команды отображается диалоговое окно Переход. Щелкните на кнопке Выделить для перехода к диалоговому окну Выделение группы ячеек. Затем выберите переключатель формулы и щелкните на кнопке ОК. При этом будут выделены все ячейки, содержащие формулы.

Определение ячеек, содержащих дату

В Excel также нет функций, способных определить, содержится ли в ячейке дата. Следующая VBA-функция, использующая внутреннюю функцию IsDate, позволяет преодолеть данное ограничение. Пользовательская функция HASDATE возвращает значение ИСТИНА, если в ячейке содержится дата.

```
Function HASDATE(cell) As Boolean
    HASDATE = IsDate(cell)
End Function
```

Данная функция может использоваться для усовершенствования формулы условного форматирования, представленной ранее в данной главе (см. “Нахождение дат, относящихся к определенному месяцу” и “Определение дат выходных дней”). Ни одна из представленных формул не способна определить, содержит ячейка дату или обычные значения. Вы можете использовать функцию И для того, чтобы форматирование применялось гарантировано лишь к ячейкам с датами.

Следующая формула условного форматирования применяет форматирование к ячейке A1, если в ней содержится дата, и дата относится к июню:

```
=И (HASDATE (A1) ; МЕСЯЦ (A1) = 6)
```

Следующая формула условного форматирования применяет форматирование к ячейке A1, если в ней содержится дата, и дата приходится на выходной день:

```
=И (HASDATE (A1) ; ИЛИ (ДЕНЬНЕД (A1) = 7 ; ДЕНЬНЕД (A1) = 1))
```

Нахождение связанной формулы

У вас может возникнуть необходимость определения ячеек, содержащих связанные формулы (формулы, использующие ссылки на другие рабочие книги). Следующая VBA-функция возвращает значение ИСТИНА, если в ячейке находится формула, содержащая внешнюю ссылку. Функция HASLINK использует универсальный оператор Like для определения того, содержит ли формула квадратные скобки.

```
Function HASLINK(cell)
    If cell.HasFormula Then
        HASLINK = cell.Formula Like "[[]]"
    Else
        HASLINK = ЛОЖЬ
    End If
End Function
```

Для применения условного форматирования к ячейкам, содержащим ссылки можно создать формулу, подобную следующей:

```
=HASLINK (A1)
```



Функция HASLINK несовершенна. Иногда она может определять в формуле наличие связи ошибочно. К примеру, в следующей формуле есть набор квадратных скобок, но она не является связанной формулой. Однако у функции HASLINK свое мнение по данному вопросу.

```
= "[&A1&]"
```

Определение недостоверных данных

Возможны ситуации, когда введенные данные должны подчиняться определенным правилам, те же, которые не удовлетворяют заданным условиям, удобно выделить с помощью определенного форматирования. Предположим, что в таблице описываются семизначные номера деталей: четыре первые знака — буквы, затем после черты следует две цифры. Например: КТПП-09, АШЕР-43.

Вы можете написать формулу условного форматирования, которая проверяет, соответствует ли номер детали такой структуре. Но такая формула будет весьма сложной и длинной.

Для большей простоты напишите пользовательскую VBA-функцию рабочего листа. VBA-оператор Like выполняет такие сравнения относительно просто. Следующая VBA-функция возвращает значение ИСТИНА, если ее аргумент не соответствует означенным выше правилам для номера детали:

```
Function INVALIDPART(n) As Boolean
    If n Like "[A-Z][A-Z][A-Z][A-Z]-##" Then
        INVALIDPART = False
    Else
        INVALIDPART = True
    End If
End Function
```

После того как данная функция определена в VBA-модуле, и чтобы применить специальное форматирование, если ячейка A1 содержит недостоверный номер детали, нужно ввести формулу:

```
=INVALIDPART(A1)
```

На рис. 19.16 показан диапазон, использующийся в формуле условного форматирования функции INVALIDPART. Ячейки, содержащие некорректные номера деталей, имеют цветную заливку.

Во многих случаях можно просто воспользоваться возможностями средства Excel проверки вводимых значений, о котором пойдет речь далее.

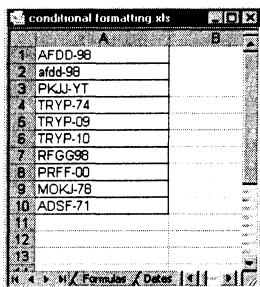


Рис. 19.16. Использование условного форматирования для выделения ячеек с недостоверными данными

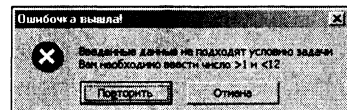


Рис. 19.17. Отображение сообщения при неверном вводе данных пользователем

Проверка вводимых значений

Средство проверки вводимых значений, появившееся в версии Excel 97, во многих отношениях напоминает средство условного форматирования. Данная функция позволяет установить определенные правила, определяющие, какое содержимое может вводиться в ячейку. Например, можно задать условие для того, чтобы ограничить вводимые данные в пределах от 1 до 12. Если пользователь делает ошибочную запись, выводится настраиваемое сообщение, подобное показанному на рис. 19.17.

Как и в случае со средством условного форматирования для определения критерия проверки вводимых значений может использоваться формула.



Средству проверки достоверности данных присущи те же недостатки, что и средству условного форматирования: если пользователь скопирует ячейку и вставит ее в ячейку, использующую данное средство, правило проверки достоверности будет удалено. Следовательно, после этого ячейка может воспринимать любые данные.

Определение критерия достоверности данных

Для того чтобы определить тип данных, разрешенный для ячейки или диапазона, выполните следующие действия:

1. Выделите ячейку или диапазон.
2. Выберите команду **Данные**⇒**Проверка**. Excel отобразит диалоговое окно **Проверка вводимых значений**.
3. Щелкните на вкладке **Параметры** (см. рис. 19.18).

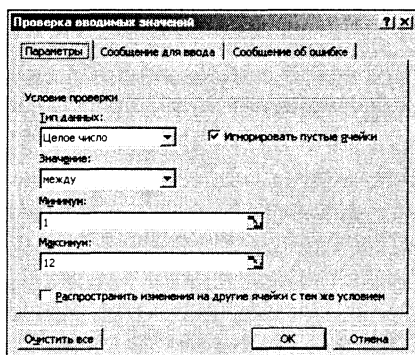


Рис. 19.18. Вкладка **Параметры** диалогового окна **Проверка вводимых значений**

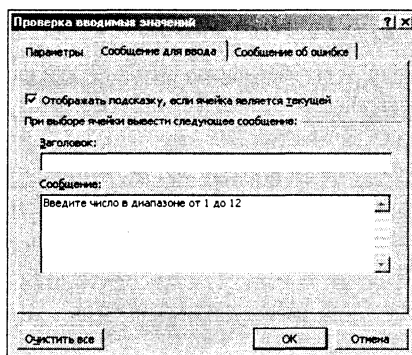


Рис. 19.19. Вкладка **Сообщение для ввода** диалогового окна **Проверка вводимых значений**

4. Выберите значение из раскрывающегося списка **Тип данных**. Для того чтобы определить формулу, необходимо выбрать значение **Другой**.
5. Определите условия, сделав выбор из списка **Значение**. От вашего выбора зависит, к каким еще элементам управления вы получите доступ.
6. Щелкните на вкладке **Сообщение для ввода** (рис. 19.19) и определите сообщение, которое будет выводиться при выборе ячейки пользователем. Данная необязательная возможность используется для того, чтобы помочь пользователю определить, какие данные должны вводиться.
7. Щелкните на вкладке **Сообщение об ошибке** (см. рис. 19.20) и определите сообщение, которое будет выводиться при вводе пользователем некорректного значения. С помощью раскрывающегося списка **Вид** можно определить, какие действия будут доступны пользователю при вводе неверных данных. Для того чтобы предотвратить ввод некорректного значения, выберите опцию **Останов**. Данное действие выполнять необязательно.
8. Щелкните на кнопке **ОК**.

После выполнения данных действий ячейка или диапазон будут содержать указанные условия проверки вводимых значений.

Типы возможных условий проверки вводимых значений

Вкладка Параметры диалогового окна Проверка вводимых значений позволяет определить условия для следующих типов данных:

- ◆ Любое значение. Выбор данного значения приводит к удалению существующих правил проверки вводимых значений. Однако отметьте, что если не сбросить флажок на вкладке Сообщение для ввода, подсказка, определенная на ней, будет по-прежнему отображаться.
- ◆ Целое число. Пользователь должен ввести целое число. Диапазон допустимых значений определяется с помощью раскрывающегося списка Значения. Например, можно указать, что вводимое значение должно быть больше или равно 100.
- ◆ Действительное. Пользователь должен ввести число. Диапазон допустимых значений определяется с помощью раскрывающегося списка Значения. Например, можно задать, что вводимое значение должно быть больше или равно 0 и меньше или равно 1.
- ◆ Список. Пользователь должен выбрать из предоставляемого вами списка значений. Диапазон, используемый для списка, определяется в поле Источник (диапазон должен представлять собой строку или столбец). Если список небольшой, его можно ввести непосредственно в поле Источник (элементы списка разделяются символом, определенным в настройках региональных стандартов системы — точкой с запятой, если используются настройки для России или Украины).

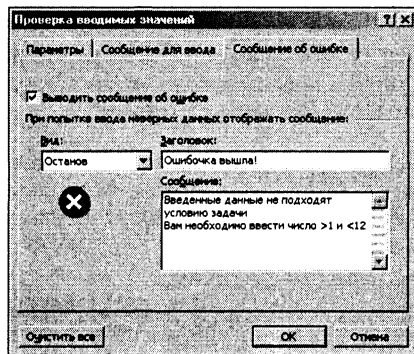


Рис. 19.20. Вкладка Сообщение об ошибке диалогового окна Проверка вводимых значений



Выбранный для списка диапазон должен находиться на том же листе. Если список находится в другом рабочем листе, можно определить для такого диапазона имя, которое затем использовать в качестве источника (поместив перед ним знак равенства). Например, если список содержится в диапазоне с именем МойСписок, введите следующее:
=МойСписок

- ◆ Дата. Пользователь должен ввести дату. Диапазон допустимых дат определяется с помощью раскрывающегося списка Значения. Например, можно задать, что вводимая дата должна быть больше или равна 1 января 2001 года и меньше или равна 31 декабря 2001.
- ◆ Время. Пользователь должен ввести время. Диапазон допустимых значений времени определяется с помощью раскрывающегося списка Значения. Например, можно задать, что вводимое время должно быть больше, чем 12:00.
- ◆ Длина текста. Длина данных (цифровых или символьных) ограничена. Допустимая длина определяется с помощью раскрывающегося списка Значения. Например, можно задать, что длина вводимых данных равна 1 (один символ).
- ◆ Другой. Логическая формула, определяющая, являются ли введенные пользователем данные допустимыми. Формула задается непосредственно в поле Формула, или же

определяется как ссылка на ячейку, содержащую формулу. В данной главе приведены примеры некоторых полезных формул.

Вкладка Параметры диалогового окна Проверка вводимых значений содержит еще два флажка:

- ◆ Игнорировать пустые ячейки. Если флажок установлен, допускаются пустые записи.
- ◆ Распространить изменения на другие ячейки с тем же условием. Если флажок установлен, вносимые изменения применяются ко всем другим ячейкам, содержащим исходные условия проверки достоверности данных.

Важно понять, что даже при включенной проверке вводимых значений пользователь может ввести ошибочные данные. Если в раскрывающемся списке Вид вкладки Сообщение об ошибке диалогового окна Проверка вводимых значений выбрано любое из значений, кроме Останов, недопустимые значения *могут* быть введены.



После того как данные введены, можно найти записи, выходящие за установленные пределы. Если щелкнуть на кнопке Обвести неверные данные панели инструментов Зависимости, ячейки, содержащие неверные данные, будут обведены (рис. 19.21). Если неверные данные исправить, обводка будет снята.

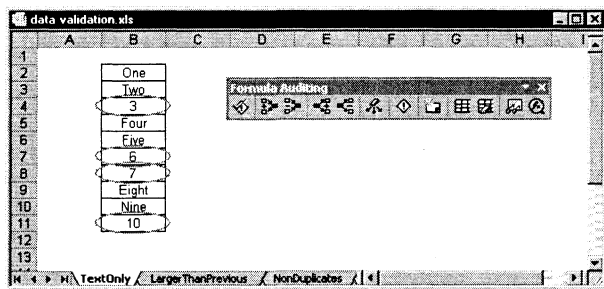


Рис. 19.21. Обведение отдельных записей (ячеек, содержащих неверные значения)

Использование формул в условиях проверки вводимых значений

В простых случаях проверки вводимых значений работа с данным средством достаточно проста и понятна. Но подлинная сила средства проявляется при использовании формул проверки вводимых значений.



Вводимая формула должна быть логической, то есть возвращать либо значение ИСТИНА, либо ЛОЖЬ. Если формула равна значению ИСТИНА, данные считаются достоверными и остаются в ячейке. Если формула равна значению ЛОЖЬ, появляется диалоговое окно с сообщением, определенным на вкладке Сообщение об ошибке диалогового окна Проверка вводимых значений.

Как отмечалось ранее, для определения формулы в диалоговом окне Проверка вводимых значений необходимо выбрать из раскрывающегося списка Тип данных значение Другой. В поле Формула можно либо напрямую ввести формулу, либо указать ссылку на ячейку, содержащую необходимую формулу.

Ссылка, используемая во введенной формуле, является относительной с привязкой к левой верхней ячейке выбранного диапазона. Здесь имеет место полная аналогия с формулами условного форматирования.

Использование формул проверки вводимых значений для ввода только определенных значений

В каждом из приведенных ниже примеров используется формула, вводимая непосредственно в диалоговом окне Проверка вводимых значений. Данные формулы могут использоваться для того, чтобы ячейка или диапазон содержали только текст, определенное значение, не дублируемые записи или текст, начинающийся с определенной буквы.

Принятие только текста

Для того чтобы диапазон воспринимал лишь текст (не значения) воспользуйтесь следующей формулой проверки вводимых значений:

=ТЕКСТ (A1)

В данной формуле предполагается, что ячейка A1 находится в левом верхнем углу выбранного диапазона.

Принятие только значений, больших, чем в предыдущей ячейке

Следующая формула проверки вводимых значений позволяет пользователям ввести значение, только если оно больше значения, находящегося в соседней сверху ячейке:

=A2>A1

В данной формуле предполагается, что ячейка A2 находится в левом верхнем углу выбранного диапазона. Заметьте, что данная формула не используется для ячеек в первой строке.

Принятие не дублированных данных

Следующая формула проверки вводимых значений не позволяет пользователям вводить повторяющиеся записи в диапазоне A1 : C20:

=СЧЕТЕСЛИ (\$A\$1 : \$C\$20 ; A1) = 1

В данной формуле предполагается, что ячейка A1 находится в левом верхнем углу выбранного диапазона. Заметьте, что первый аргумент функции СЧЕТЕСЛИ является абсолютной ссылкой, а второй он изменяется для каждой из ячеек. На рис. 19.22 показана формула проверки значений в действии (формула вызывает диалоговое окно с сообщением об ошибке).

Принятие текста, начинающегося с буквы "А"

Следующая формула проверки вводимых значений демонстрирует, как можно выполнить проверку введения определенного символа. В данном случае формула используется для гарантии того, что вводимая пользователем запись будет текстовой строкой, начинающейся с буквы "А" (в верхнем или нижнем регистре).

=ЛЕВСИМВ (A1) = " а "

В данной формуле предполагается, что ячейка A1 находится в левом верхнем углу выбранного диапазона.

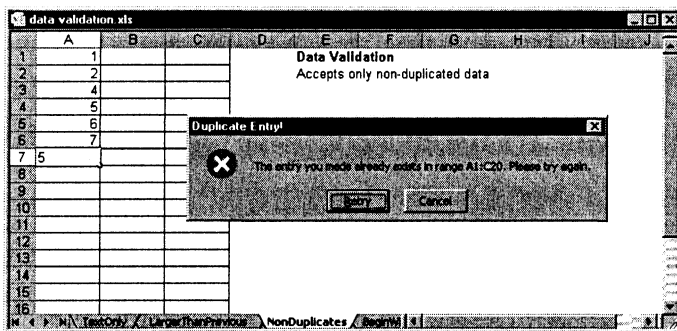


Рис. 19.22. Использование проверки вводимых значений для исключения возможности ввода повторяющихся значений

Следующая формула является вариацией на тему предыдущей. Здесь формула обеспечивает гарантию того, что запись начинается с буквы "А" и содержит ровно пять символов.

=ЛЕВСИМВ (А1 ; "А????") =1

Использование пользовательских функций в формулах проверки вводимых значений

Ранее в данной главе приводилось описание того, как можно использовать пользовательские VBA-функции в формулах условного форматирования. По известным только разработчикам причинам Excel не позволяет использовать такую возможность в случае формул проверки вводимых значений. Если вы попытаетесь сделать это, на экран будет выведено сообщение об ошибке.

Чтобы обойти данное ограничение, вы можете использовать пользовательские функции в формулах ячейки, а затем определить формулу проверки вводимых значений со ссылкой на такую ячейку.



Часть VI данной книги посвящена пользовательским VBA-функциям.

Резюме

В данной главе рассмотрены возможности двух полезных средств, доступных, начиная с версии Excel 97: условное форматирование и проверка вводимых значений. В ней также приведено множество примеров совместного использования формул и двух данных средств.

Следующая глава посвящена тому, что я называю мегаформулой.

Создание мегаформул

В этой главе...

- ◆ Что такое мегаформула?
- ◆ Создание мегаформулы: простой пример
- ◆ Примеры мегаформул
- ◆ Преимущества и недостатки использования мегаформул
- ◆ Резюме

Данная глава посвящена описанию полезной методики, позволяющей комбинировать несколько формул в одну, которую я называю *мегаформулой*. Такой подход позволяет устранить необходимость использования промежуточных формул и может даже повысить скорость вычислений. Недостатком является то, что результирующая формула практически непостижима и ее редактирование не всегда представляется возможным.

Что такое мегаформула?

Часто при работе с электронными таблицами для получения желаемого результата приходится использовать промежуточные формулы. Другими словами, формула может зависеть от других формул, которые в свою очередь могут зависеть еще от следующих. После того как удалось заставить все формулы работать правильно, можно удалить промежуточные формулы и создать одну единственную (более сложную) формулу. За неимением более подходящего термина я называю такие формулы *мегаформулами*.

В чем состоит преимущество использования мегаформул? Они занимают меньшее число ячеек, а также могут повысить скорость пересчета. Кроме того, вы можете произвести впечатление на людей, известных своим умением “формулопостроения”. Недостатки? Формула с большой вероятностью не будет подлежать расшифровке или модификации даже со стороны человека, ее создавшего.



Описанные в данной главе методы оказали помощь в создании многих сложных формул, встречающихся во всей данной книге.

Ограничением техники мегаформул является то, что формулы Excel не могут содержать более 1024 символов.

Создание мегаформулы: простой пример

Создание мегаформулы большей частью состоит в копировании текста формулы и его вставки в другую формулу. Давайте начнем с относительно простого примера. Изучите электронную таблицу, показанную на рис. 20.1. В данной таблице используются формулы для вычисления займов под залог недвижимости.

В разделе результатов рабочего листа используется информация, вводимая в разделе ячеек вводимых данных, и содержатся формулы, показанные в табл. 20.1.

Таблица 20.1. Формулы, используемые для вычислений

Ячейка	Формула	Описание
C10	=C4*C5	Вычисляет первый взнос
C11	=C4-C10	Вычисляет сумму займа
C12	=ПЛТ (C7/12; C6; -C11)	Вычисляет месячные выплаты
C13	=C12*C6	Вычисляет суммарные выплаты
C14	=C13-C11	Вычисляет выплаченный процент

Предположим, что нас на самом деле интересует выплаченный процент (ячейка C14). Вы, конечно же, можете скрыть строки, содержащие лишнюю информацию. Однако существует еще и возможность создания единственной формулы, которая бы выполняла работу нескольких промежуточных.

Формула, вычисляющая выплаченный процент, зависит от формул в ячейках C11 и C13 (прямая зависимость). Кроме этого, формула в ячейке C13 зависит от формулы в ячейке C12, которая в свою очередь зависима от ячейки C11. Таким образом, для вычисления общего дохода используется пять формул. Ниже приведено описание действий, которые необходимо выполнить для того, чтобы создать единственную формулу, вычисляющую общий процент (без использования промежуточных формул). В ячейке C14 содержится следующая формула:

$$=C13-C11$$

Ниже приводится описание того, как преобразовать данную формулу в мегаформулу:

1. Ссылку на ячейку C13 замените формулой, содержащейся в данной ячейке. Перед этим заключите данную формулу в круглые скобки. Теперь формула в ячейке C14 имеет вид:

$$=(C12*C6)-C11$$
2. Ссылку на ячейку C12 замените формулой, содержащейся в данной ячейке. Теперь формула в ячейке C14 имеет вид:

$$=(ПЛТ (C7/12; C6; -C11) *C6) -C11$$
3. Ссылку на ячейку C11 замените формулой, содержащейся в данной ячейке. Перед копированием формулы, ее необходимо заключить в круглые скобки. Теперь формула в ячейке C14 имеет вид:

$$=(ПЛТ (C7/12; C6; -(C4-C10)) *C6) - (C4-C10)$$
4. Ссылку на ячейку C10 замените формулой, содержащейся в данной ячейке. Перед копированием формулы, ее необходимо заключить в круглые скобки. После этого формула в ячейке C14 примет вид:

$$=(ПЛТ (C7/12; C6; -(C4-(C4*C5)) *C6) - (C4-(C4*C5))$$

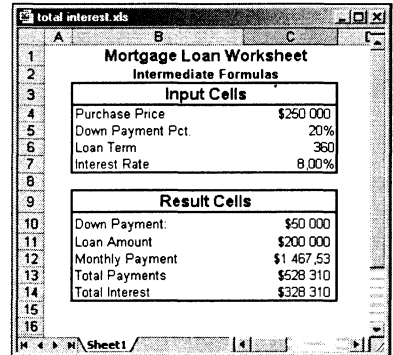


Рис. 20.1. В данной электронной таблице для вычисления займов под залог недвижимости используется значительное количество формул

Теперь формула содержит ссылки только на ячейки вводимых данных. Можно не опасаться удалить формулы в ячейках C10:C13. Теперь работу, возлагаемую раньше на промежуточные формулы, выполняет одна единственная мегаформула.

Если вы не являетесь специалистом мирового уровня по написанию формул Excel, маловероятно, что вы сможете получить подобную формулу без предварительного создания промежуточных формул.



Предыдущий пример был приведен для демонстрации процесса создания мегаформулы. Данная техника *не* является самым эффективным способом вычисления общего процента по займу. Для выполнения подобных вычислений в Excel имеется более прямой путь: вы можете использовать одну из функций из надстройки "Пакет анализа".

Копирование текста из формулы

Создание мегаформулы состоит в копировании текста формулы и замене им ссылки на ячейку. Для копирования содержимого формулы активизируйте ячейку и нажмите клавишу <F2>. Затем выделите текст формулы (без знака равенства), используя сочетания клавиш <Shift+Home>, а затем <Shift+→>. Затем, нажав сочетание клавиш <Ctrl+C>, скопируйте выделенный текст в буфер обмена. Активизируйте ячейку, содержащую мегаформулу, и нажмите клавишу <F2>. Используйте клавиши со стрелками совместно с клавишей <Shift> для выделения заменяемой ссылки. Наконец, используйте сочетание клавиш <Ctrl+V> для замены выделенного текста содержимым буфера обмена.

В некоторых случаях для поддержки корректной работы формулы копируемый текст должен заключаться в круглые скобки. Если после вставки текста в формулу, возвращаемое ее значение изменяется, отмените вставку, нажав <Ctrl+Z>. Заключите копируемую формулу в скобки и попробуйте вставить снова.

Создание мегаформулы по существу состоит в подстановке текста формул вместо ссылок на ячейки. Подстановки осуществляются до тех пор, пока в мегаформуле не останутся ссылки только на ячейки вводимых данных. На каждом шаге вы имеете возможность проверить свою работу, следя за тем, не изменяется ли возвращаемое формулой значение. В предыдущем примере несколько раз возникала необходимость заключать копируемую формулу в круглые скобки.

Примеры мегаформул

Данный раздел содержит три примера использования мегаформул. Данные примеры являются демонстрацией того, как применение мегаформул может рационализировать выполнение множества задач, включая очистку списков сотрудников путем удаления вторых имен и инициалов, возвращение положения последнего символа пробела в строке и определение действительности номера кредитной карточки.

Использованию мегаформул для удаления вторых имен и инициалов

Рассмотрим рабочий лист со столбцом, содержащим фамилии и имена людей (см. рис. 20.2). Предположим, что рабочий лист содержит тысячи фамилий, и вам необходимо подчистить все записи, удалив вторые

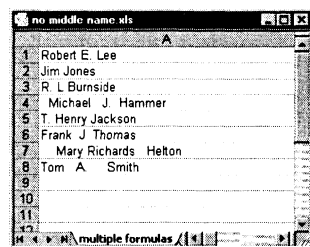


Рис. 20.2. Перед вами стоит задача удаления из записей вторых имен и инициалов

имена и инициалы. Вы не можете написать VBA-макрос, а на редактирование списка вручную уйдет не один час. Вся надежда на использование формул. Заметьте, что не во всех записях есть вторые имена или инициалы, что делает задание менее тривиальным. Хотя данная задача и не является очень сложной, ее решение обычно включает использование нескольких промежуточных формул.

На рис. 20.3 представлено более удобное решение, для получения которого потребовалось шесть промежуточных формул, описанных в табл. 20.2. Список сотрудников приведен в столбце А, а результат отображается в столбце Н. В столбцах с В по G приведены промежуточные формулы.



Следует отметить, что данный алгоритм не идеален. Например, он не будет работать, если есть только фамилия человека (например, Тарабров). Также он не сработает, если есть два вторых имени (например, Джон Яков Роберт Смит). Данный недостаток обусловлен тем, что формула просто ищет в записи второй пробел. В данном примере формула возвратит результат Джон Роберт Смит. Далее в данной главе будет описан метод формулы массива, используемый для определения последнего пробела в строке.

	A	B	C	D	E	F	G	H
1	Robert E. Lee	Robert E. Lee	7	10	10	Robert	Lee	Robert Lee
2	Jim Jones	Jim Jones	4	#ЗНАЧ!	4	Jim	Jones	Jim Jones
3	R. L. Burnside	R. L. Burnside	3	5	5	R	Burnside	R. Burnside
4	Michael J. Hammer	Michael J. Hammer	8	11	11	Michael	Hammer	Michael Hammer
5	T. Henry Jackson	T. Henry Jackson	3	9	9	T	Jackson	T. Jackson
6	Frank J. Thomas	Frank J. Thomas	6	8	8	Frank	Thomas	Frank Thomas
7	Mary Richards Helton	Mary Richards Helton	5	14	14	Mary	Helton	Mary Helton
8	Tom A. Smith	Tom A. Smith	4	7	7	Tom	Smith	Tom Smith
9								
10								
11								

Рис. 20.3. Для удаления фамилий и инициалов потребовалось шесть промежуточных формул

Таблица 20.2. Промежуточные формулы в первой строке рабочего листа, показанного на рис. 20.3

Ячейка	Промежуточная формула	Описание действия
B1	=СЖПРОБЕЛЫ (A1)	Удаляет лишние пробелы
C1	=НАЙТИ (" "; B1; 1)	Находит позицию первого пробела
D1	=НАЙТИ (" "; B1; C1+1)	Находит позицию второго пробела, если он есть
E1	=ЕСЛИ (ЕОШИБКА (D1) ; C1; D1)	Использует первый пробел, если второго не существует
F1	=ЛЕВСИМВ (B1; C1-1)	Извлекает имя
G1	=ПРАВСИМВ (B1; ДЛСТР (B1) -E1)	Извлекает фамилию
H1	=F1&" "&G1	Объединяет имя и фамилию

Приложив немного усилий, можно устранить все промежуточные формулы, заменить их все единственной мегаформулой. Такой результат достигается путем написания всех промежуточных формул и последующего редактирования конечной результирующей формулы (в данном случае, формулы в столбце Н) путем замены всех ссылок на ячейки,

копиями формул, содержащихся в этих ячейках. К счастью, вы имеете возможность использовать для копирования и вставки буфер обмена (см. предыдущую врезку "Копирование текста формулы"). Этот процесс должен продолжаться до тех пор, пока единственной оставшейся ссылкой будет ссылка на ячейку A1. По завершении процесса вы получите следующую мегаформулу, записанную в одной ячейке:

```
=ЛЕВСИМВ(СЖПРОБЕЛЫ(A1);НАЙТИ(" ";СЖПРОБЕЛЫ(A1);1)-1)&" "&ПРАВСИМВ(СЖПРОБЕЛЫ(A1);ДЛСТР(СЖПРОБЕЛЫ(A1))-ЕСЛИ(ЕОШИБКА(НАЙТИ(" ";СЖПРОБЕЛЫ(A1);НАЙТИ(" ";СЖПРОБЕЛЫ(A1);1)+1));НАЙТИ(" ";СЖПРОБЕЛЫ(A1);1);НАЙТИ(" ";СЖПРОБЕЛЫ(A1);НАЙТИ(" ";СЖПРОБЕЛЫ(A1);1)+1)))
```

Если формула работает правильно, столбцы с промежуточными формулами можно удалить, так как они больше не используются.

Пошаговое описание процедуры

Если вам все еще не до конца понятен данный процесс, ниже приведено пошаговое описание используемой процедуры:

1. Изучите формулу в ячейке H1. В ней есть две ссылки (на ячейки F1 и G1):

```
=F1&" "&G1
```

2. Активизируйте ячейку G1 и скопируйте формулу (без знака равенства) в буфер обмена.
3. Активизируйте ячейку H1 и замените ссылку на ячейку G1 содержимым буфера обмена. Теперь в ячейке H1 будет содержаться формула:

```
=F1&" "&ПРАВСИМВ(B1;ДЛСТР(B1)-E1)
```

4. Активизируйте ячейку F1 и скопируйте формулу (без знака равенства) в буфер обмена.
5. Активизируйте ячейку H1 и замените ссылку на ячейку F1 содержимым буфера обмена. Теперь в ячейке H1 будет содержаться формула:

```
=ЛЕВСИМВ(B1;C1-1)&" "&ПРАВСИМВ(B1;ДЛСТР(B1)-E1)
```

6. Теперь рассмотрите ссылки на три ячейки (B1, C1 и E1). Ссылки на них необходимо заменить формулами, содержащимися в данных ячейках.
7. Замените ссылку на ячейку E1 формулой, содержащейся в данной ячейке. Результат будет таким:

```
=ЛЕВСИМВ(B1;C1-1)&" "&ПРАВСИМВ(B1;ДЛСТР(B1)-ЕСЛИ(ЕОШИБКА(D1);C1;D1))
```

8. Заметьте, что формула в ячейке H1 теперь имеет две ссылки на ячейку D1. Скопируйте формулу из ячейки D1 и замените обе ссылки на нее. Теперь формула выглядит так:

```
=ЛЕВСИМВ(B1;C1-1)&" "&ПРАВСИМВ(B1;ДЛСТР(B1)-ЕСЛИ(ЕОШИБКА(НАЙТИ(" ";B1;C1+1));C1;НАЙТИ(" ";B1;C1+1)))
```

9. Замените четыре ссылки на ячейку C1 формулой, содержащейся в данной ячейке. Результат будет таким:

```
=ЛЕВСИМВ(B1;НАЙТИ(" ";B1;1)-1)&" "&ПРАВСИМВ(B1;ДЛСТР(B1)-ЕСЛИ(ЕОШИБКА(НАЙТИ(" ";B1;НАЙТИ(" ";B1;1)+1));НАЙТИ(" ";B1;1);НАЙТИ(" ";B1;НАЙТИ(" ";B1;1)+1)))
```

10. И наконец, замените девять ссылок на ячейку B1 формулой, содержащейся в данной ячейке. Теперь вы получите:

```
=ЛЕВСИМВ(СЖПРОБЕЛЫ(A1);НАЙТИ(" ";СЖПРОБЕЛЫ(A1);1)-  
1)&" "&ПРАВСИМВ(СЖПРОБЕЛЫ(A1);ДЛСТР(СЖПРОБЕЛЫ(A1))-  
ЕСЛИ(ЕОШИБКА(НАЙТИ(" ";СЖПРОБЕЛЫ(A1);НАЙТИ(" ";СЖПРОБЕЛЫ(A1);1)+1  
));НАЙТИ(" ";СЖПРОБЕЛЫ(A1);1);НАЙТИ(" ";СЖПРОБЕЛЫ(A1);НАЙТИ  
(" ";СЖПРОБЕЛЫ(A1);1)+1))
```

Обратите внимание, что теперь формула в ячейке H1 имеет ссылки только на ячейку A1. Создание мегаформулы завершено; она выполняет точно ту же работу, что и все промежуточные формулы (которые теперь можно удалить).

Сравнение скорости и эффективности

Из-за большой сложности мегаформулы может показаться, что использование такой методики приводит к замедлению вычислений. На самом деле ничего подобного не происходит. В качестве тестового примера я создал рабочую книгу, использующую мегаформулу 65536 раз. Затем была создана другая рабочая книга, в которой для получения 65536 результатов каждый раз использовалось шесть промежуточных формул. Результаты были сравнены со скоростью работы пользовательской VBA-функции, выполняющей те же операции. В табл. 20.3 представлены результаты, полученные для трех этих методов.

Таблица 20.3. Сравнение промежуточных формул, мегаформулы и VBA-функции

Метод	Время пересчета (секунды)	Размер файла
Промежуточные формулы	10,8	24,4МБ
Мегаформула	6,2	8,9МБ
VBA-функция	106,7	8,6МБ

Результат, конечно же, будет зависеть от производительности системы и количества установленной оперативной памяти.

Как не трудно заметить, использование мегаформулы приводит к ускорению вычислений и *существенному* уменьшению размера рабочей книги. VBA-функция оказалась намного медленнее — результаты фактически несопоставимы. Такой результат достаточно типичен для VBA-функций, они всегда медленнее встроенных функций Excel.

Использование мегаформулы для нахождения положения последнего пробела в строке

Как отмечалось, в приведенном выше примере есть слабое место: для определения второго имени формула выполняет поиск второго пробела. Более правильно было бы искать *последний* пробел. К сожалению, в Excel нет простого пути определения позиции первого с конца строки определенного символа. В примере данного подраздела приводится решение данной проблемы и описывается способ нахождения позиции первого с конца текстовой строки определенного символа.



В данной технике используются массивы; материал, относящийся к данной теме, приведен в части IV.

В данном примере описано, как создать мегаформулу, которая бы возвращала позицию последнего пробела в строке. Конечно же, данную формулу можно изменить для работы с любым другим символом.

Создание промежуточных формул

Замысел состоит в создании массива содержащихся в строке символов, расположенных в обратном порядке. После того как массив будет создан, мы сможем использовать функцию ПОИСКПОЗ для нахождения первого пробела в массиве.

Посмотрите на рис. 20.4, на котором показан результат работы промежуточных формул. Ячейка A1 содержит случайное имя, состоящее из 12 символов. В диапазоне B1 : B12 содержится следующая формула массива:

```
{=СТРОКА(ДВССЫЛ("1:" &ДЛСТР(A1)))}
```

	A	B	C	D	E	F	G
1	Jim E. Brown	1	12	n	6	7	
2		2	11	w			
3		3	10	o			
4		4	9	r			
5		5	8	B			
6		6	7				
7		7	6				
8		8	5	E			
9		9	4				
10		10	3	m			
11		11	2	i			
12		12	1	J			
13							

Рис. 20.4. Все эти промежуточные формулы в конце концов преобразованы в мегаформулу

Эта формула вводится во всем диапазоне B1 : B12 путем его выделения, ввода формулы и нажатия клавиш <Ctrl+Shift+Enter>. Фигурные скобки вводить не нужно. Excel добавляет их для обозначения формулы массива. Данная формула возвращает массив из двенадцати последовательных целых чисел.

В диапазоне C1 : C12 содержится следующая формула массива:

```
{=ДЛСТР(A1)+1-B1:B12}
```

Данная формула по существу располагает целые числа, полученные в столбце B, в обратной последовательности.

Диапазон D1 : D12 содержит следующую формулу массива:

```
{ПСТР(A1;C1:C12;1)}
```

Данная формула использует функцию ПСТР для извлечения отдельных символов из ячейки A1. Функция ПСТР использует в качестве второго аргумента массив C1 : C12. Результатом будет массив символов, расположенных в обратном порядке.

В ячейке E1 содержится формула:

```
=ПОИСКПОЗ(" ";D1:D12;0)
```

Данная формула, не являющаяся формулой массива, использует функцию ПОИСКПОЗ для возврата позиции первого пробела в диапазоне D1 : D12. В примере, приведенном на

рис. 20.4, формула возвращает значение 6. Это говорит о том, что первый пробел является шестым символом с конца текста в ячейке A1.

В ячейке F1 содержится формула:

=ДЛСТР (A1) +1-E1

Данная формула возвращает позицию последнего пробела в строке.

Вам интересно, как все эти формулы объединяются в одну строку? Читайте дальше и все узнаете.

Создание мегаформулы

В данный момент в ячейке F1 содержится искомый нами результат. Попробуем объединить все промежуточные формулы в одну. Цель, опять таки, состоит в том, чтобы создать формулу, содержащую ссылки лишь на ячейку A1. Вот последовательность действий, ведущих к намеченной цели:

1. Формула в ячейке F1 содержит ссылку на ячейку E1. Замените данную ссылку текстом формулы в ячейке E1. В результате данного действия формула в ячейке F1 приобретет вид:

=ДЛСТР (A1) +1-ПОИСКПОЗ (" ";D1:D12;0)

2. Теперь формула содержит ссылку на диапазон D1:D12. Данный диапазон содержит одну формулу массива. Замена ссылки формулой массива приводит к появлению в ячейке F1 следующей формулы массива:

{=ДЛСТР (A1) +1-ПОИСКПОЗ (" ";ПСТР (A1;C1:C12;1);0)}



Так как ссылка в ячейке F1 заменена формулой массива, теперь вы должны ввести формулу в ячейке F1 как формулу массива (после ввода формулы нажать клавиши <Ctrl+Shift+Enter>).

3. Теперь формула в ячейке F1 содержит ссылку на диапазон C1:C12, который также содержит формулу массива. Замените ссылку на диапазон C1:C12 содержащейся в нем формулой, перенеся данную формулу в ячейку F1:

{=ДЛСТР (A1) +1-ПОИСКПОЗ (" ";ПСТР (A1;ДЛСТР (A1) +1-B1:B12;1);0)}

4. Затем заменяем ссылку на диапазон B1:B12 содержащейся в нем формулой. Вот полученный результат:

{=ДЛСТР (A1) +1-ПОИСКПОЗ (" ";ПСТР (A1;ДЛСТР (A1) +1-СТРОКА (ДВССЫЛ ("1:"&ДЛСТР (A1))));1);0)}

Теперь формула в ячейке F1 ссылается лишь на ячейку A1; именно это мы и стремились получить. Мегаформула выполняет всю необходимую работу, так что все промежуточные формулы могут быть смело удалены.

Хотя в примере использовалась 12-значная запись и при создании формулы использовались диапазоны из 12 строк, конечная формула не имеет ни одной ссылки на эти диапазоны. Следовательно, данная мегаформула будет работать со значениями любой длины.

Запуск мегаформулы

На рис. 20.5 показан рабочий лист, содержащий в столбце A список сотрудников. Столбец B содержит мегаформулу, разработанную в предыдущем подразделе. Столбец C содержит формулу, извлекающую символы, следующие после последнего пробела, которые составляют фамилию, представленную в столбце A.

Для примера, в ячейке C1 содержится следующая формула:

=ПРАВСИМВ (A1 ; ДЛСТР (A1) - B1)

При желании вы можете удалить формулы, содержащиеся в столбце B, и создать специальную формулу, возвращающую фамилию. Для этого замените ссылку на ячейку B1 содержащейся в ней формулой. Результатом будет следующая формула массива:

{=ПРАВСИМВ (A1 ; ДЛСТР (A1) - (ДЛСТР (A1) +1 - ПОИСКПОЗ (" " ; ПСТР (A1 ; ДЛСТР (A1) +1 - СТРОКА (ДВССЫЛ ("1:" & ДЛСТР (A1)) ; 1) ; 0)))}

	A	B	C	D
1	Paula M. Smith	9	Smith	
2	Michael Alan Jones	13	Jones	
3	Mike Helton	5	Helton	
4	Tom Alvin Jacobs	10	Jacobs	
5	John Jacob Robert Smith	18	Smith	
6	Mr. Hank R. Franklin	12	Franklin	
7	James Jackson Jr.	14	Jr.	
8	Jill M. Horneq	8	Horneq	
9	Rodger K. Moore	10	Moore	
10	Andy R. Maxwell	8	Maxwell	
11	Michelle Theresa Hunt	17	Hunt	
12				

Рис. 20.5 Столбец B содержит мегаформулу, возвращающую позицию последнего пробела в записи, находящейся в столбце A



Текст формулы, скопированной из ячейки B1, заключается в круглые скобки. Если этого не сделать, формула не будет правильно производить вычисления.

Использование мегаформулы для определения действительности номера кредитной карточки

Может не всем это известно, но с помощью относительно сложного алгоритма анализа цифр номера кредитной карточки можно определить его действительность. В дополнение к этому, проанализировав первые цифры и длину номера, можно определить и тип кредитной карточки. В табл. 20.4 представлена информация, описывающая четыре основных вида кредитных карточек.



Проверка в данном случае означает определение правильности структуры серийного номера кредитной карточки. Конечно же, данная техника не позволяет определить, соответствует ли номер активному счету кредитной карточки.

Таблица 20.4. Информация о четырех видах кредитных карточек

Кредитная карточка	Начальные цифры	Количество цифр
Mastercard	51-55	16
Visa	4	13 или 16
American Express	34 или 37	15
Discover	6011	16

Проверить номер кредитной карточки можно путем обработки контрольной суммы составляющих его цифр. Все номера счетов, используемые большинством кредитных карточек используют “десятичный” алгоритм проверки знаков. Ниже приведены последовательные операции стандартной процедуры:

1. Добавьте нули впереди номера счета для получения 16-значного числа.
2. Начиная с первой цифры, удвойте через раз цифры номера счета. Если в результате получается двузначное число, сложите оба знака.
3. Сложите восемь значений, полученных на втором этапе, и сумму пропущенных цифр исходного номера.

4. Если полученная на третьем этапе сумма без остатка делится на 10, номер кредитной карточки действителен.

В примере данного подраздела описана мегаформула, определяющая действительность номера кредитной карточки.

Базовые формулы

На рис. 20.6 показан рабочий лист, созданный для анализа номера кредитной карточки и определения его действительности. Для вынесения вердикта данная рабочая книга использует совсем небольшое количество формул.

Digit Number	Digit	Digit Multiplier	Equals	Sum of the digits
1	0	2	0	0
2	0	1	0	0
3	0	2	0	0
4	4	1	4	4
5	3	2	6	6
6	8	1	8	8
7	4	2	8	8
8	8	1	8	8
9	4	2	8	8
10	2	1	2	2
11	2	2	4	4
12	0	1	0	0
13	1	2	2	2
14	0	1	0	0
15	6	2	12	3
16	5	1	5	5
17				
18				
19				
20				
21				58
22				

Рис. 20.6. Формулы данной рабочей книги определяют действительность номера кредитной карточки

В данной рабочей книге номер кредитной карточки вводится в ячейке F1, без пробелов и дефисов. Далее вводится формула в ячейке F2. Данная формула при необходимости добавляет нули в начале номера до получения 16-значного номера. Остальные формулы используют строку в ячейке F2.

=ПОВТОР("0";16-ДЛСТР(F1))&F1



При вводе номера кредитной карточки, содержащего более 15 знаков, следите, чтобы Excel не округлил его до 15 знаков. Вы можете поместить перед номером апостроф или предварительно применить к ячейке текстовый формат (на вкладке Число диалогового окна Формат ячеек).

Столбец A содержит ряд целых чисел от 1 до 16, представляющих положение цифр кредитной карточки.

Столбец B содержит формулы, извлекающие цифры из ячейки F2. Для примера, в ячейке B5 содержится следующая формула:

=ПСТР(\$F\$2;A5;1)

Столбец C содержит множители для каждого из знака кредитной карточки: 2 и 1 поочередно.

Столбец D содержит формулы, проводящие умножение чисел в столбце B на множители в столбце C. Например, формула в ячейке D5 имеет вид:

=B5*C5

Столбец E содержит формулы, суммирующие знаки, отображаемые в столбце D. Однозначные числа возвращаются без изменений. Если же число двузначно, в столбце E отображается сумма его знаков. Например, если в столбце D отображается число 12, формула в E возвратит значение 3 (то есть, 1+2). Следующая формула выполняет данное действие:

```
=ЦЕЛОЕ ((D5/10)+ОСТАТ((D5);10))
```

Ячейка E21 содержит простую формулу СУММ, складывающую значение в столбце E:
=СУММ(E5:E20)

Следующая формула в ячейке G1 определяет остаток деления ячейки E21 на 10. Если он равен 0, номер кредитной карточки действителен, и формула отображает сообщение VALID. В остальных случаях формула возвращает сообщение INVALID:

```
=ЕСЛИ(ОСТАТ(E21;10)=0;"VALID";"INVALID")
```

Преобразование в формулы массива

Вы должны понимать, что так как промежуточные формулы находятся в нескольких строках, мегаформула, которая будет создана, будет формулой массива.

Сначала необходимо преобразовать все формулы в формулы массива. Заметьте, что в столбцах A и C находятся значения, а не формулы. Для того чтобы значения использовать в мегаформуле, они должны генерироваться формулами, если точнее, формулами массива.

Введите в диапазоне A5:A20 приведенную ниже формулу массива. Данная формула возвращает ряд из 16 последовательных целых чисел.

```
{=СТРОКА(ДВССЫЛ("1:16"))}
```

Для столбца B: выделите диапазон B5:B20 и введите следующую формулу массива, извлекающую знаки из номера кредитной карточки:

```
{=ПСТР($F$2;A5:A20;1)}
```

Затем в столбце C необходимо ввести формулу массива, генерирующую чередующиеся значения, умножаемые на 2 и 1. Следующая формула вводится в диапазоне C5:C20:

```
{=ОСТАТ(СТРОКА(ДВССЫЛ("1:16")));2)+1}
```

Для столбца D: выделите диапазон D5:D20 и введите следующую формулу массива:

```
{=B5:B20*C5:C20}
```

И наконец, выделите диапазон E5:E20 и введите следующую формулу массива:

```
{=ЦЕЛОЕ((D5:D20/10)+ОСТАТ((D5:D20);10))}
```

Теперь мы имеем пять столбцов из шестнадцати строк, и всего лишь пять реальных формул. Данные формулы являются многоячеечными формулами массива.

Построение мегаформулы

Процесс создания мегаформулы для данной задачи начинается с ячейки G1, содержащей конечный результат. Исходная формула в данной ячейке имеет вид:

```
=ЕСЛИ(ОСТАТ(E21;10)=0;"VALID";"INVALID")
```

Сначала заменим ссылку на ячейку E21 содержащейся в ней формулой. Формула в ячейке G1 приобретет вид:

```
=ЕСЛИ(ОСТАТ(СУММ(E5:E20);10)=0;"VALID";"INVALID")
```

Затем заменим ссылку на диапазон E5:E20 содержащейся в нем формулой массива. Теперь формула становится формулой массива, поэтому для ее ввода необходимо ис-

пользовать сочетание клавиш <Ctrl+Shift+Enter>. Формула в ячейке G1 после данной подстановки принимает следующий вид:

```
{=ЕСЛИ(ОСТАТ(СУММ(ЦЕЛОЕ((D5:D20/10)+ОСТАТ((D5:D20);10)));10)=0;"VALID";"INVALID")}
```

Заменяем ссылку на диапазон D5:D20 содержащейся в нем формулой массива. Формула в ячейке G1 приобретет следующий вид:

```
{=ЕСЛИ(ОСТАТ(СУММ(ЦЕЛОЕ((B5:B20*C5:C20/10)+ОСТАТ((B5:B20*C5:C20);10)));10)=0;"VALID";"INVALID")}
```

Теперь заменим ссылку на диапазон C5:C20 содержащейся в нем формулой массива. Обратите внимание, текст скопированной формулы заключается в скобки. Вот полученный результат:

```
{=ЕСЛИ(ОСТАТ(СУММ(ЦЕЛОЕ((B5:B20*(ОСТАТ(СТРОКА(ДВССЫЛ("1:16"));2)+1)/10)+ОСТАТ((B5:B20*(ОСТАТ(СТРОКА(ДВССЫЛ("1:16"));2)+1));10)));10)=0;"VALID";"INVALID")}
```

Замена ссылки на диапазон B5:B20 содержащейся в нем формулой массива приводит к такому результату:

```
{=ЕСЛИ(ОСТАТ(СУММ(ЦЕЛОЕ((ПСТР($F$2;A5:A20;1)*(ОСТАТ(СТРОКА(ДВССЫЛ("1:16"));2)+1)/10)+ОСТАТ((ПСТР($F$2;A5:A20;1)*(ОСТАТ(СТРОКА(ДВССЫЛ("1:16"));2)+1));10)));10)=0;"VALID";"INVALID")}
```

Заменяем ссылку на диапазон A5:A20 содержащейся в нем формулой массива. Формула в ячейке G1 приобретет следующий вид:

```
{=ЕСЛИ(ОСТАТ(СУММ(ЦЕЛОЕ((ПСТР($F$2;СТРОКА(ДВССЫЛ("1:16"));1)*(ОСТАТ(СТРОКА(ДВССЫЛ("1:16"));2)+1)/10)+ОСТАТ((ПСТР($F$2;СТРОКА(ДВССЫЛ("1:16"));1)*(ОСТАТ(СТРОКА(ДВССЫЛ("1:16"));2)+1));10)));10)=0;"VALID";"INVALID")}
```

И наконец, заменим две ссылки на ячейку F2 формулой, содержащейся в данной ячейке. После выполнения подстановок мы получим следующую формулу:

```
{=ЕСЛИ(ОСТАТ(СУММ(ЦЕЛОЕ((ПСТР(ПОВТОР("0";16-ДЛСТР(A2))&A2;СТРОКА(ДВССЫЛ("1:16"));1)*(ОСТАТ(СТРОКА(ДВССЫЛ("1:16"));2)+1)/10)+ОСТАТ((ПСТР(ПОВТОР("0";16-ДЛСТР(A2))&A2;СТРОКА(ДВССЫЛ("1:16"));1)*(ОСТАТ(СТРОКА(ДВССЫЛ("1:16"));2)+1));10)));10)=0;"VALID";"INVALID")}
```

Теперь можно удалить ненужные больше промежуточные формулы. Результирующая мегаформула, имеющая длину в 275 символов, выполняет работу 51 промежуточной формулы!

Преимущества и недостатки использования мегаформул

Если вы ознакомились с представленными в данной главе примерами, то вероятно осознали, что основное преимущество мегаформул заключается в устранении промежуточных формул. Это позволяет рационализировать рабочий лист, уменьшить размер файла рабочей книги и даже привести к повышению скорости пересчета.

В чем же недостатки? Для создания мегаформулы, конечно же, необходимо затратить дополнительные усилия и время. И вы, конечно же, не могли не заметить, что разобраться в мегаформуле практически невозможно, даже человеку ее создавшему. Если вы решили использовать мегаформулу, прежде чем приступить к ее созданию, убедитесь, что промежуточные формулы работают правильно. Лучше даже где-нибудь хранить одну копию промежуточных формул на случай обнаружения ошибки или необходимости внесения изменений.

Резюме

В данной главе описана полезная методика, позволяющая скомбинировать несколько формул в одну сложную (мегаформулу). В ней также представлено несколько примеров создания таких формул.

В следующей главе рассмотрена возможность создания формул с целью отладки рабочих книг.

Инструменты и методы отладки формул

В этой главе...

- ◆ Отладка формул?
- ◆ Проблемы формул и их решение
- ◆ Средства проверки данных Excel
- ◆ Средства проверки данных сторонних разработчиков
- ◆ Резюме

Ошибки совершаются везде. При создании формул Excel ошибки встречаются весьма часто. В данной главе описаны стандартные ошибки формул, а также представлено рассмотрение средств и методов, призванных помочь в создании формул, корректно выполняющих ту работу, для которой они предназначались.

Отладка формул?

Термин *отладка* используется для обозначения процесса нахождения и исправления ошибок в компьютерных программах. Если говорить строго, формулы Excel не являются компьютерными программами. Вместе с тем, формулам присущи проблемы того же рода, что и компьютерным программам. Если созданная формула не работает так, как предполагалось, источник проблемы необходимо идентифицировать и устранить.

Конечная цель в разработке решения электронной таблицы состоит в получении правильного результата. В случае простых рабочих листов обычно не возникает проблем в определении правильности результата. Но по мере увеличения размера и сложности рабочих листов проверка достоверности становится более сложной задачей.

Поиск ошибок электронных таблиц

Использование электронных таблиц может оказаться рискованным и опасным. Можно легко принять за достоверный любой результат, выдаваемый электронной таблицей. Если такой результат используется для принятия важного решения, гарантия того, что результат является правильным, крайне необходима.

Исследований, связанных с ошибками в электронных таблицах, было проведено не очень много. Проведенные в этой области работы выявили, что от 20 до 40 процентов всех электронных таблиц содержат ошибки какого-то типа. Если вам интересны подобные исследования, посетите узел Spreadsheet Research, поддерживаемый Ray Panko (Рей Панко) из Гавайского университета. Вот его адрес:

<http://panko.cba.hawaii.edu/ssr/>

Внесение на рабочий лист изменений, даже весьма незначительных, может вызвать эффект домино, приводящий к возникновению ошибок в других ячейках. Скажем, можно

очень легко ввести значение в ячейке, содержащей формулу. Такая простая ошибка может существенно повлиять на другие формулы рабочего листа, а сама ошибка может быть обнаружена спустя большой промежуток времени после ее возникновения. Или же она может быть вообще *никогда* не обнаружена.

Проблемы формул и их решение

Ошибки формул разбиты на шесть общих категорий:

- ◆ **Синтаксические ошибки.** Проблема с синтаксисом формулы. Например, записаны лишние скобки или же сделана ошибка в написании имени функции.
- ◆ **Логические ошибки.** Формула не возвращает ошибку, однако она логически неправомерна и приводит к некорректному возвращаемому результату.
- ◆ **Неверные ссылки.** С логикой у формулы все в порядке, однако в формуле используется неверная ссылка на ячейку. Как простой пример, формула СУММ может включать не все данные, которые вы хотите просуммировать.
- ◆ **Циклические ссылки.** Циклическая ссылка встречается, когда формула прямо или косвенно ссылается на свою собственную ячейку. Циклические ссылки иногда создаются умышленно, но в большинстве случаев они приводят к возникновению проблемы.
- ◆ **Ошибка записи формулы массива.** При вводе (или редактировании) формулы массива необходимо использовать комбинацию клавиш <Ctrl+Shift+Enter>. Если этого не сделать, Excel не распознает такую формулу как формулу массива.
- ◆ **Ошибка незавершенных вычислений.** Формула просто не вычисляется до конца. Microsoft признала наличие определенных проблем в механизме вычислений некоторых версий Excel. Для гарантированного полного вычисления формул используйте клавиши <Ctrl+Alt+F9>.

Синтаксические ошибки обычно легче всего обнаружить и исправить. В большинстве случаев вы будете знать, в каком месте была совершена ошибка. К примеру, Excel не позволит ввести формулу с недостающими скобками. Другие синтаксические ошибки также обычно приводят к возвращению в ячейке сообщения об ошибке.

В оставшейся части данного раздела приводится описание некоторых общих проблем формул и также даны советы по их обнаружению и устранению.

Недостающие скобки

В формуле каждой открывающей скобке должна соответствовать закрывающая. Обычно, если в формуле отсутствует необходимая скобка, Excel не позволит ввести такую формулу. Исключением из данного правила являются простые формулы, использующие функцию. Например, если ввести следующую формулу (с недостающей закрывающей скобкой), Excel ее примет как представленное выражение и сам добавит недостающую скобку.

=СУММ(A1:500

Формула может содержать равное число открывающих и закрывающих скобок, но при этом скобки могут быть расставлены неправильно. Для примера рассмотрим следующую формулу, преобразующую текстовую строку так, что первый ее символ становится прописным, а все остальные символы строчными. Эта формула содержит пять пар скобок, расставленных должным образом.

=ПРОПИСН(ЛЕВСИМВ(A1)) & ПРАВСИМВ(СТРОЧН(A1)); ДЛСТР(A1)-1

Следующая формула также имеет пять пар скобок, однако расставлены они неверно. В результате получается правильная с точки зрения синтаксиса формула, которая просто возвращает неверный результат.

=ПРОПИСН (ЛЕВСИМВ (А1) &ПРАВСИМВ (СТРОЧН (А1) ; ДЛСТР (А1) -1))

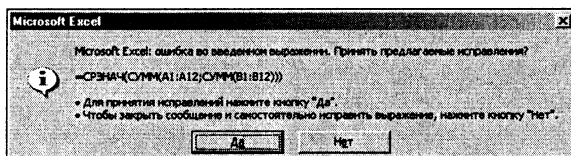
Часто неправильное расположение скобок приводит к возникновению синтаксической ошибки, сопровождаемой сообщением о том, что введено слишком много или слишком мало аргументов функции.



Excel может оказать вам помощь при работе с недостающими скобками. Если при редактировании формулы подвести указатель к скобке и немного подождать, Excel примерно на одну секунду выделит данную скобку и соответствующую ей “парную” полужирным начертанием.

Автоисправление формул

При вводе формулы, содержащей синтаксическую ошибку Excel пытается определить проблему и предлагает ее предполагаемое решение. Приведенный рисунок демонстрирует пример предлагаемого решения.



Будьте осторожны при принятии предлагаемых исправлений, так как они не всегда оказываются верными. Для примера введем следующую формулу (с недостающими скобками):

=СРЗНАЧ (СУММ (А1 : А12 ; СУММ (В1 : В12))

Excel предложит следующие исправления для формулы:

=СРЗНАЧ (СУММ (А1 : А12 ; СУММ (В1 : В12)))

У вас может возникнуть желание без размышлений принять такие изменения. В этом случае будет получена синтаксически корректная функция, однако это будет не та функция, которую я хотел написать:

=СРЗНАЧ (СУММ (А1 : А12) ; СУММ (В1 : В12))

Ячейки заполнены знаками

Ячейка может заполняться набором символов “решетки” (#) по одной из следующих причин:

- ◆ Ширины столбца недостаточно для вмещения форматированного численного значения. Для исправления такой ситуации можно либо расширить ячейку, либо использовать другой числовой формат.
- ◆ Ячейка содержит формулу, возвращающую некорректную дату или время. Excel, например, не поддерживает даты до 1900 года или отрицательные значения времени. Попытка отображения таких значений приводит к заполнению ячейки символами #. Увеличение ширины ячейки не исправляет данную ошибку.

Пустые ячейки не пустые

Может быть, некоторые из вас обнаружили, что при нажатии клавиши <пробел> содержимое ячейки уничтожается. На самом деле при этом вставляется невидимый символ пробела, что не эквивалентно уничтожению содержимого ячейки.

К примеру, следующая формула возвращает число непустых ячеек в диапазоне A1:A10. Если некоторые ячейки из данного диапазона “удалены” с помощью клавиши <пробел>, они будут включены в счет, а формула возвратит неправильный результат.

=СЧЁТЗ (A1:A10)

Если формула отказывается игнорировать пустые ячейки, убедитесь, действительно ли они пустые.

Формула возвращает ошибку

Формула может возвращать одно из значений ошибки:

- ◆ #ДЕЛ/0
- ◆ #Н/Д
- ◆ #ИМЯ?
- ◆ #ПУСТО!
- ◆ #ЧИСЛО!
- ◆ #ССЫЛКА!
- ◆ #ЗНАЧ!

В последующих подразделах рассмотрены возможные проблемы, вызывающие данные ошибки.



В Excel 2002 появилась новая возможность определения типа ошибки на печати. Для доступа к данному средству выберите команду Файл⇒Параметры страницы и щелкните на вкладке Лист; можно задать вывод ошибок ячеек в виде дефиса, пустого места или значения #Н/Д.

Поиск ошибок

Кнопка Источник ошибки на панели инструментов Зависимости может помочь в обнаружении ячейки, приводящей к возникновению ошибки. Довольно часто ошибка в одной ячейке возникает вследствие наличия ошибки в другой. Активизируйте содержащую ошибку ячейку и щелкните на кнопке Источник ошибки. Excel отобразит стрелки, указывающие на источник ошибки.

Ошибка #ДЕЛ/0!

Деление на ноль недопустимо. Если попытаться выполнить такую операцию, Excel отобразит хорошо знакомое значение ошибки #ДЕЛ/0!.

Так как Excel считает значение пустой ячейки равным нулю, данная ошибка будет получена и при делении на отсутствующее значение. Это стандартная ситуация при создании формул для еще не введенных значений, что и показано на рис. 21.1. В ячейке D2 находится следующая формула, которая скопирована вниз:

=(C2-B2)/C2

Эта формула находит разницу между значениями в столбцах В и С. Начиная с июня данные отсутствуют, поэтому формула возвращает ошибку #ДЕЛ/0!.

Для исключения вывода ошибки можно воспользоваться функцией ЕСЛИ, которая проверяет наличие пустых ячеек в столбце С:

=ЕСЛИ (C2=0; " "; (C2-B2) / C2)

Данная формула выводит пустую строку, если ячейка С2 пустая или содержит 0; в противном случае выводится вычисляемое значение.

Другое решение заключается в использовании функции ЕСЛИ для проверки существования любого условия ошибки. Следующая формула как раз отображает пустую строку в случае получения ошибки любого типа.

=ЕСЛИ (ЕОШИБКА ((C2-B2) / C2) ; " "; (C2-B2) / C2)

1	Month	Last Year	This Year	Change
2	January	175	188	6.9%
3	February	156	166	6.0%
4	March	198	175	-13.1%
5	April	144	187	23.0%
6	May	132	149	11.4%
7	June	198		#ДЕЛ/0!
8	July	202		#ДЕЛ/0!
9	August	184		#ДЕЛ/0!
10	September	140		#ДЕЛ/0!
11	October	198		#ДЕЛ/0!
12	November	232		#ДЕЛ/0!
13	December	255		#ДЕЛ/0!
14				

Рис. 21.1. Ошибка #ДЕЛ/0! возникает в случае отсутствия значений в столбце С2

Ошибка #Н/Д

Ошибка #Н/Д возникает, если в любой из ячеек, на которую ссылается формула, отображается значение #Н/Д.



Некоторые пользователи для явного указания отсутствующих данных любят использовать обозначения =ЗНАЧ() или #Н/Д. При этом становится совершенно ясно, что данные недоступны и не удалены случайно. Если создавать диаграмму типа графика для ячеек, содержащих значение #Н/Д, отсутствующие данные интерполируются. В случае же пустой ячейки отсутствующим данным соответствует разрыв кривой.

Ошибка #Н/Д возникает также тогда, когда функция поиска не может найти соответствие.

Ошибка #ИМЯ?

Ошибка #ИМЯ? возникает в следующих случаях:

- ♦ Формула содержит неопределенное имя ячейки или диапазона.
- ♦ Формула содержит текст, который Excel интерпретирует как неопределенное имя. Например, ошибка в написании имени функции приводит к отображению ошибки #ИМЯ?.
- ♦ Формула использует функцию рабочего листа, определенную в надстройках, а надстройка не установлена.



Excel присущи определенные проблемы с именованными диапазонами. Если удалить имя ячейки или диапазона, используемое в формуле, формула продолжает использовать это имя, хоть оно больше не определено. Как результат, формула отображает ошибку #ИМЯ?. Казалось бы, что Excel мог бы преобразовывать имена в ссылки на соответствующие ячейки, однако этого не происходит.

Ошибка #ПУСТО!

Данная ошибка возникает, когда формула пытается использовать пересечение двух диапазонов, которые в действительности не имеют общих ячеек. Оператором пересечения областей

является пробел между ссылками. Следующая формула, например, возвращает значение #ПУСТО!, так как два выбранных диапазона не пересекаются.

=СУММ(B5:B14 A16:F16)

Следующая формула уже не возвращает значение #ПУСТО!, вместо этого она отображает содержимое ячейки B9, являющейся пересечением двух диапазонов.

=СУММ(B5:B14 A9:F16)

Ошибка #ЧИСЛО!

Ошибка #ЧИСЛО! возникает в одном из следующих случаев:

- ◆ Для функции, использующей числовой аргумент, задан аргумент другого типа.
- ◆ Используемая итерации функция не может найти результат. К функциям, использующим итерации, относится, например СТАВКА.
- ◆ Формула возвращает слишком большое или слишком маленькое значение. Excel поддерживает величины в пределах от 1E-307 до 1E+307

Ошибка #ССЫЛКА!

Ошибка #ССЫЛКА! возникает в случае использования формулой ошибочной ссылки на ячейку. Данная ошибка может встретиться в следующих ситуациях:

- ◆ Ячейка, на которую ссылалась формула, была удалена. Например, следующая формула отображает ошибку #ССЫЛКА!, если удалена строка 1, столбец A или столбец B.

=A1/B1

- ◆ Формула скопирована в новое место, где относительные ссылки на ячейки становятся недействительными. Например, если скопировать следующую формулу из ячейки A2 в ячейку A1, формула возвратит ошибку #ССЫЛКА!, так как в ней будет присутствовать ссылка на несуществующую ячейку.

=A1-1

- ◆ Ячейка была вырезана (командой Правка⇒Вырезать) и затем вставлена в ячейку, на которую ссылается формула. Формула отобразит #ССЫЛКА!.

Обратите внимание на цвет

Когда вы редактируете содержимое ячейки, содержащей формулу, Excel использует цветовое выделение для обозначения ссылок на ячейки и диапазоны. При этом на рабочем листе контурами такого же цвета выделяются соответствующие ячейки и диапазоны. Таким образом, можно визуально просмотреть используемые формулой данные.

Цветные контуры можно изменять, модифицируя ссылки на ячейку или диапазон. Для изменения используемой ссылки перетащите границу контура или маркер выделения (в правом нижнем углу контура).

Ошибка #ЗНАЧ!

Ошибка #ЗНАЧ! весьма распространена и встречается в следующих ситуациях:

- ◆ В качестве аргумента функции использованы данные несоответствующего типа, или же формула пытается произвести операцию, используя некорректные данные. Например, формула складывающая значение с текстовой строкой возвратит значение #ЗНАЧ!.

- ◆ В качестве аргумента функции используется диапазон, тогда как аргументом должно быть единственное значение.
- ◆ Пользовательская функция не вычислена. В некоторых версиях Excel вставка или перемещение листов может приводить к возникновению данной ошибки. Для принудительного пересчета воспользуйтесь клавишами <Ctrl+Alt+F9>.
- ◆ Пользовательская функция пытается выполнить недопустимую операцию. Например, пользовательская функция не может изменить среду Excel или внести изменения в другие ячейки.
- ◆ При вводе формулы массива вы забыли нажать клавиши <Ctrl+Shift+Enter>.

Абсолютные и относительные ссылки

Как говорилось в главе 2, ссылки на ячейки могут быть относительными (например, A1), абсолютными (например, \$A\$1) или смешанными (например, A\$1 или \$A1). Тип ссылки становится значимым, только если формула копируется в другие ячейки.

Довольно часто ошибки возникают вследствие использования относительных ссылок вместо абсолютных. На рис. 21.2 ячейка C1 содержит налоговую ставку, используемую в формулах в столбце C. Формула в ячейке C4 имеет следующий вид:

=B4+(B4*\$C\$1)

Отметьте, что ссылка на ячейку C1 абсолютна. При копировании формулы в другие ячейки столбца C она продолжает ссылаться на ячейку C1. Если бы ссылка на ячейку C1 была относительной, формула возвращала бы неверный результат.

Item	Price	Price + Tax
C-092	\$149.95	\$160.82
R-112	\$79.99	\$85.79
G-972	\$39.95	\$42.05

Рис. 21.2. Формулы в ячейках C4:C6 используют абсолютную ссылку на ячейку C1

Приоритет операторов

В Excel установлены строгие правила относительно порядка выполнения математических операторов. В табл. 21.1 приведены операторы с указанием их приоритета (чем меньше число, тем раньше выполняется оператор). Из данной таблицы видно, например, что умножение имеет более высокий приоритет по сравнению со сложением. Поэтому умножение будет выполняться раньше.

Таблица 21.1. Приоритет операторов в формулах Excel

Символ	Оператор	Приоритет
-	Отрицание	1
%	Процент	2
^	Степень	3
* и /	Умножение и деление	4
+ и -	Сложение и вычитание	5
&	Объединение текста	6
=, <, > и <>	Сравнение	7

Если у вас возникают сомнения в вопросах подобного рода (или если вы просто хотите сделать выражение более понятным), используйте круглые скобки. Применение круглых скобок дает гарантию выполнения операций в верном порядке. Следующая формула, например, производит умножение ячейки A1 на ячейку A2, а затем к результату добавляет 1. Умножение выполняется раньше, так как данная операция имеет более высокий приоритет.

$$=1+A1*A2$$

Ниже приводится более наглядная запись данной формулы. Использование скобок не обязательно, но с ними порядок выполнения действий становится очевидным.

$$=1+(A1*A2)$$

Обратите внимание, что операторы отрицания и вычитания выглядят совершенно одинаково. Это, как вы можете догадаться, может приводить к возникновению путаницы. Рассмотрим две формулы:

$$=-3^2$$

$$=0-3^2$$

Первая формула, как и ожидается, возвращает значение 9. Вторая же формула возвращает значение -9. Возведение в квадрат всегда дает положительный результат, так как же Excel смог получить отрицательное значение?

В первой формуле знак минус является оператором отрицания и имеет более высокий приоритет. Во втором же случае знак минус — оператор вычитания, имеющий приоритет ниже, чем оператор возведения в степень. Таким образом, значение 3 возводится в квадрат, а затем результат вычитается из нуля, что и дает отрицательное значение.



Интерпретация оператора отрицания проводится в Excel довольно необычно. Другие электронные таблицы (например Lotus 1-2-3 или Quattro Pro) возвращают -9 в обеих формулах. Более того, при использовании языка VBA Excel также возвращает результат -9 для этих выражений.

Использование круглых скобок в приведенной ниже формуле заставляет Excel интерпретировать оператор как знак минус, а не как оператор отрицания. Данная формула возвращает значение 9.

$$=0+(-3^2)$$

Формулы не вычислены

Если в рабочем листе используются пользовательские функции, написанные на VBA, может случиться так, что у использующей их формулы возникают проблемы с пересчетом, и формула возвращает неверный результат. Для принудительного пересчета всех формул используйте сочетание клавиш <Ctrl+Alt+F9>.



В версиях до Excel 2000 данная комбинация клавиш не документирована.

Реальные и отображаемые значения

Вы можете столкнуться с ситуацией, когда кажется, что сложение значений выполняется с ошибкой. На рис. 21.3 показан рабочий лист, содержащий в каждой из ячеек диапазона B2 : B4 следующую формулу:

$$=1/3$$

Ячейка B5 содержит следующую формулу:

$$=СУММ(B2 : B4)$$

Все эти ячейки отформатированы для отображения чисел с тремя знаками после запятой. Как видно, складывается впечатление, что формула в ячейке B5 дает неправильный результат (казалось бы, результат должен равняться 0,999). Данная формула, конечно же, возвращает *правильный* результат, так как она использует *реальные* значения из диапазона B2 : B4, а не те, что отображаются на экране.

Установив на вкладке Вычисления диалогового окна Параметры (для открытия данного диалогового окна выберите команду Сервис⇒Параметры) флажок точность, как на экране, можно указать Excel использовать в вычислениях отображаемые значения.

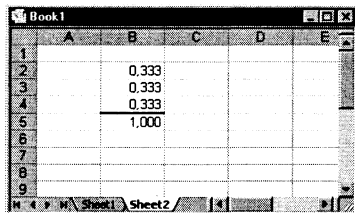


Рис. 21.3. Простая демонстрация “неправильного” сложения чисел



Действие флажка точность, как на экране распространяется не только на формулы, но и на непосредственно вводимые в ячейках значения. Например, если ячейка содержит значение 4,68 и отображается без десятичных знаков (то есть, как 5), установление данного флажка приведет к преобразованию 4,68 в 5,00. Данное изменение постоянно, и восстановление прежнего значения при сбросе флажка точность, как на экране невозможно.

Ошибки плавающей запятой

Компьютеры, по своей природе, не обладают бесконечной точностью. Excel хранит числа в двоичном представлении, используя для них байт, что соответствует точности до 15 знака. Некоторые числа нельзя точно выразить с использованием восьми байт, поэтому они аппроксимируются.

Для демонстрации того, как данное положение может привести к возникновению проблем, введем в ячейке A1 следующую формулу:

$$=(5,1-5,2)+1$$

Результатом данного выражения должно быть 0,9. Однако, если отформатировать данную ячейку для отображения 15 десятичных знаков, будет отображено значение 0,8999999999999999. Это происходит из-за того, что сначала выполняется действие в скобках, а промежуточный результат сохраняется в двоичном коде после аппроксимации. Затем формула добавляет к данному значению 1, а ошибка аппроксимации отображается в конечном результате.

В большинстве случаев данный тип ошибки не представляет никакой проблемы. Однако, если результат такой формулы проверяется логическим оператором, возникновение проблемы *не исключено*. К примеру, приведенная ниже формула (если в ячейке A1 находится предыдущая формула) возвращает значение ЛОЖЬ:

$$=A1=,9$$

Одним из возможных способов устранения данной ошибки является использование функции Excel ОКРУГЛ. Следующая формула, благодаря сравнению значения в ячейке A1, округленного до одного десятичного знака, возвращает значение ИСТИНА.

=ОКРУГЛ(A1;1)=0,9

Вот еще один пример проблемы “точности”. Попробуйте ввести следующую формулу:

=1,333+1,225-1,333-1,225

Если вы используете Excel 97 или более позднюю версию программы, формула возвратит значение 0. Предыдущие версии программы возвращали результат 2,22044604925031E-16 (число очень близкое к 0).



Начиная с Excel 97, если операция сложения или вычитания дает результат очень близкий к нулю, проводится компенсация ошибок, полученных в результате преобразования операнда в двоичный код и обратно. При выполнении предыдущего примера в Excel 97 (и в более поздних версиях), программа корректно находит результат.

Ошибка “фантомных ссылок”

После открытия рабочей книги вы можете увидеть сообщение, извещающее о существовании ссылок на другие источники данных. Данное сообщение возникает, когда рабочая книга содержит несвязанные формулы

В подавляющем большинстве случаев, проблема связанных ссылок возникает из-за неправильности имен. Выберите команду Вставка⇒Имя⇒Присвоить и просмотрите список имен. Если вы встретите имя со ссылкой на #ССЫЛКА!, удалите его.



Такие фантомные ссылки могут образовываться при копировании рабочего листа, содержащего имена. Более детальную информацию об именах можно почерпнуть из главы 3.

Ошибка циклических ссылок

Циклическая ссылка — это формула, содержащая ссылку на ячейку, содержащую формулу. Ссылка может быть прямой или косвенной. О том, как можно выявить циклические ссылки, речь пойдет в следующем разделе данной главы “Средства проверки данных Excel”.



Как говорилось в главе 14, возможны ситуации, когда циклические ссылки создаются преднамеренно.

Средства проверки данных Excel

В Excel есть масса инструментов, призванных оказывать помощь в нахождении ошибок формул. В данном разделе описаны встроенные в Excel средства проверки данных.

Нахождение ячейки определенного типа

Диалоговое окно Выделение группы ячеек позволяет задать тип ячеек, которые должен выделить Excel. Чтобы отобразить это диалоговое окно, выберите команду Правка⇒Перейти (либо

воспользуйтесь клавишами <F5> или <Ctrl+G>. При этом открывается диалоговое окно Переход. Щелкните в нем на кнопке В42ыделить, что и приведет к открытию диалогового окна Выделение группы ячеек, показанного на рис. 21.4.



Если при выборе команды Правка⇒Перейти выбран диапазон из нескольких ячеек, команда будет применена только к выбранным ячейками. Если же выбрана отдельная ячейка, команда применяется ко всему рабочему листу.

Диалоговое окно Выделение группы ячеек используется для выделения ячеек определенного типа, что часто облегчает поиск ошибок. Например, если установить переключатель формулы, Excel выделит все ячейки, которые содержат формулы. Если уменьшить масштаб просмотра рабочего листа, можно хорошо понять его структуру (см. рис. 21.5).

Выделение ячеек с формулами может также использоваться для выявления еще одной распространенной ошибки — случайной замены формулы значением. Если вы увидите невыделенную ячейку в группе ячеек с формулами, весьма вероятно, что такая ячейка содержала формулу, которая в последствии заменена значением.

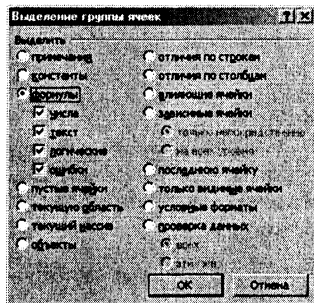


Рис. 21.4. Диалоговое окно Выделение группы ячеек

Просмотр формул

Для того чтобы разобраться в незнакомого рабочей книге, полезно просмотреть сами формулы, а не результаты их вычислений. Для этого выберите команду Сервис⇒Параметры и на вкладке Вид установите флажок формулы. Можно также перед использованием данной команды создать для данной рабочей книги новое окно просмотра. Это позволяет одновременно просматривать в одном окне сами формулы, а в другом результаты их вычислений. Для открытия нового окна выберите команду Окно⇒Новое.

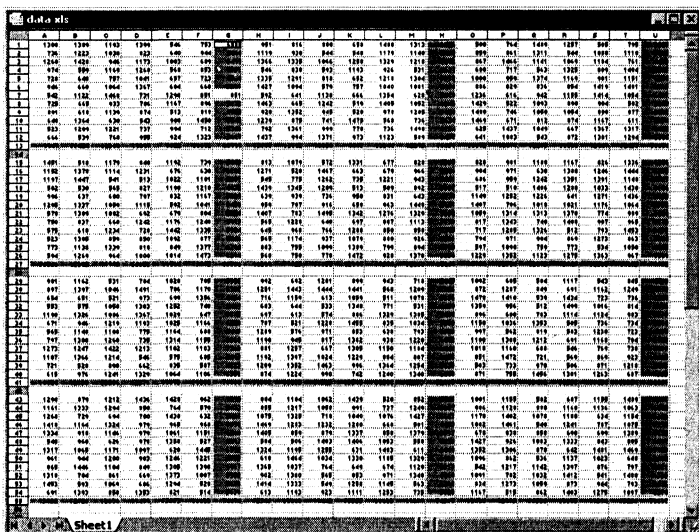


Рис. 21.5. Уменьшение масштаба и выделение всех формул может дать хорошее представление о структуре рабочего листа



Для переключения между режимом нормального просмотра и режимом просмотра формул используется комбинация клавиш <Ctrl+>.



В Excel 2002 для переключения режима просмотра формул используется команда Сервис⇒Зависимости формул⇒Режим проверки формул. При включении данного режима отображается панель инструментов Зависимости.

На рис. 21.7 представлен пример отображения рабочей книги в двух окнах. В верхнем окне используется режим обычного просмотра (результаты вычисления формул), а в нижнем окне отображаются сами формулы.

При включенном режиме просмотра формул Excel выделяет цветом ячейки, используемые формулой в активной ячейке. Например, на рис. 21.6 активизирована ячейка C11. Ячейки, используемые формулой данной ячейки, выделены в обоих окнах.

A	B	C	D	E	F	G	H	I	J
1	Commission F	5.50%	Normal commission rate						
2	Sales Goal	15%	Improvement from prior month						
3	Bonus Rate	6.50%	Paid if Sales Goal is attained						
4									
5	Sales Rep	Last Month	This Month	Change	Pct. Change	Met Goal?	Commission		
6	Murray	101,233	108,444	7,211	7.1%	TRUE	7,048.86		
7	Knuckles	120,933	108,434	(12,499)	-10.3%	FALSE	5,963.87		
8	Lefty	139,832	165,901	26,069	18.6%	TRUE	10,783.57		
9	Lucky	98,323	100,083	1,760	1.8%	FALSE	5,504.57		
10	Scarface	78,322	79,923	1,601	2.0%	FALSE	4,395.77		
11	Total	536,643	562,765	24,142	4.5%		33,696.63		

A	B	C	D	E
1	Commission Rate	0.055	Normal commission r:	
2	Sales Goal	0.15	Improvement from pric	
3	Bonus Rate	0.065	Paid if Sales Goal is :	
4				
5	Sales Rep	Last Month	This Month	Pct. Change
6	Murray	101233	108444	=C6-B6
7	Knuckles	120933	108434	=C7-B7
8	Lefty	139832	165901	=C8-B8
9	Lucky	98323	100083	=C9-B9
10	Scarface	78322	79923	=C10-B10
11	Total	=SUM(B6:B10)	=SUM(C6:C10)	=SUM(D6:D10)

Рис. 21.6. Отображение формул (нижнее окно) и результатов их вычисления (верхнее окно)

Отслеживание связей ячейки

Для того чтобы понять, как можно отследить взаимосвязи ячеек, необходимо ознакомиться с двумя концепциями:

- ♦ **Влияющие ячейки.** Применимо лишь к ячейкам, содержащим формулу; к влияющим ячейкам относятся все ячейки, вносящие вклад в результат формулы. Термин *прямое влияние* относится к ячейкам, используемым непосредственно в формуле. Если сами ячейки в формуле не используются, но их значения используются другими ячейками, на которые непосредственно ссылается формула, говорят о *непрямом влиянии*.

- ◆ **Зависимые ячейки.** Это ячейки в формуле, зависящие от отдельной ячейки. Зависимые ячейки составляют все формулы, которые ее используют. Опять таки, зависимость может быть прямой или непрямой.

Определение влияющих на формулу ячеек часто способно пролить свет на то, почему данная формула не работает правильно. Знание того, какие ячейки формул зависят от определенной ячейки, также полезно. Например, если вы собираетесь удалить формулу, информация о наличии зависимостей будет совсем не лишней.

Нахождение влияющих ячеек

Определить ячейки, используемые формулой в активной ячейке, можно множеством способов.

- ◆ Нажмите клавишу <F2>. Ячейки, используемые непосредственно в формуле, будут обведены цветными контурами, цвет контура соответствует ссылке на ячейку в формуле.
- ◆ Выберите команду **Правка**⇒**Перейти** (или нажмите клавишу <F5>) для открытия диалогового окна **Переход**. Затем щелкните на кнопке **Выделить** для открытия диалогового окна **Выделение группы ячеек**. Затем выберите переключатель **Влияющие ячейки** и установите либо флажок **только непосредственно** (для нахождения ячеек влияющих непосредственно), либо флажок **на всех уровнях** (для нахождения ячеек, влияющих как напрямую, так и косвенно). Щелкните на кнопке **ОК**, и Excel выделит все влияющие на формулу ячейки. С помощью данной техники можно находить влияющие ячейки только в том рабочем листе, в котором находится формула.
- ◆ Используйте комбинацию клавиш <Ctrl+> для выделения на активном рабочем листе всех влияющих непосредственно ячеек.
- ◆ Используйте комбинацию клавиш <Ctrl+Shift+> для выделения на активном рабочем листе всех влияющих ячеек (напрямую и косвенно).
- ◆ С помощью команды **Сервис**⇒**Зависимости формул**⇒**Панель зависимостей** отобразите панель инструментов **Зависимости**. Щелкните на кнопке **Влияющие ячейки** для отображения стрелок, указывающих на влияющие ячейки. Для просмотра нескольких уровней влияния щелкните на данной кнопке несколько раз. На рис. 21.7 представлен рабочий лист со стрелками, настроенными на отображение ячеек, влияющих на формулу в ячейке C13.

Нахождение зависимых ячеек

Найти формулы, зависящие от отдельной ячейки, можно множеством способов.

- ◆ Выберите команду **Правка**⇒**Перейти** (или нажмите клавишу <F5>) для открытия диалогового окна **Переход**. Затем щелкните на кнопке **Выделить** для открытия диалогового окна **Выделение группы ячеек**. Затем выберите переключатель **Зависимые ячейки** и установите либо флажок **только непосредственно** (для нахождения ячеек, зависящих напрямую), либо **на всех уровнях** (для нахождения ячеек, зависящих как напрямую, так и косвенно). Щелкните на кнопке **ОК**, и Excel выделит все зависимые от активной ячейки. С помощью данного способа можно находить зависимые ячейки только в том рабочем листе, в котором находится формула.
- ◆ Используйте комбинацию клавиш <Ctrl+>] для выделения на активном рабочем листе всех зависимых напрямую ячеек.
- ◆ Используйте комбинацию клавиш <Ctrl+Shift+>] для выделения на активном рабочем листе всех зависимых ячеек (напрямую и не косвенно).

- ◆ С помощью команды **Сервис**⇒**Зависимости** **формул**⇒**Панель зависимостей** отобразите панель инструментов **Зависимости**. Щелкните на кнопке **Зависимые** ячейки для отображения стрелок, указывающих на зависимые ячейки. Для просмотра нескольких уровней зависимости щелкните на данной кнопке несколько раз.

Sales Rep	Last Month	This Month	Sales	Margin	Met Goal	Commission
Murray	101,233	106,444	7,241	7.1%	TRUE	7,048.86
Knuckles	120,933	106,434	(12,499)	10.3%	FALSE	5,963.87
Lefty	139,832	165,901	26,069	18.6%	TRUE	10,783.57
Lucky	98,323	100,083	1,760	1.8%	FALSE	5,504.57
Scarface	78,322	79,923	1,601	2.0%	FALSE	4,395.77
Total	538,643	562,785	24,142	4.5%	FALSE	83,696.63

Average Commission Rate: 4.99%

Рис. 21.7. На данном рабочем листе добавлены стрелки для обозначения ячеек, влияющих на формулу в ячейке C13

Отслеживание ошибочных значений

Кнопка Проверка наличия ошибок панели инструментов Зависимости призвана оказывать помощь в нахождении ячеек, приводящих к возникновению ошибочных значений. Ошибка в одной ячейке часто является результатом наличия ошибки во влияющей ячейке. Для обозначения источника ошибки Excel создает цветные стрелки.

Исправление ошибок циклической ссылки

При случайном вводе формулы, содержащей циклическую ссылку, Excel выводит сообщение предупреждения. При щелчке на кнопке ОК Excel отображает панель инструментов Циклические ссылки (рис. 21.8). Если вам не удастся установить источник проблемы, воспользуйтесь инструментом Найти циклическую ссылку (раскрывающийся список), доступным на указанной панели, для выделения ячеек, вовлеченных в циклическую ссылку. Начните с выбора первой ячейки из списка и продолжайте двигаться по нему вниз до тех пор, пока не будет выяснена причина возникновения проблемы.

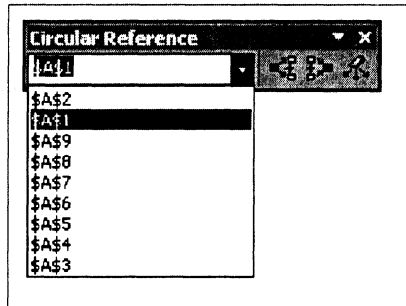


Рис. 21.8. Панель инструментов Циклические ссылки

Использование средства фоновой проверки ошибок Excel 2002

Если вы используете Excel 2002, то можете найти полезным новое средство автоматической проверки ошибок.



Диалоговое окно Параметры в Excel 2002 имеет несколько новых вкладок, одной из которых является вкладка Проверка ошибок (рис. 21.9). Включение данного средства осуществляется установкой флажка Включить фоновую проверку ошибок. В дополнение, раздел Проверять наличие следующих ошибок содержит ряд флажков, позволяющих указать, ошибки какого типа должны проверяться.

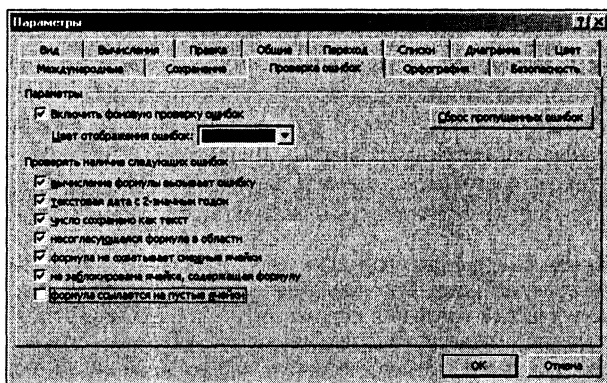


Рис. 21.9. Excel 2002 способен выполнять проверку формул на предмет потенциально возможных ошибок

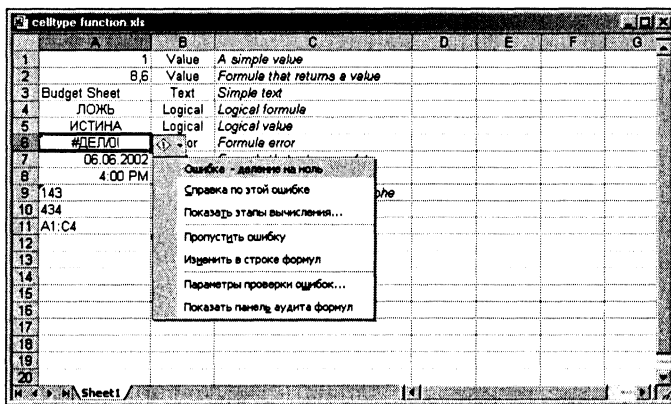


Рис. 21.10. При щелчке на смарт-тэге вам предоставляется список возможных действий

Когда средство проверки ошибок включено, Excel постоянно следит за рабочим листом, включая формулы в нем. При нахождении потенциальной ошибки Excel помещает в левом верхнем углу такой ячейки маленький треугольник. Если ячейка активизирована, появляется смарт-тэг. При щелчке на смарт-тэге пользователю предлагается несколько вариантов дальнейшего развития событий. На рис. 21.10 показаны варианты действий, доступные в случае

щелчка на смарт-тэге в ячейке, содержащей ошибку #ДЕЛ/0!. Доступные действия изменяются в зависимости от типа ошибки.

Во многих случаях пользователи выбирают команду Пропустить ошибку. Выбор данной команды исключает ячейку из списка ошибок. Вы можете снова отобразить все пропущенные ошибки; для этого необходимо щелкнуть на кнопке Сброс пропущенных ошибок, находящейся на вкладке Проверка ошибок диалогового окна Параметры.

Вы можете воспользоваться командой Сервис⇒Проверка наличия ошибок для последовательного отображения ячеек с потенциально возможными ошибками, подобно тому, как это делает программа проверки орфографии. Диалоговое окно Проверка ошибок показано на рис. 21.11. Обратите внимание, что это диалоговое окно немодальное, то есть когда оно открыто, вы по-прежнему имеете доступ к содержимому рабочего листа.

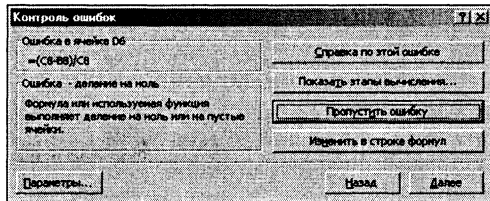


Рис. 21.11. Использование диалогового окна контроля ошибок для последовательного просмотра найденных Excel потенциальных ошибок



Важно понять, что средство проверки ошибок несовершенно. Практически, ему весьма далеко до совершенства. Другими словами, нельзя считать, что рабочий лист полностью лишен ошибок только на основе того, что Excel не выявил источников потенциальной опасности! Кроме того, имейте в виду, что данное средство не сможет отследить такую часто встречающуюся ошибку, как замена ячейки с формулой значением.

Проверка формул в Excel 2002

В Excel 2002 имеется новое средство, позволяющее просмотреть различные части вложенной формулы, представленные в порядке вычисления формулы.



Информация данного подраздела применима только к Excel 2002.

Для использования эксперта формул выделите содержащую формулу ячейку, а затем выберите команду Сервис⇒Зависимости формул⇒Вычислить формулу или же щелкните на кнопке Вычислить формулу панели инструментов Зависимости. Любое из этих действий вызывает диалоговое окно Вычисление формулы, показанное на рис. 21.12.

Щелкните на кнопке Вычислить для отображения результата вычисления выражения формулы. При каждом щелчке на кнопке производится следующее вычисление. Данное средство может оказаться весьма полезным в некоторых случаях, но обычно оно оставляет желать лучшего.

Средства проверки данных сторонних разработчиков

Существуют средства проверки данных, разработанные для Excel сторонними разработчиками ПО. Описанию данных программных продуктов (Power Utility Pak, Spreadsheet Detective и Excel Auditor) и посвящены последующие подразделы.

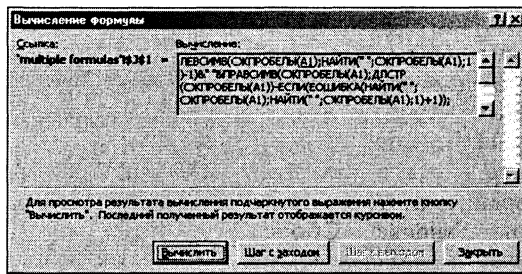


Рис. 21.12. Вычисление формул в Excel 2002 демонстрирует промежуточные результаты на всех этапах

Power Utility Pak

Power Utility Pak включает ряд полезных утилит, выполняющих проверку данных рабочих листов. Ниже приведен их список:

- ◆ **Workbook Summary Report (Общий отчет рабочей книги).** Выводит удобный общий отчет для всей рабочей книги.
- ◆ **Workbook Link Report (Отчет о связях рабочей книги).** Выводит отчет, описывающий все связи рабочей книги.
- ◆ **Worksheet Map (Карта рабочего листа).** Создает удобную карту, упрощающую нахождение ячеек различных типов (рис. 21.13).

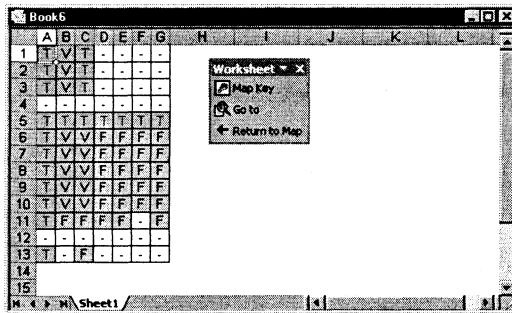


Рис. 21.13. Карта рабочего листа, созданная с помощью Power Utility Pak

- ◆ **Compare Sheets or Ranges (Сравнение листов или диапазонов).** Выполняет ячейочное сравнение двух рабочих листов или диапазонов.
- ◆ **Date Report (Отчет о датах).** Создает полезный отчет по всем ячейкам, связанным с датами.
- ◆ **Name Lister (Составитель списка имен).** Составляет список всех имен рабочей книги. В отличие от диалогового окна Excel Присвоение имени данная утилита включает в список также все скрытые имена.

- ◆ **Formula Report (Отчет о формулах).** Генерирует полезный (и подготовленный к печати) список всех формул, содержащихся на рабочем листе. Данная утилита более полезна, чем режим просмотра формул Excel.
- ◆ **VBA Project Summary Report (Итоговый VBA-отчет).** Генерирует отчет, описывающий VBA-процедуры, используемые на рабочей книге.

Spreadsheet Detective

Программа Spreadsheet Detective, разработанная Southern Cross Software, — это всеобъемлющее средство проверки данных в Excel. Для получения полной информации (и оценочной версии программы) посетите следующий узел:

<http://www.uq.net.au/detective/>

Excel Auditor

Программа Excel Auditor, разработана Byg Software, находящейся в Великобритании. Данный продукт включает множество инструментов, призванных помочь в поиске и исправлении ошибок электронных таблиц. Для получения полной информации посетите следующий узел:

<http://www.BygSoftware.com>

Резюме

В данной главе рассмотрены различные типы встречающихся ошибок формул. В ней описаны методы выявления ошибок и приведены некоторые советы по их устранению. Были также описаны встроенные средства проверки данных Excel, а также три программных продукта сторонних производителей.

Следующая глава является первой из четырех глав, посвященных созданию пользовательских функций рабочего листа с использованием VBA.

Часть VI

Разработка пользовательских функций

ГЛАВА 22

“Введение в VBA”

ГЛАВА 23

“Основы процедур функций”

ГЛАВА 24

“Принципы программирования на VBA”

ГЛАВА 25

“Примеры пользовательских
функций VBA”

Глава 22

Введение в VBA

В этой главе ...

- ◆ Несколько слов о VBA
- ◆ Введение в редактор Visual Basic
- ◆ Резюме

Эта глава ознакомит вас со средой VBA. Visual Basic for Applications представляет собой язык программирования Excel и используется для создания пользовательских функций. Перед тем как вы сможете создавать такие функции, используя VBA, вы должны иметь как базисные понятия о языке программирования VBA, так и некоторые знания о редакторе Visual Basic.

Несколько слов о VBA

Приложение Excel 5 было одной из первых программ, появившихся на рынке и поддерживающих VBA. Язык VBA является лучшим вариантом языка подготовки сценариев, созданным компанией Microsoft. Сейчас он включается во все прикладные программы пакета Office 2002 года, его также можно встретить и у сторонних производителей. В Excel существует два основных способа использования VBA:

- ◆ для автоматизации выполнения задач;
- ◆ для создания пользовательских функций, которые можно использовать при построении формул.



В Excel также существует другой способ создания функций, используя язык создания макросов XLM. Он является старым средством, но все же совместимым с разными версиями Excel. Данная книга полностью игнорирует язык XLM и всецело фокусирует свое внимание на языке VBA.

VBA является сложной темой для обсуждения — далеко более сложной, чем можно описать в этой книге. Поскольку в данной книге речь идет о формулах, я обращаю внимание только на один из самых важных (и не менее полезный) аспект VBA — создание пользовательских функций. Эти функции можно использовать (иногда их называют функциями, создаваемыми конечным пользователем) в формулах.

Введение в редактор Visual Basic

Перед тем как начать работать над созданием функций, следует ознакомиться с редактором Visual Basic. Данный редактор позволяет работать с модулями VBA, которые являются контейнерами VBA-кода.

В Excel 5 и в Excel 95 VBA-модуль представлен в виде отдельного листа рабочей книги. Начиная с Excel 97, VBA-модули больше не отображаются в таком виде. Точнее говоря, вы

используете редактор VBA для просмотра VBA-модулей и для работы с ними. В Excel 97 и в более поздних версиях VBA-модули представлены в виде упакованных в рабочей книге файлов; они невидимы до тех пор, пока не активизировать редактор Visual Basic.



В данной главе предполагается, что вы пользуетесь Excel 97 или более поздней версией данного продукта. Предыдущие версии не имели отдельного редактора.

Запуск редактора Visual Basic

Работа в Excel, можно перейти к редактору VBA, используя один из ниже предложенных способов:

- ◆ нажмите <Alt+F11>;
- ◆ выберите Сервис⇒Макрос⇒Редактор Visual Basic;
- ◆ щелкните на кнопке Редактор Visual Basic, расположенной на панели инструментов Visual Basic (она скрыта по умолчанию).

На рис 22.1 изображен общий вид редактора Visual Basic. Возможно, окно редактора, изображенное на рисунке, не совсем похоже на окно Visual Basic, отображаемое на экране вашего монитора. Данное окно настраиваемое. Окна можно скрывать, менять их размер, присоединять друг к другу, переорганизовывать и т. д.

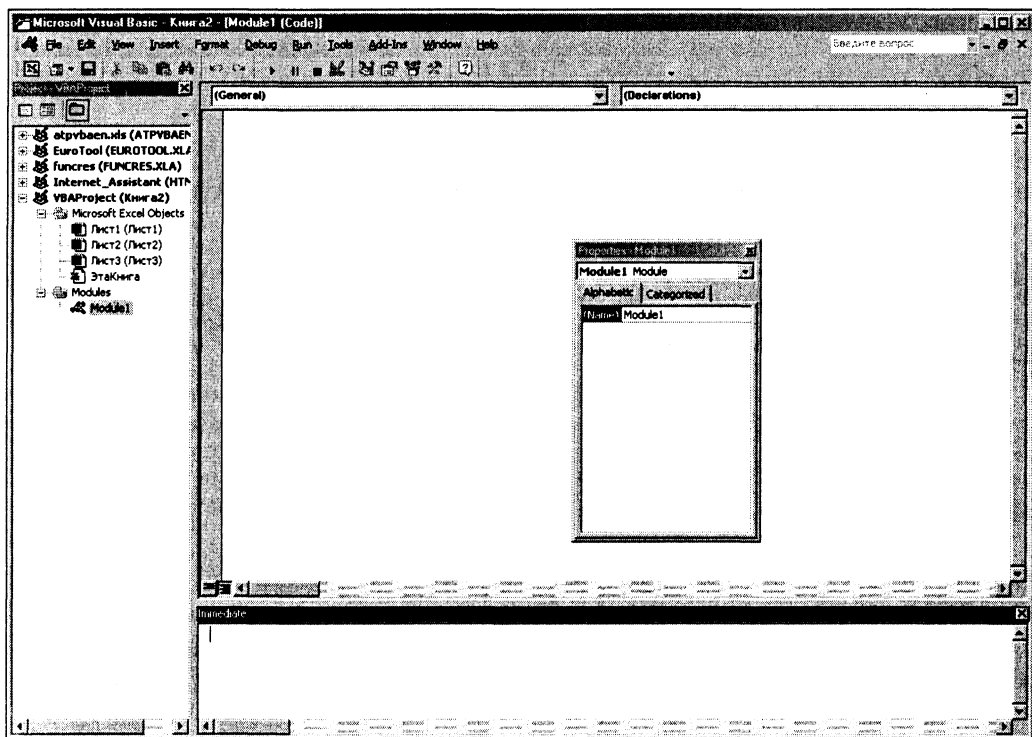


Рис. 22.1. Окно редактора Visual Basic

Компоненты редактора Visual Basic

Редактор Visual Basic состоит из ряда компонент. В этом разделе коротко описаны некоторые ключевые компоненты.

Строка меню

Строка меню редактора Visual Basic работает так же, как и любая другая строка меню. В ее состав входят команды, используемые для работы с компонентами редактора Visual Basic. Характерной чертой редактора Visual Basic является наличие контекстных меню. В сущности, щелчок правой кнопкой мыши на пустом месте окна редактора Visual Basic, инициирует вызов контекстного меню доступа к стандартным командам.

Панель инструментов

Стандартная панель инструментов по умолчанию находится прямо под строкой меню, доступной является только одна из шести панелей редактора Visual Basic. Панели в редакторе Visual Basic управляются так же, как и панели в Excel: их можно перенастраивать, перемещать, а также отображать другие панели.

Окно проекта

Окно проекта отображает древовидную схему, включающую каждую рабочую книгу, открытую в текущий момент в Excel (включая надстройки и скрытые рабочие книги). В редакторе Visual Basic каждая рабочая книга называется *проектом*. Я рассмотрю окно проекта детальнее в следующем разделе. Если окно проекта не отображено на экране, нажмите сочетание клавиш <Ctrl+R>.

Окно кода

Окно кода содержит код VBA. Каждый элемент проекта имеет соответствующее ему окно кода. Для того чтобы просмотреть окно кода для объекта, необходимо дважды щелкнуть кнопкой мыши на окне проекта. Или выделить элемент и щелкнуть на кнопке отображения кода, находящейся в верхней части окна проекта.

Например, для того чтобы просмотреть окно кода для отдельного объекта Лист1 рабочей книги, необходимо дважды щелкнуть на нем в окне проекта. Пока вы не добавили в файл VBA-код, окно кода будет пустым. Детально я рассмотрю окно кода далее в этой главе.

Окно свойств

Окно свойств включает в себя перечень всех допустимых свойств, характерных для выбранного объекта. Можно использовать это окно для изучения и изменения свойств. Можно использовать клавишу быстрого доступа <F4> для отображения окна свойств.

Окно быстрого доступа

Окно быстрого доступа является наиболее удобным для непосредственного выполнения VBA-операторов, отчетов тестирования и отладки кода. Это окно может либо отображаться, либо нет. Если окно быстрого доступа не отображено на экране, нажмите <Ctrl+G>. Для закрытия этого окна щелкните на кнопке **Закрыть**, находящейся в верхней части заголовка окна.

Использование окна проекта

Когда вы работаете в редакторе Visual Basic, каждая рабочая книга и надстройки, открытые в данный момент, считаются проектом. Проектом можно считать совокупность объектов, упорядоченных в виде структуры. Есть возможность раскрывать проект, щелкая на знаке “плюс” слева от имени проекта в окне проекта. Или же можно скрывать содержимое объекта, щелкая на знаке “минус” слева от имени проекта. На рис. 22.2 отображено окно проекта, содержащее список трех проектов (одна надстройка и две рабочие книги).

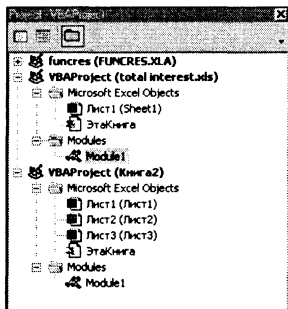


Рис. 22.2. Окно проекта, состоящее из трех проектов

Если вы попытаетесь ввести изменения в проект, защищенный паролем, вам будет предложено ввести пароль.

Каждый проект увеличивается в размере для отображения, по крайней мере одного объекта, называемого Microsoft Excel Object. Этот объект увеличивается в размере, включая элементы каждого рабочего листа и диаграммы на рабочей книге (каждый лист считается объектом), а также другой объект, называемый ЭтаКнига (который представляет объект рабочей книги). Если в проекте есть VBA-модули, листинг проекта также отображает

объект Modules, состоящий из модулей, представленных в нем. Проект также включает в себя объект, называемый Forms (который содержит объекты UserForm), и объект, называемый Class Modules (состоящий из объектов модулей классов). Данная книга обращает внимание исключительно на стандартные VBA-модули и не рассматривает объекты, включенные в объекты Microsoft Excel, объекты пользовательских форм или объекты модулей классов.



Если вы пользуетесь Excel 2002, проект может состоять из другого объекта, называемого References. Этот объект состоит из перечня ссылок, которые используются проектом. Ссылки можно удалить или добавить с помощью команды Tools⇒References. В отличие от других объектов, представленных в окне проекта, объекты ссылок не имеют модулей кода, связанных с ними.

Переименование проекта

По умолчанию все проекты называются VBAProject. В окне проекта имя рабочей книги появляется после имени проекта. Например, проект может быть представлен как:

VBAProject (budget.xls)

Можно изменить имя проекта более наглядным и описательным. Для этого:

1. Выберите проект в окне проектов.
2. Убедитесь, что отображено окно свойств (нажмите <F4>, если оно скрыто).
3. Измените имя с VBAProject на другое.

После внесения изменений, в окне проекта будет отображено новое имя.

Добавление нового модуля VBA

В новой рабочей книге Excel нет никаких VBA-модулей. Для добавления таких модулей в проект, необходимо выбрать имя проекта в окне проекта и выбрать Insert⇒Module.

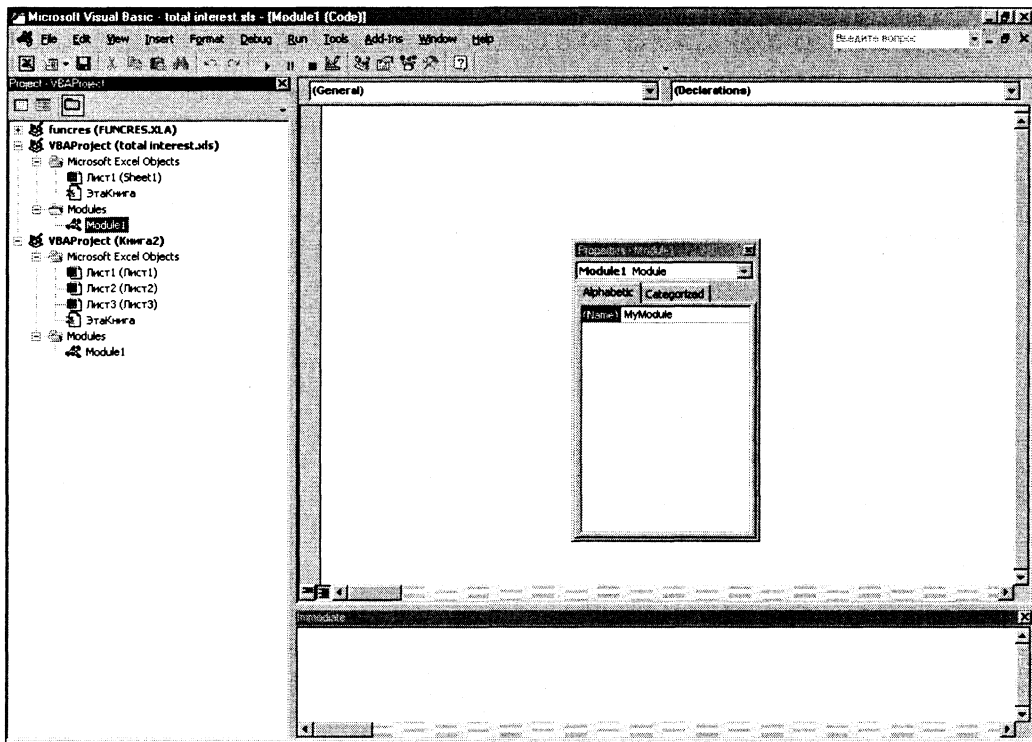


Рис. 22.3. Для изменения имени VBA-модуля используйте окно свойств



При создании функций нужно следить, чтобы они *находились* в стандартном VBA-модуле, а не в окне кода для объекта листа или рабочей книги. Если окно кода функции не будет находиться в VBA-модуле, она не будет работать!

Переименование модуля

VBA-модули имеют имена, назначенные по умолчанию, такие как Module1, Module2 и т. д. Для того чтобы переименовать VBA-модуль, необходимо выбрать его в окне проекта, а затем изменить имя, используя окно свойств (VBA-модуль имеет только одно свойство — Name). Если окно свойств не отображено на экране, нажмите <F4>. На рис. 22.3 представлен VBA-модуль, переименованный в MyModule.

Удаление VBA-модуля

Если вы хотите удалить VBA-модуль из проекта, выделите в окне проекта его имя и выберите в команду File⇒Remove xxx, (где xxx — имя модуля). Вам будет задан вопрос, хотите ли вы экспортировать модуль перед удалением. При экспортировании создается резервный файл с содержимым текущего модуля. Модули можно импортировать и экспортировать в любой проект.

Использование окна кода

За исключением объектов ссылки, каждый объект в проекте имеет связанное с ним окно кода. Обычно к этим объектам относят:

- ◆ саму рабочую книгу (объект называется ЭтаКнига в окне проекта);
- ◆ рабочий лист или рабочую диаграмму (например, Лист1 или Диаграмма1 в окне проекта);
- ◆ VBA-модуль;
- ◆ UserForm;
- ◆ модуль класса (особый тип модуля, который позволяет создавать новые классы модулей);
- ◆ ссылку (только в Excel 2000).



В данной книге внимание обращается исключительно на VBA-модули, в которых хранятся пользовательские функции рабочих листов.

Сворачивание и разворачивание окон

В любое время в редакторе Visual Basic может быть открыто много окон кода. На рис. 22.4 представлен пример.

Окна кода очень похожи на окна рабочих листов Excel. Их можно сворачивать, разворачивать, скрывать, переорганизовывать и т. д. Многие считают, что легче развернуть окно кода, в котором ведется работа. Однако у вас может возникнуть потребность отобразить одновременно два и больше открытых окон с кодом. Например, для сравнения кодов двух модулей или копировании кода одного модуля в другой.

Сворачивание окна кода изменяет его область. Можно щелкнуть на кнопке **Заккрыть**, находящуюся в верхней части строки заголовка для того, чтобы полностью закрыть окно. Для повторного его открытия необходимо дважды щелкнуть на соответствующем объекте в окне проекта.

Нельзя закрыть рабочую книгу непосредственно из редактора Visual Basic. Следует перейти в окно Excel и закрыть его обычным способом.

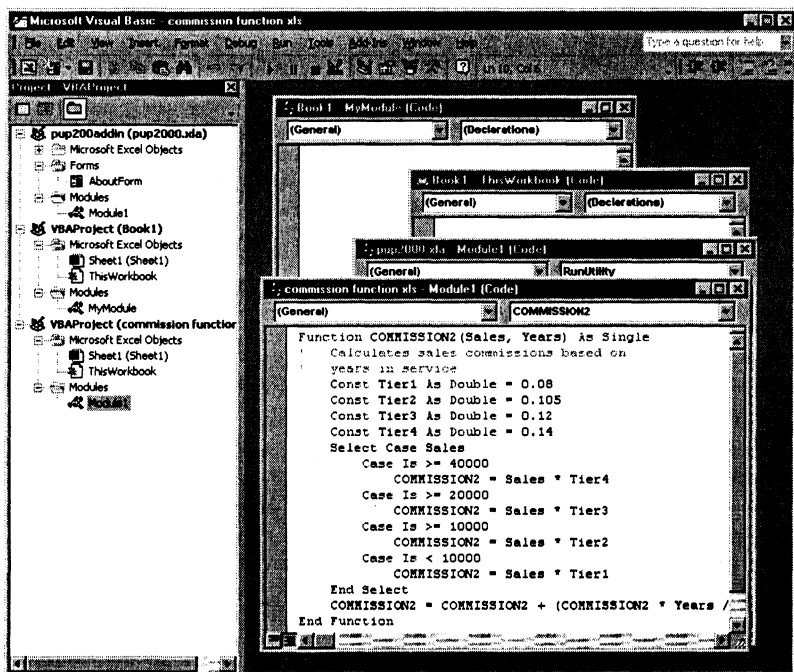


Рис. 22.4. Загроможденное окно введения кода

Сохранение кода

Обычно модуль может содержать три типа кода:

- ◆ **Процедура.** Процедура представляет собой набор инструкций, выполняющих определенные действия. Например, процедура, объединяющая несколько частей рабочей книги в один краткий отчет.
- ◆ **Функция.** Функция представляет собой набор инструкций, которые возвращают одну переменную или массив. Функции можно использовать в формулах.
- ◆ **Объявление.** Объявление — это предоставление информации о переменных, которые используются в VBA-коде. Например, можно объявлять тип переменных, которые будут использоваться в будущем.

В модуле VBA может храниться любое количество процедур и объявлений.



Данная книга обращает внимание исключительно на функции, поскольку они являются единственным типом кода, который можно использовать для построения формул.

Ввод кода

В данном разделе описаны различные способы введения VBA-кода в окне кодов. Для функций окно кода всегда относится к VBA-модулю. Добавить код в VBA-модуль можно тремя способами:

- ◆ набрав его вручную с клавиатуры;
- ◆ используя средство записи макросов (можно записать макрос и впоследствии конвертировать его в VBA-код);
- ◆ скопировать код из другого модуля и вставить его в тот модуль, с которым вы работаете.

Введение кода вручную

Иногда самый простой путь решения проблемы является самым лучшим. Можно получить код VBA, введя его вручную с клавиатуры. Ввод теста и его редактирование в VBA-модуле осуществляется обычным способом. Можно выделить текст и скопировать его или вырезать и вставить в другом месте.

Пользуйтесь клавишей <Tab> для того, чтобы делать отступы в строках, которые логически связаны одна с другой, — например, в блоках условных операторов `If EndIf`. Наличие абзаца является необязательным, но оно упрощает чтение кода.

Длина простого оператора в VBA зависит только от пользователя. Однако для повышения удобочитаемости можно преобразовать один длинный оператор в несколько коротких. Для этого заканчивайте строку пробелом и символом подчеркивания, затем нажмите клавишу ввода и продолжайте вводить код оператора в текущей строке. Таким образом, код будет представлен в виде оператора, расположенного в трех строках:

```
If IsNumeric(MyCell) Then_
    Result = "Number" Else_
    Result = "Non-Number"
```

Обратите внимание, что отступ добавлен в две последние строки оператора. Делать это необязательно, но помогает понять показать, что эти три строки представляют единый оператор.

После ввода операторов редактор Visual Basic для улучшения представления кода выполняет следующие действия:

- ◆ Вставляет пробелы между операторами. Например, при вводе `Ans=1+2` (без пробелов), VBA преобразует это в следующий вид:

```
Ans = 1 + 2
```

- ◆ Приводит в порядок регистр букв ключевых слов, свойств и методов. Если ввести такой текст:

```
user=application.username
```

редактор Visual Basic преобразует его в:

```
user = Application.UserName
```

- ◆ Поскольку вид имен переменных не имеет значения, редактор Visual Basic упорядочивает все переменные, имена которых начинаются с одной и той же буквы таким образом, что регистр букв введенных вами переменных не имеет никакого значения. Например, если вы сначала определите переменную как `myvalue` (все строчные) и потом введете переменную `MyValue` (в смешанном регистре), VBA поменяет все последующие экземпляры этой переменной на `MyValue`. Исключением этого может быть случай, когда вы определяете переменную с помощью оператора `Dim` или функционально подобным ему. Тогда переменная представляется в том виде, в каком и была объявлена.
- ◆ Проверяет операторы на наличие синтаксических ошибок. Если редактор Visual Basic найдет такого рода ошибки, он изменяет цвет строки и выводит предупреждающее сообщение.

ние. Можно настроить различные параметры поведения редактора Visual Basic в диалоговом окне Options (это можно сделать, выбрав команду меню Tools⇒Options).



Как и Excel, редактор Visual Basic имеет многоуровневые команды отмены действий и повторного выполнения команд. Поэтому, если возникла ситуация, при которой случайно удален важный оператор, можно, щелкнув несколько раз на кнопке Undo (или нажав сочетание клавиш <Ctrl+Z>), вернуть потерю. После выполнения команды Undo, можно выбрать команду Edit⇒ReDo Delete (или щелкнуть на кнопке Redo) для выполнения недавно отмененных изменений.

Использование средства записи макросов

Другим способом добавления кода в VBA-модуль является запись его в виде макроса Excel. Как бы вы ни пытались, записать процедуру *функции* (тип процедуры, используемой в формулах рабочего листа) не представляется возможным. Все записанные макросы являются внутренними процедурами. Использование средств записи макроса поможет определить различные свойства, которые можно использовать при построении функций. Например, для записи выполняемых действий включите утилиту записи макроса во время изменения имени пользователя. Для этого выполните следующие операции:

1. Выберите Сервис⇒Макрос⇒Начать запись.
2. В диалоговом окне записи макроса оставьте без изменений стандартные настройки и щелкните на кнопке ОК начала для записи. Excel отобразит небольшую панель Остановить запись.
3. Выберите Сервис⇒Параметры.
4. В диалоговом окне параметров перейдите на вкладку Общие.
5. Внесите изменения (любые) в поле имени пользователя.
6. Щелкните на кнопке ОК для закрытия диалогового окна опций.
7. Щелкните на кнопке Остановить запись на соответствующей панели.
8. Нажмите <Alt+F11> для запуска редактора Visual Basic.
9. В окне проекта выберите проект, соответствующий рабочей книге.
10. Дважды щелкните на VBA-модуле, содержащем записанный код. Обычно, это будет модуль с наибольшим номером (например, Module3).

Далее вы увидите, что VBA-процедура будет иметь подобный вид:

```
Sub Macro1()  
,  
'Macro1 Macro  
'Macro recorded 6/1-2001 by Bob Smith  
,  
  
    With Application  
        UserName = "Robert Smith"  
        StandartFont = "Arial"  
        StandartFontSize = "10"  
        DefaultFilePath = "d:\xlfiles"  
        EnavleSound = False  
        RollZoom = False  
    End With  
End Sub
```


Следует отметить, что здесь представлена внутренняя процедура, а не процедура функции. Другими словами, нельзя использовать эту процедуру при работе с формулами. Однако, если проанализировать вид кода, то можно увидеть ссылку на объект имени пользователя. Эту информацию можно использовать при создании функции. Например, следующая функция использует объект имени пользователя. Во время выполнения в формуле эта функция возвращает имя пользователя.

```
Function User()  
    USER = Application.UserName  
End Function
```

Для определения типов различных объектов можно проконсультироваться со справочной системой Excel, хотя использование средств записи макроса дает возможность решить проблему эффективнее, если наверняка не известно, что нужно искать. После того как предмет поиска наконец определен, можно обратиться к справочной системе за подсказкой.

Копирование VBA-кода

В этой главе вами уже рассмотрен процесс непосредственного ввода кода, а также запись действий для создания VBA-кода. Последний способ получения кода в VBA-модуле — его копирование из другого модуля. Например, если есть записанная функция для одного проекта, ее потом впоследствии можно использовать в текущем проекте. Вместо того чтобы вводить заново код, следует открыть рабочую книгу, сделать активным модуль и, используя обычные операции управления содержимым буфера обмена, скопировать код в текущий VBA-модуль.

Также можно копировать VBA-код из других источников. Например, можно найти листинг на Web-странице или в группе новостей. В таком случае нужно выделить в браузере текст, скопировать его в буфер и затем вставить в модуль.

Сохранение проекта

Как и в любой другой прикладной программе, в редакторе Visual Basic нужно периодически сохранять созданные данные. Для этого необходимо выбрать меню команду File⇒Save, нажать <Ctrl+S> или щелкнуть на кнопке сохранения, расположенную на стандартной панели.



Сохраняя проект, вы фактически сохраняете рабочую книгу Excel. Аналогично, если вы сохраняете рабочую книгу Excel, автоматически сохраняются изменения, внесенные в рабочие книги проекта Visual Basic.

В редакторе Visual Basic нет команды File⇒Save As. Для сохранения проекта под другим именем необходимо запустить Excel и использовать команду Excel — Файл⇒Сохранить как.

Резюме

В данной главе изложено введение в VBA, который представляет собой язык создания функций рабочих листов. В ней описаны различные компоненты редактора Visual Basic и рассмотрены методы создания кода в VBA-модуле.

В следующей главе будут рассмотрены процедуры функций VBA.

Основы процедур функций

В этой главе...

- ◆ Простой пример VBA-функции
- ◆ Работа процедуры функции
- ◆ Использование диалогового окна вставки функции
- ◆ Тестирование и отладка функций
- ◆ Резюме

В предыдущих главах данной книги рассматривались функции рабочих листов и способы построения сложных формул с их помощью. Эти функции, так же как и другие, доступные в надстройке “Пакет анализа”, обеспечивают большую гибкость при создании формул. Однако можно столкнуться с ситуациями, когда тяжело обойтись без пользовательских функций. В этой главе описаны случаи, в которых лучше применять пользовательские функции, созданные с помощью VBA. В ней также приведены методы тестирования и отладки пользовательских функций.

Для чего нужны пользовательские функции?

Наверняка, вы знакомы с функциями рабочих листов Excel — даже новичкам известно, как пользоваться большинством функций, такими как СУММ, СРЗНАЧ и ЕСЛИ. В Excel внедрено более 300 предопределенных функций, более того, после установки надстройки “Пакет анализа” становятся доступными дополнительные функции.

Можно использовать средства редактора Visual Basic для создания дополнительных функций, которые также известны как *пользовательские* функции. При существовании многих функций, доступных в Excel и VBA, можно задать вопрос, зачем же создавать новые функции. Ответ следующий: для упрощения работы и для предоставления формулам большей гибкости.

Например, можно создать функцию, значительно сокращающую стандартные формулы. Сокращенные формулы читабельнее и с ними намного проще работать. Однако важно понять, что пользовательские функции в формулах работают гораздо медленнее, чем встроенные. А в производительных системах скоростное различие часто протекает незамеченным.

Процесс создания пользовательской функции несложен. На самом деле многие наслаждаются созданием таких функций. Данная книга поможет вам в создании ваших собственных функций. В этой и в последующих главах вы увидите много разнообразных примеров функций, которые можно применить в своих целях.

Простой пример VBA-функции

Без лишней суеты я приведу вам простой пример функции VBA. Эта функция, названная USER, не предполагает использование каких-либо аргументов. Используя пользовательскую функцию, формула отображает имя пользователя с помощью символов верхнего регистра. Для создания этой функции необходимо:

1. Начать работу с новой книгой (на самом деле это необязательно, но сейчас проще придерживаться этого этапа)
2. Нажать <Alt+F11> для запуска редактора Visual Basic
3. Щелкнуть кнопкой мыши на имени рабочей книги в окне проекта. Если окно проекта скрыто, нажать <Ctrl+R> для его отображения
4. Выбрать Insert⇒Module для добавления VBA-модуля в проект
5. Ввести следующий код в окне кода:

```
Function USER ()
    ' Возвращает имя пользователя
    USER = Application.UserName
    USER = UCase (USER)
End Function
```



Я приводил это предупреждение в предыдущей главе, но его стоит повторить: при создании пользовательской функции нужно убедиться, что она находится в обычном VBA-модуле, а не в модуле кода листа или книги.

Опробуйте пользовательскую функцию. Перейдите в Excel (нажав <Alt+F11>) и введите следующую формулу в любую ячейку рабочей книги.

```
=USER ()
```

Задачи пользовательских функций

При совершенствовании пользовательских функций вам следует осознавать их назначение. Функция, используемая в формуле рабочего листа, должна быть *пассивной*. Другими словами, она не должна менять данные в рабочем листе.

Можно попробовать написать пользовательскую функцию, изменяющую формат ячейки. Например, может оказаться полезным иметь функцию, которая изменяет цвет текста в ячейке в зависимости от значения ячейки. Как бы вы не пытались, но такую функцию невозможно записать — все пробуют это сделать, но ни у кого не получается. Чтобы вы не делали, функция всегда будет возвращать ошибку из-за попыток внести изменения в рабочий лист. Нужно помнить, что функция может возвращать только переменную. Она не выполняет действия с объектами.

При правильном вводе VBA-кода, функция будет выполнена и имя пользователя отобразится (символами верхнего регистра) в ячейке.



Если формула возвращает значение ошибки, убедитесь, что VBA-код пользовательской функции находится в VBA-модуле (модуль листа или книги не является объектом). Также необходимо проверить, находится ли модуль в проекте той книги, в которую вы пытаетесь ввести формулу.

Когда Excel вычисляет значения рабочего листа, он рассчитает и пользовательскую функцию. Каждый оператор в функции анализируется и выполняется, а результат возвращается на рабочий лист. Функцию можно использовать неограниченное количество раз и применять ее любого значения ячеек.

Обратите внимание, что пользовательская функция работает так же, как и любая другая внедренная функция. Ее можно вставлять в формулу с помощью команды меню Вставка⇒Функция или используя кнопку вставки функции в строке формул. В диалоговом окне вставки функции пользовательские функции представляются как функции, созданные пользователем. Как и другие функции, их можно использовать в более сложных формулах. Например:

```
= "Hello" & USER ()
```

Следующую функцию можно использовать для подсчета количества символов в имени:
`=ДЛСТР (USER ())`

Если вам не нравится то, что имя записано символами верхнего регистра, исправьте процедуру следующим образом:

```
Function USER()  
    Возвращает имя пользователя  
    USER = Application.UserName  
End Function
```

После внесения изменений в функцию, возобновите работу Excel и нажмите <F9> для пересчета. В каждой ячейке, в которой используется функция пользователя, будет отображен отличный результат.

Работа процедуры функции

Процедура функции имеет структуру. Ниже рассмотрены некоторые технические особенности применения пользовательской функции. Они представляют собой общие инструкции по объявлению функций, присваиванию им имен, использованию функций в формулах и заданию аргументов.

Объявления функции

Для объявления функции используется общепринятый синтаксис:

```
[Public | Private][Static] Function name([arglist])  
    [As type]  
    [statements]  
    [name = expression]  
    [Exit Function]  
    [statements]  
    [name = expression]  
End Function
```

- ◆ *Public* (общий) показывает, что данная функция является доступной для других процедур в других модулях рабочей книги (необязательно).
- ◆ *Private* показывает, что данная функция доступна только для других процедур в одном и том же модуле (необязательно). При использовании ключевого слова *Private* функция не будет отображаться в диалоговом окне вставки функции.
- ◆ *Static* показывает, что значения переменных, объявленные в функции, являются зарезервированными для вызовов (необязательно).
- ◆ *Function* является ключевым словом, отображающим начало функции (обязательно).
- ◆ *Name* может быть любым допустимым для переменной именем. Когда функция заканчивает работу, ее результатом является значение, присваиваемое имени функции (обязательно).
- ◆ *Arglist* представляет собой список нескольких переменных, представляющих аргументы функции. Аргументы заключаются в круглые скобки. Нужно использовать точку с запятой для разделения этих аргументов (наличие аргументов не обязательно).
- ◆ *Type* показывает тип данных, возвращаемых функцией (необязательно).
- ◆ *Statements* являются полноценными VBA-операторами (необязательно).

- ◆ *Exit Function* — это оператор, результат которого — выход из функции (необязательно).
- ◆ *End Function* — ключевое выражение, показывающее конец работы функции (обязательно).

Выбор имени функции

Каждая функция должна иметь уникальное имя, и при выборе имени функций нужно придерживаться следующих правил:

- ◆ Можно использовать символы алфавита, цифры и некоторые символы пунктуации, но первый символ обязательно должен быть буквой.
- ◆ Можно использовать любую комбинацию букв верхнего и нижнего регистров.
- ◆ Нельзя использовать такие имена, которые похожи на адрес ячеек рабочей книги (такие как J21). Формально можно использовать такое имя для функции, но Excel не воспримет его как функцию.
- ◆ VBA не различает регистры. Для придания имени функции лучшей читабельности нужно использовать символы разных регистров (лучше написать InterestRate, чем interestrate).
- ◆ Нельзя использовать пробелы или точки. Для придания именам функции лучшей читабельности можно использовать символ нижнего подчеркивания (Interst_Rate).
- ◆ В имя функции нельзя вставлять следующие символы: #, \$, %, !. Они являются символами объявления типа, которые имеют специальное назначение в VBA.
- ◆ Имя функции может состоять не более чем из 255 символов — правда, никто не создает такие длинные имена!

Использование функций в формулах

Использование пользовательской VBA-функции в формулах аналогично использованию встроенных функций. Вы должны убедиться в том, что Excel может определить место расположения функции. Если функция находится в той же рабочей книге, что и формула, не нужно выполнять дополнительных действий. Если же они находятся в разных рабочих книгах, следует объяснить Excel, где ее найти. Это можно сделать следующим образом:

- ◆ **Начните имя функции из ссылки на файл.** Например, если вы хотите использовать функцию CountNames (подсчет имен), объявленной в рабочей книге как Myfuncs.xls, вам следует использовать следующую формулу:

```
=Myfuncs.xls!CountNames(A1:A1000)
```

Если вставить функцию с помощью диалогового окна вставки функции, ссылка на рабочую книгу будет вставлена автоматически.

- ◆ **Настройка ссылки на рабочую книгу.** Это можно сделать с помощью средств редактора Visual Basic Tools⇒References (см. рис. 23.1). Если функция объявлена в рабочей книге ссылки, имя рабочего листа использовать нет необходимости. Даже когда зависимая рабочая книга задана как ссылка, диалоговое окно вставки функции продолжит вставлять ссылку на рабочую книгу (даже если это не необходимо).



По умолчанию все проекты получают имя `VBAProject` — такое имя появляется в списке доступных ссылок в диалоговом окне `References`. Чтобы убедиться, что выбран правильный проект в диалоговом окне ссылок, проследите за диалоговым окном, в котором показано имя рабочей книги для выбранного элемента. Еще лучше изменить имя проекта на описательное. Для этого нужно выбрать проект, нажать `<F4>` для отображения окна свойств и потом указать имя объекта, отличающееся от `VBAProject`.

- ◆ **Создание надстройки.** При создании надстройки в рабочей книге, в которой есть пользовательские функции, не нужно использовать ссылку на файл, чтобы использовать одну из функций в формуле; однако надстройку необходимо обязательно установить в Excel. Надстройки рассмотрены далее в этой главе (смотри “Создание надстроек”).

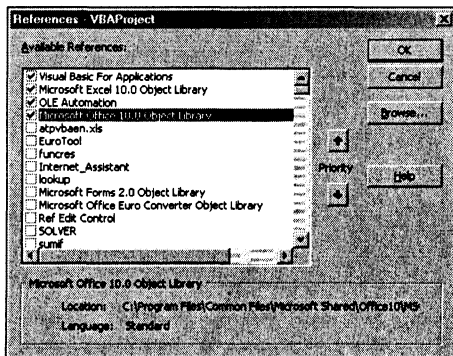


Рис. 23.1. Используйте диалоговое окно `References` для создания ссылки на проект, в котором содержится пользовательская VBA-функция

Аргументы функций

Пользовательские функции, подобно встроенным в Excel функциям, могут изменяться в зависимости от используемых в них аргументов. Придерживайтесь следующих советов при определении аргументов в VBA-функциях:

- ◆ Функция может не иметь аргументов.
- ◆ Функция может иметь фиксированное количество обязательных аргументов (от 1 до 60).
- ◆ Функция может содержать комбинацию обязательных и необязательных аргументов.



Посмотрите примеры функций, использующих различные типы аргументов, в главе 23.

Использование диалогового окна вставки функции

Диалоговое окно вставки функции Excel — это удобный инструмент, позволяющий выбирать необходимую функцию рабочей книги из списка доступных функций. Диалоговое окно вставки функции также позволяет отобразить список функций и отображает подсказку об аргументах функции.



Пользовательская функция, определенная с помощью ключевого слова `Private`, не отображается в диалоговом окне вставки функции.

По умолчанию пользовательские функции представляются в виде списка категорий определенных пользователем, но их можно отнести и к любой другой категории. Также можно добавить к функции описание.

Добавление описания функции

При выборе функции в диалоговом окне вставки функции появляется краткое описание этой функции (см. рис. 23.2).

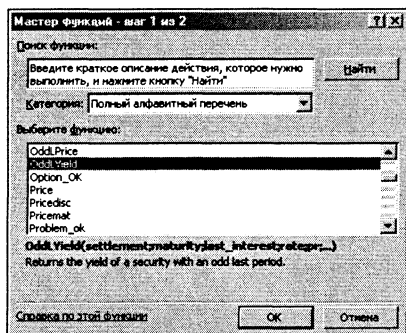


Рис. 23.2. Краткое описание выбранной функции в диалоговом окне вставки функции

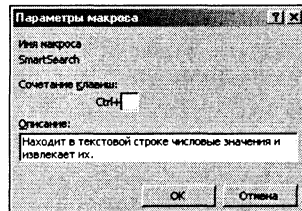


Рис. 23.3. Подготовка описания функции в диалоговом окне параметров макроса



Если не описывать функцию, в диалоговом окне вставки функции появится следующий текст: *Дополнительные сведения о функции и ее аргументах см. в справочной системе.*

Следующие инструкции рассказывают, как создать описание для пользовательской функции:

1. Создайте функцию в редакторе Visual Basic.
2. Перейдите в Excel и выберите Сервис⇒Макрос⇒Макросы (или нажмите <Alt+F8>). Диалоговое окно макроса представит в виде списка доступные внедренные процедуры, но не функции.
3. Введите имя функции в поле имени макроса. Убедитесь, что вы правильно его ввели.
4. Щелкните на кнопке Параметры для отображения диалогового окна настроек макроса. Если кнопка неактивна, вероятнее всего введено неправильное имя функции.
5. Введите описание функции в поле описания (см. рис. 23.3).
6. Щелкните на кнопке ОК, а затем на Отмена.



При использовании диалогового окна Мастер функции диалоговое окно аргументов функции появится после щелчка на кнопке ОК. Для встроенных функций в окне аргументов функции отображается описание для каждого аргумента. К сожалению, нет возможности обеспечить такое описание для пользовательской функций.

Определение категории функции

Странно, но в Excel нельзя напрямую определить пользовательскую функцию в отдельную категорию. Если вы хотите, чтобы ваша функция была определена в отдельную категорию, заданную пользователем, вам нужно дополнительно изменить VBA-код.

Например, предположим, что вы создали функцию и назвали ее COMMISSION, и вы желаете, чтобы данная функция появилась в категории Финансовые (это первая катего-

рия в диалоговом окне вставки функции. Для выполнения этого необходимо выполнить следующие изменения:

```
Application.MacroOptions Macro:="COMISSION", Category:=1
```

Один способом выполнения этого оператора является использование окна быстрого доступа редактора Visual Basic. На рис 23.4 представлен пример. Просто введите оператор и нажмите <Enter>. Потом сохраните рабочую книгу, тем самым в ней будет сохранено определение функции в указанной категории. Данный оператор необходимо выполнить только один раз. Другими словами, нет необходимости назначать категорию функции каждый раз, когда открыта книга.

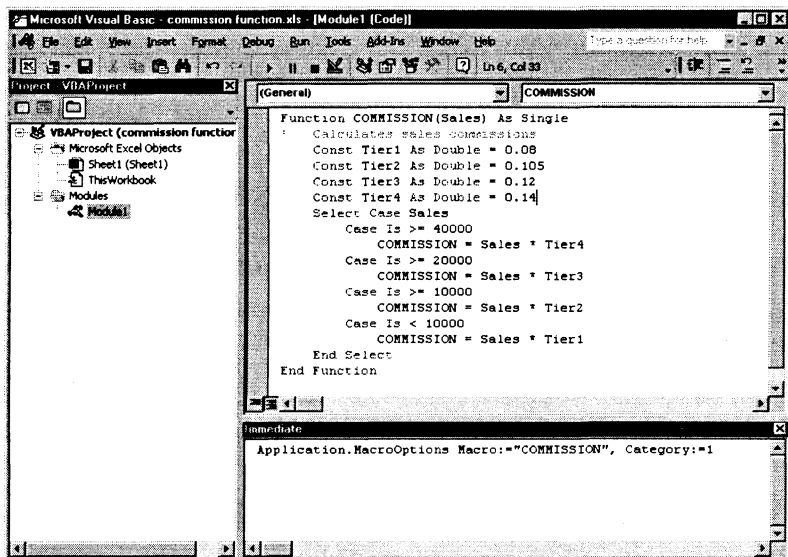


Рис. 23.4. Выполнение VBA-оператора, назначающего функции категорию

Аналогично можно создать соответствующую процедуру, и потом выполнить ее.

```
Sub AssignToFunctionCategory()  
    Application.MacroOptions Macro:="COMISSION", Category:=1  
End Sub
```

После выполнения процедуры ее можно удалить.

Конечно, можно заменить существующее имя функции и определить другую категорию функции. Процедура определения категории функции (AssignToFunctionCategory) может состоять из любого числа операторов, каждый из которых соответствует отдельной функции.

В табл. 23.1 представлен перечень категорий функций, которые можно использовать. Отметим, что несколько из этих категорий (10–13) обычно не выводятся в диалоговом окне вставки функции. При отнесении функции к одной из этих категорий она, как ни странно, отображается в диалоговом окне.

Таблица 23.1. Категории функций

Номер категории	Название категории
0	Все (категория не определена)
1	Финансовая
2	Дата и время
3	Математические
4	Статистические
5	Ссылки и массивы
6	Работа с базой данных
7	Текстовые
8	Логические
9	Проверка свойств и значений
10	Команды
11	Настройки
12	Управления макросами
13	Динамический обмен данными/внешние
14	Пользовательские
15	Инженерные

Тестирование и отладка функций

Естественно, тестирование и отладка функций — это важный этап; следует внимательно проверить правильность проведения запланированных вычислений. В этом разделе рассматриваются некоторые методы отладки, которые могут оказаться весьма полезными.



Если вы новичок в программировании, чтение данной главы будет более эффективным, если вы изучите материал предыдущей главы.

Записываемый VBA-код может содержать три основных типа ошибок:

- ◆ **Синтаксические ошибки.** Ошибкой в написании оператора, является например, неправильное введение ключевого слова, отсутствие его или несогласованные скобки. Редактор Visual Basic предоставляет возможность познакомиться с синтаксическими ошибками, отображая появляющееся на экране окно. Функцию нельзя использовать до тех пор, пока синтаксические ошибки не будут исправлены.
- ◆ **Ошибки выполнения.** Ошибки, появляющиеся во время выполнения функции. Например, попытка выполнить математическую операцию над строковой переменной приведет к такой ошибке. Если вы не знаете о ее существовании заранее, то не будете знать о ее источнике, пока она не проявится.
- ◆ **Логические ошибки.** Ошибки выполняемых кодов, возвращающих неверный результат.



Для проверки кода в VBA-модуле на наличие синтаксических ошибок, нужно выбрать Debug⇒Compile. В результате на экране отобразятся сведения о первой синтаксической ошибке, если она существует. Исправьте эту ошибку и запустите операцию еще раз для нахождения других ошибок.

Ошибка в коде иногда называется сбоем. Процесс нахождения и устранения такой ошибки называется отладкой.

При тестировании функции с использованием формулы рабочего листа сложно локализовать ошибки выполнения, поскольку (в отличие от синтаксических ошибок) они не появляются во всплывающем окне ошибок. В случае появления ошибок выполнения формула, использующая функцию, просто будет возвращать ошибочное значение (#ЗНАЧ!). В этом разделе описано несколько методов отладки пользовательских функций.



При тестировании пользовательской функции следует использовать ее только в одной формуле рабочего листа. При использовании ее в более чем одной формуле, код функции будет выполняться для каждой формулы.

Использование окна сообщений VBA

Оператор MsgBox, используемый в VBA-коде, отображает на экране специальное окно. Для контроля значения необходимых переменных можно использовать операторы MsgBox в ключевых местах кода. Следующий пример демонстрирует функцию, обращающую текстовую строку, выступающую в роли ее аргумента. Например, переданное функции слово Привет должно возвращаться как тевирП. Однако, если попробовать использовать эту функцию в формуле, то можно убедиться, что она не работает из-за логической ошибки.

```
Function REVERSETEXT(text) As String
    ' Возвращает аргумент в обратном порядке
    TextLen = Len(text)
    For i = TextLen To 1 Step -1
        REVERSETEXT = Mid(text, i, 1) & REVERSETEXT
    Next i
End Function
```

Можно вставить в код временный оператор MsgBox, который поможет понять источник проблемы. Ниже приведена функция с оператором MsgBox, вставленным внутрь цикла:

```
Function REVERSETEXT(text) As String
    ' Возвращает аргумент в обратном порядке
    TextLen = Len(text)
    For i = TextLen To 1 Step -1
        REVERSETEXT = Mid(text, i, 1) & REVERSETEXT
        MsgBox REVERSETEXT
    Next i
End Function
```

При вычислении функции появляется всплывающее окно один раз для каждой итерации цикла. В окне сообщений отображается текущее значение переменной REVERSETEXT. Другими словами, данный подход позволяет контролировать промежуточные результаты по выполнению функции. На рис. 23.5 приведен графический пример этого метода.

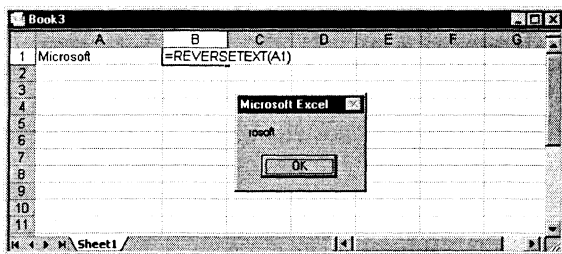


Рис. 23.5. Использование оператора MsgBox для тестирования значений во время выполнения функции

Информация, отображенная в ряде окон сообщений, показывает, что текстовая строка создается в цикле, хотя новый текст добавляется в начало строки, а не в конец. Правильный оператор присвоения имеет следующий вид:

```
REVERSETEXT = REVERSETEXT & Mid(text, i, 1)
```

При нормальной работе функции следует убедиться, что все операторы MsgBox удалены. Они очень раздражают пользователя.

Для отображения более чем одной переменной в окне сообщений, нужно объединить переменные и вставить пробел между каждой парой переменных. Ниже представлен оператор, отображающий значения трех переменных (x, y и z) в окне сообщений:

```
MsgBox x & " " & y & " " & z
```

Если пренебречь пробелами между переменными, то различить их не представляется возможным.

Использование в коде операторов Debug.Print

Если вам кажется, что операторы MsgBox достаточно надоедливы, то лучший выход из ситуации — это вставить некоторый временный код, записывающий переменные непосредственно в окно быстрого доступа редактора Visual Basic (смотри врезку “Использование окна быстрого доступа”). Для записи значений выбранных переменных нужно использовать оператор Debug.Print.

Например, если вы хотите проконтролировать значение переменных внутри цикла, нужно использовать следующий код:

```
Function VOWELCOUNT (r)
    Count = 0
    For i = 1 To Len(r)
        Ch = UCase(Mid(r, i, 1))
        If Ch Like "[AEIOU]" Then
            Count = Count + 1
            Debug.Print Ch, i
        End If
    Next i
    VOWELCOUNT = Count
End Function
```

В этом случае значения двух переменных (Ch и i) выводятся в окне быстрого доступа всякий раз, когда выполняется оператор Debug.Print. На рис. 23.6 отображен результат операции вычисления, если аргументом функции является California.

После отладки функции необходимо убедиться в том, что операторы Debug.Print удалены.

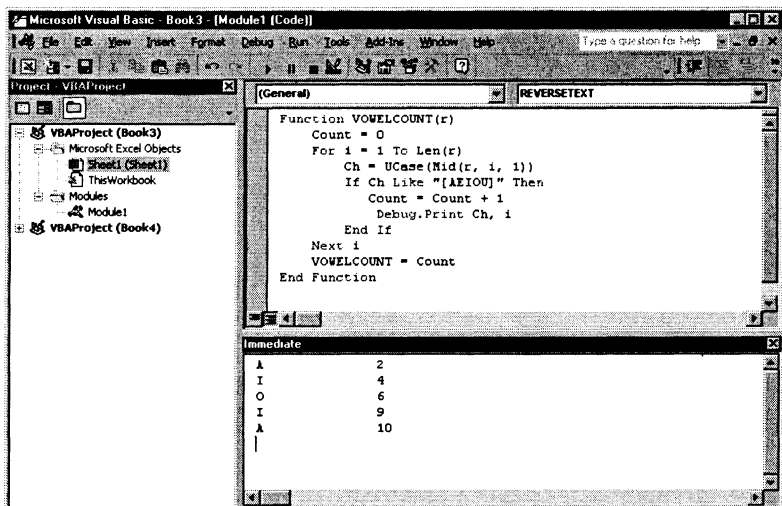


Рис. 23.6. Использование окна быстрого доступа редактора Visual Basic для отображения результатов во время работы функции

Вызов функции из процедуры

Другим способом тестирования процедуры является вызов функции из процедуры. Для этого в модуль нужно добавить временную процедуру и вставить оператор, вызывающий функцию. Это особенно удобно, поскольку ошибки выполнения будут отображаться при их появлении.

Следующая функция содержит ошибку (ошибку выполнения). Как я отмечал ранее, ошибки выполнения не отображаются во время тестирования при использовании формулы рабочего листа. В нашем случае функция возвращает ошибку (#ЗНАЧ!).

```

Function REVERSETEXT(text) As String
    ' Возвращает аргумент в обратном порядке
    TextLen = Len(text)
    For i = TextLen To 1 Step -1
        REVERSETEXT = REVERSETEXT And Mid(text, i, 1)
    Next i
End Function

```

Для установления источника ошибки выполнения можно вставить следующую процедуру:

```

Sub Test()
    x = REVERSETEXT("Hello")
    MsgBox x
End Sub

```

Эта процедура просто вызывает функцию REVERSETEXT и присваивает результат переменной, названной x. Оператор MsgBox отображает результат.

Можно выполнить процедуру непосредственно из редактора Visual Basic. Просто установите курсор в любом месте процедуры и выберите Run⇒Run Sub/UserForm (или просто нажмите <F5>). Когда процедура будет выполнена, на экране отобразится сообщение об ошибке, как показано на рис. 23.7.

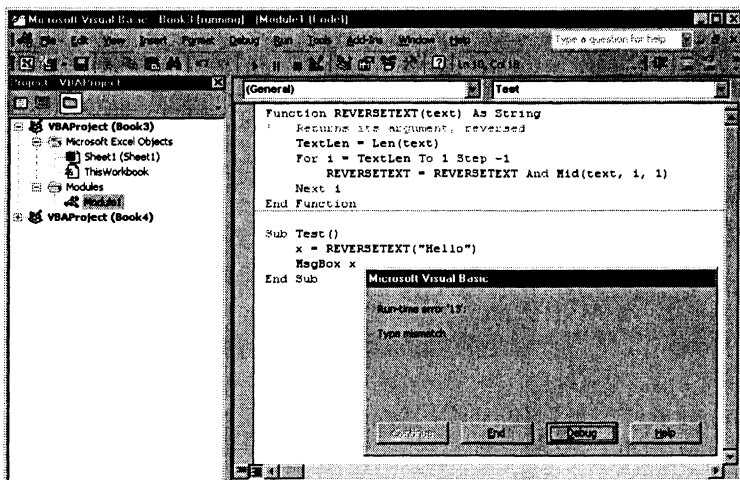


Рис. 23.7. Ошибка выполнения, определенная с помощью VBA

Щелкните на кнопке Debug, и редактор Visual Basic выделит цветом оператор, в котором предположительно содержится ошибка (см. рис. 23.8). Сообщение об ошибке не объясняет, как исправить ошибку, но оно ограничивает вас в выборе правильного решения. После определения оператора, содержащего ошибку, ее можно изучить более детально, и использовать окно быстрого доступа (см. врезку “Использование окна быстрого доступа”) для более точного нахождения причины ошибки.

В нашем случае проблема заключается в неправильном использовании оператора And (И) вместо оператора объединения (&). Правильный оператор будет иметь вид:

```
REVERSETEXT = REVERSETEXT & Mid(text, i, 1)
```



При щелчке на кнопке Debug процедура все еще выполняется, — она просто приостанавливается и переводится в “режим останова”. После внесения корректив следует нажать <F5> для продолжения ее выполнения в обычном режиме, или <F8> — в пошаговом; кроме того, можно приостановить выполнение щелчком на кнопке Rest.

Использование окна быстрого доступа

Окно быстрого доступа редактора Visual Basic очень полезно при отладке кода. Для активации этого окна необходимо нажать <Ctrl+G>.

Можно ввести VBA-оператор в окне быстрого доступа и сразу же посмотреть результат. Например, введите следующий код в окне быстрого доступа и нажмите <Enter>:

```
Print Sqr(1156)
```

Редактор Visual Basic отобразит результат операции извлечения квадратного корня (34).

Окно быстрого доступа особенно полезно при отладке ошибок выполнения, если VBA находится в режиме паузы. Например, для проверки текущих значений переменных или проверки типа данных переменной.

Часто ошибки возникают из-за неправильного задания типа данных. Например, в следующем операторе показан тип данных переменной Counter (который вероятнее всего должен быть целочисленным).

```
? TypeName(Counter)
```

При обнаружении типа данных для переменной Counter, отличной от целочисленного, проблема может быть решена вручную.

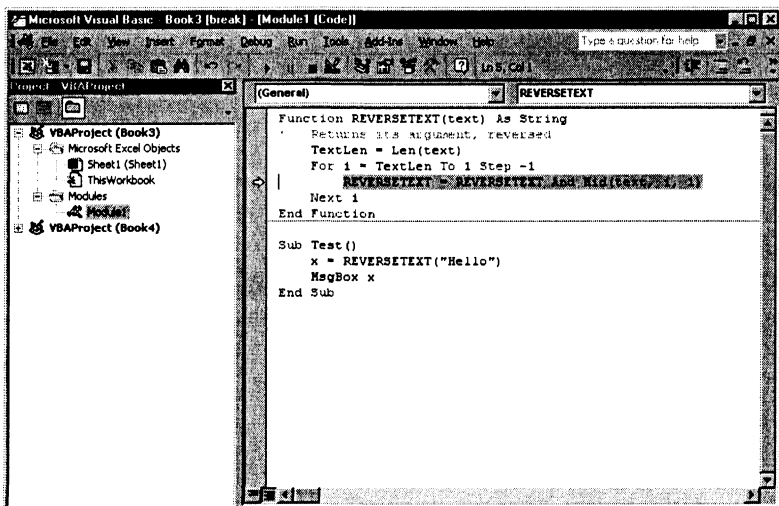


Рис. 23.8. Выделенный оператор с ошибкой выполнения

Установка точек останова в функции

Задание точки останова в коде — еще один режим отладки. Выполнение кода приостанавливается, когда VBA находит точку останова. Для проверки значений переменных можно воспользоваться окном быстрого доступа или использовать <F8> для пошаговой проверки кода.

Для определения точки останова необходимо установить курсор в коде оператора, выполнение которого необходимо приостановить, затем выбрать Debug⇒Toggle Breakpoint. Также можно нажать <F9> или же щелкнуть кнопкой мышки на вертикальной полосе, расположенной в левой части окна кода. Все эти действия будут выделять цветом оператор, напоминая о том, что точка останова активизирована (точку видно в левой части окна кода). В коде можно задать любое количество точек останова. Для удаления точки останова нужно установить курсор в код оператора, содержащий ее, и нажать <F9>. На рис. 23.9 показана функция, содержащая точку останова.

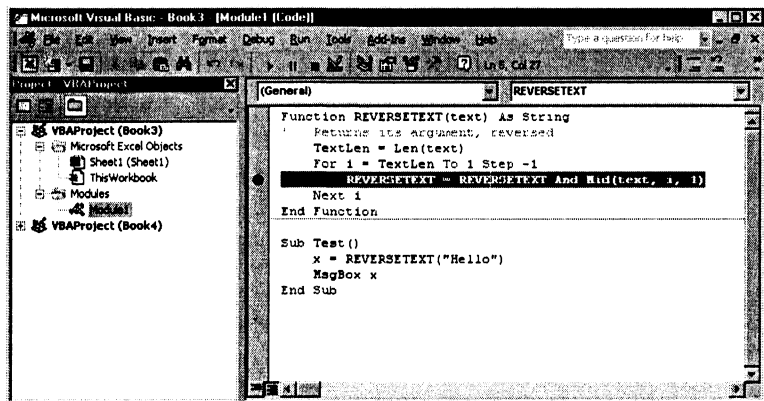


Рис. 23.9. Выделенный оператор содержит точку останова

Создание надстроек

При создании функций, используемых достаточно часто, их можно разместить в файле надстройки. Основное преимущество подобной операции — предоставление возможности использования функции в формулах любой рабочей книги без применения спецификатора имени файла.

Допустим, у вас есть функция ZAPSPACES и она находится в файле Myfuncs.xls. Вам нужно ввести следующую формулу:

```
=Myfuncs.xls!ZAPSPACES(A1:C12)
```

При создании надстройки на основе файла Myfuncs.xls и сохранения ее можно преобратить ссылкой на файл и ввести формулу следующим образом:

```
=ZAPSPACES(A1:C12)
```

Создание надстроек на основе рабочей книги — это достаточно простая задача. Следующие инструкции описывают, как создать надстройку из обычного файла рабочей книги:

1. Создайте функции и убедитесь, что они работают надлежащим образом.
2. Активизируйте редактор Visual Basic и выберите рабочую книгу в окне проекта. Выберите Tools⇒xxx Properties и перейдите на вкладку Protection. Выставьте флажок опции Lock Project for Viewing и дважды введите пароль. Щелкните на кнопке ОК.

Эту операцию нужно выполнить только в том случае, если вы хотите предохранить ваш макрос от вмешательства в него или в диалоговые окна постороннего пользователя.

3. Возобновите работу Excel. Выберите команду Файл⇒Свойства, щелкните на вкладке Документ и введите краткое описательное название в поле Название, и более длинное описание в поле Заметки.

Этот этап не обязателен, но он делает надстройку проще в использовании, отображая описательный текст в диалоговом окне подключения надстроек.

4. Выберите в меню команду Файл⇒Сохранить как.
5. В диалоговом окне сохранения выберите из списка типов файла опцию Надстройка Microsoft Excel (*.xla) (рис. 23.10).
6. Если вы не хотите использовать папку, определяемую по умолчанию, можете указать другую.
7. Щелкните на кнопке Сохранить. Копия рабочей книги будет сохранена (с расширением .xla) как надстройка, а исходная рабочая книга останется открытой.

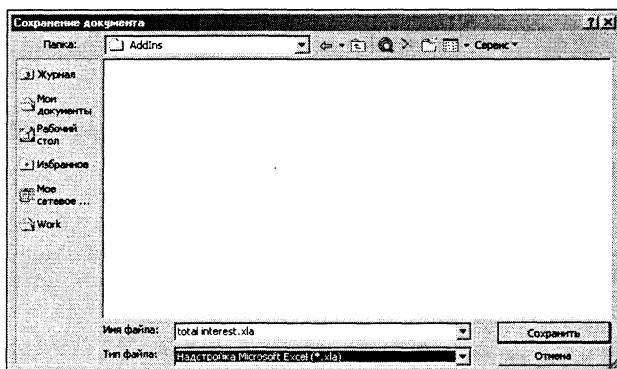


Рис. 23.10. Сохранение рабочей книги как надстройки

Несколько слов о безопасности

Компания Microsoft никогда не рекламировала продукт Excel как создающий приложения с безопасным исходным кодом. Особенности пароля не обеспечивают достаточный уровень защиты Excel от несанкционированного доступа других пользователей к приложениям, которые вы хотели бы сохранить недоступными. Существует несколько программ по взлому таких паролей. Средства защиты в Excel 2002 более надежные, чем в предыдущих версиях, но существует возможность, что их тоже можно взломать. Если вы абсолютно не уверены в том, можно ли допускать разглашение вашего кода или формул, то Excel — далеко не лучшая программа для разработки.



Рабочая книга, преобразуемая в надстройку, должна, по крайней мере, содержать хотя бы один рабочий лист. Например, если рабочая книга состоит только из диаграмм Excel 5/95, то опция Надстройка Microsoft Excel (*.xla) в диалоговом окне Сохранение документа будет недоступна.



В предыдущих версиях Excel (до Excel 97) для изменения надстроек нужно было открывать оригинальные XLS-файлы, вносить изменения и повторно сохранять надстройку. В Excel 97 и более поздних версиях в этом нет необходимости. До тех пор, пока надстройка не защищена, в нее можно вносить изменения в редакторе Visual Basic и затем сохранять эти изменения. В случае, если надстройка защищена, необходимо вводить пароль для снятия защиты. Поэтому, при работе в Excel 97 и более поздних версиях, в хранении XLS-версии надстройки нет необходимости.

После создания надстройки можно установить ее, используя стандартную процедуру: выберите Сервис⇒Надстройки и щелкните в появившемся диалоговом окне на кнопке Обзор. Укажите расположение файла *.xla.

Резюме

В этой главе рассмотрены специальные методы, которые могут пригодиться при разработке эффективных функций. Вами рассмотрены типы аргументов, которые можно использовать. В главе описаны способы добавления пользовательской функции в категорию диалогового окна вставки функции. В данной главе также представлены методы отладки функций, инструкции по созданию надстроек, необходимые для внешнего хранения функций.

В следующей главе речь пойдет об основных способах программирования на языке VBA.

Принципы программирования на VBA

В этой главе...

- ◆ Пример процедуры функции
- ◆ Использование комментариев в коде
- ◆ Использование переменных, типов данных и констант
- ◆ Использование выражений присвоения
- ◆ Использование массивов
- ◆ Использование встроенных функций VBA
- ◆ Управление выполнением кода
- ◆ Работа с диапазонами
- ◆ Резюме

В этой главе описываются основные элементы языка и принципы программирования на VBA. Если до изучения VBA вами уже изучался другой язык программирования, то большая часть излагаемого материала будет вам известна. Но в VBA имеются некоторые нюансы, поэтому даже опытные программисты смогут найти для себя что-то новое.

Пример процедуры функции

Для начала рассмотрим пример процедуры функции. Функция, которая называется `REMOVE SPACES`, принимает текстовый аргумент и возвращает его без пробелов. Например, в приведенном ниже коде функцией `REMOVE SPACES` возвращается значение `Thisistext`.

```
=REMOVE SPACES ("Thisistext")
```

Чтобы написать такую функцию, откройте модуль VBA и в окне кода напишите следующую процедуру:

```
Function REMOVE SPACES (cell) As String
    Удаляет пробелы из ячейки
    Dim CellLength As Integer
    Dim Temp As String
    Dim i As Integer
    CellLength = Len (cell)
    Temp = ""
    For i=1 To CellLength
        Character = Mid (cell,i,1)
        If Character <> Chr (32) Then Temp = Temp&Character
    Next i
    REMOVE SPACES = Temp
End Function
```

Рассмотрим подробно каждую строку кода:

- ◆ Первая строка функции называется *строкой объявления*. Обратите внимание, что процедура начинается с ключевого слова `Function`, после которого стоит название функции (`REMOVESPACES`). В функции используется один аргумент (`cell`); имя аргумента заключается в круглые скобки. Выражение `As String` определяет тип возвращаемого значения функции. Слово `As` вводить необязательно.
- ◆ Вторая строка является обычным комментарием (необязательная строка) — на это указывает апостроф, стоящий в начале строки. Комментарий описывает, что делает функция.
- ◆ В следующих трех строках используется ключевое слово `Dim` для объявления трех переменных, используемых в процедуре: `CellLength`, `Temp` и `i`. Описывать переменные не обязательно, но (как показано далее) это хороший тон в программировании.
- ◆ В следующей строке процедуры переменной `CellLength` присваивается значение. Для определения длины содержимого аргумента (`cell`) используется оператор `Len`.
- ◆ В следующей строке создается переменная `Temp`, которой присваивается пробел.
- ◆ В следующих четырех строках описан цикл `For-Next`. Операции между `For` и `Next` выполняются столько раз, сколько задано значением `CellLength`. Например, предположим, что аргумент `cell` содержит строку `Bob Smith`. Операции в цикле повторяются девять раз, по одной итерации для каждого символа в строке.
- ◆ В цикле переменная `Character` содержит текущий символ, который описывается функцией `Mid`. Оператор `If` определяет, является ли символ пробелом (функция `Chr` в VBA эквивалентна функции СИМВОЛ в Excel, а аргумент `32` соответствует пробелу). Если символ не является пробелом, то он подставляется в конец строки, сохраняемой в переменной `Temp`. Если символ оказался пробелом, то переменная `Temp` не изменяется и процедура переходит к следующему символу. При желании можно заменить этот оператор следующим:

```
If Character <> " " Then Temp = Temp&Character
```

- ◆ По завершении цикла в переменной `Temp` содержатся все символы, которые сначала были указаны в аргументе `cell`, кроме пробелов.
- ◆ Строка с переменной `Temp` присваивается имени функции. Эта строка является значением, возвращаемым функцией.
- ◆ Процедура `Function` завершается оператором `End Function`.

В процедуре `REMOVESPACES` используются следующий распространенные элементы языка VBA:

- ◆ Комментарии (в начале строки стоит апостроф)
- ◆ Объявление переменных
- ◆ Три оператора присвоения
- ◆ Три встроенные функции VBA (`Len`, `Mid` и `Chr`)
- ◆ Циклическая структура (`For-Next`)
- ◆ Условная структура `If-Then`
- ◆ Объединение строк (с помощью оператора `&`)

Неплохо для первого раза, да? Примечания в этой главе содержат более подробную информацию об этих и других принципах программирования.



Функция `REMOVESPACES`, описанная здесь, предназначена только для учебных целей. Того же результата можно добиться при использовании функции `ПОДСТАВИТЬ`, которая намного рациональнее. Например, следующая формула удаляет все пробелы в тексте в ячейке `A1`.

```
=ПОДСТАВИТЬ (A1 ; " " ; "")
```

Использование комментариев в коде

Комментарием называется текст описания, вставленный в код. В VBA текст комментария не выполняется. Используйте комментарии для описания своих действий (поскольку предназначение некоторых операций VBA не всегда очевидно).

Комментарий можно начать с новой строки или написать их в одной строке с операциями, но только *после* них. Признаком комментария служит апостроф. VBA игнорирует любой текст до конца строки, который находится после апострофа. Исключениями являются апострофы внутри кавычек. Например, в следующем операторе нет комментария, хотя в нем содержится апостроф:

```
Result = "Can't calculate"
```

В следующем примере написана процедура функции VBA с тремя комментариями:

```
Function MYFUNC ()  
' Эта функция не делает ничего кроме возвращения  
  x=0 'x представленного константой  
' Возвращение результата  
  MYFUNC = x  
End Function
```

Созданную функцию можно проверить, включив в нее дополнительные операторы. Не удаляйте впоследствии операторы тестирования, а преобразуйте их в комментарий, поставив в начале строки апостроф. Во время выполнения процедуры VBA проигнорирует подобные инструкции. Чтобы преобразовать комментарии обратно в операторы, удалите апостроф.



На панели инструментов `Edit` находятся две очень полезные кнопки. Выделите группу операторов и воспользуйтесь кнопкой `Comment Block`, чтобы преобразовать операторы в комментарий. Кнопка `Uncomment Block` преобразует группу комментариев обратно в операторы.

Использование переменных, типов данных и констант

Переменной называется место в памяти компьютера, у которого есть имя. Переменные могут быть различных *типов данных* — от простых значений логического типа `Boolean` (`TRUE` или `FALSE`) до больших значений с двойной точностью (читайте в следующем разделе). Переменной задается значение с помощью оператора присвоения, которым является знак равенства.

Далее приведено несколько примеров использования операторов присвоения для различных типов переменных. Имена переменных находятся слева от знака равенства. Значение каждой переменной находится справа от знака равенства.

```

x = 1
InterestRate = 0.075
LoanPayoffAmount = 243089
DataEntered = False
x = x + 1
MyNum = YourNum * 1.25
HallOfFamer = "Tony Gwynn"
DateStarted = #3/14/2001#

```

В VBA есть множество *зарезервированных слов*, их нельзя использовать в качестве имен переменных и процедур. Если попытаться использовать одно из этих слов, появится сообщение об ошибке. Например, хотя зарезервированное слово `Next` может быть описательным именем переменной, следующий оператор вызовет синтаксическую ошибку:

```
Next = 132
```

К сожалению, иногда сообщения о синтаксических ошибках ни о чем не говорят, т.е. не показательны. Предыдущий оператор вызывает синтаксическую ошибку в Excel 2002 (ранние версии Excel могут выдать другой тип ошибки). Поэтому если оператор присвоения вызывает ошибку, которая не имеет смысла, обратитесь к справочной системе, чтобы проверить, не имеет ли имя переменной специального назначения в VBA.

Определение типов данных

VBA облегчает программистам жизнь, потому что он автоматически регулирует характеристики данных. *Тип данных* говорит о том, в каком виде данные хранятся в памяти — как целые или действительные числа, строковые переменные и т.д.

Хотя VBA автоматически заботится о типизации данных, все не так просто — при этом понижается скорость выполнения формул и снижается эффективность использования памяти. Чтобы достичь оптимальной производительности выполнения операций, следует разобраться с типами данных. В принципе, лучше всего работать с данными, которые занимают меньше всего байт в памяти. Когда VBA работает с данными, скорость выполнения операций зависит от количества байт, находящихся в распоряжении VBA. Другими словами, чем меньше памяти используется данными, тем быстрее VBA получает и оперирует данными. В табл. 24.1 приведены встроенные типы данных VBA.

Таблица 24.1. Типы данных VBA

Тип данных	Объем занимаемой памяти	Диапазон принимаемых значений
Byte	1 байт	От 0 до 255
Boolean	2 байта	TRUE или FALSE
Integer	2 байта	От -32 768 до 32 767
Long	4 байта	От -2 147 483 648 до 2 147 483 648
Single	4 байта	От -3,402823E38 до -1,401298E-45 (для отрицательных значений); от 1,401298E-45 до 3,402823E38 (для положительных значений)
Double	8 байт	-1,79769313486231E308 до -4,94065645841247E-324 (для отрицательных значений); от 4,94065645841247E-324 до 1,79769313486231E308 (для положительных значений)
Currency	8 байт	От -922 337 203 685 477,5808 до 922 337 203 685 477,5808

Тип данных	Объем занимаемой памяти	Диапазон принимаемых значений
Decimal	14 байт	±79 228 162 514 264 337 493 543 950 335 без запятой; ±7,9228162514264337493543950335 с 28 символами после запятой
Date	8 байт	С 1 января 0100 года по 31 декабрь 9999 года
Object	4 байта	Любая связь с объектом
String (переменная длина)	10 байт + длина строки	От 0 до примерно 2 миллиардов
String (фиксированная длина)	Длина строки	От 1 до примерно 65 400
Variant (с числами)	16 байт	Любое числовое значение вплоть до диапазона значений двойной точности
Variant (с символами)	22 байта + длина строки	От 0 до примерно 2 миллиардов

Объявление переменных

Переменную, перед использованием в процедуре, нужно *объявить*. В объявлении переменной указывается ее имя и тип, при этом очевидны два преимущества:

- ◆ *Процедуры выполняются быстрее, а память используется рациональнее.* Используемый по умолчанию тип данных — `variant` — приводит к многократным проверкам, занимающим много времени и памяти. Если в VBA указать специальный тип переменной, то переменную не нужно проверять; под нее выделяется столько памяти, сколько нужно для хранения ее значений.
- ◆ *Если использовать оператор `Option Explicit`, можно избежать проблем неправильного написания имен переменных.* Предположим, что использовалась необъявленная переменная `CurrentRate`. В определенном месте процедуры вводится выражение `CurrentRate = ,075`. Из-за пропущенного объявления переменной функция может вернуть неправильный результат. Читайте врезку “Привычка объявления переменных”.

Переменные объявляются с помощью ключевого слова `Dim`. Например, в следующей строке описывается переменная `Count` типа `integer`.

```
Dim Count As Integer
```

Оператором `Dim` можно объявить сразу несколько переменных. Например,

```
Dim x As Integer, y As Integer, z As Integer
Dim First As Long, Last As Double
```



В отличие от других языков, VBA не позволяет указать тип данных для группы переменных, разделяя их запятыми. Например, следующее выражение — хотя и правильное — не описывает все переменные как `integer`:

```
Dim i, j, k As Integer
```

В этом выражении только переменная `k` описана как `integer`. Чтобы для всех переменных указать тип `integer`, следует объявить их следующим образом:

```
Dim i As Integer, j As Integer, k As Integer
```

Если тип переменной не указывается, то VBA использует стандартный тип данных — `variant`. Данные, описанные как `variant`, ведут себя весьма изменчиво: они меняют тип в зависимости от выполняемых действий. В следующей процедуре показано, как переменная принимает различные типы данных:

```
Function VARIANT_DEMO ()  
    MyVar = "123"  
    MyVar = MyVar / 2  
    MyVar = "Answer: " & MyVar  
    VARIANT_DEMO = MyVar  
End Function
```

Привычка объявления переменных

Чтобы приучить себя объявлять все используемые переменные, первым оператором в модуле VBA используйте следующий:

```
Option Explicit
```

Этот оператор останавливает выполнение процедуры, как только VBA встречает необъявленную переменную. VBA выводит сообщение об ошибке (*Compile error: Variable not defined*) и предлагает объявить переменную, прежде чем выполнение процедуры продолжится.

Чтобы удостовериться, что оператор `Option Explicit` вводится в каждом новом модуле VBA, выставьте опцию `Require Variable Declaration` на вкладке `Editor` диалогового окна `Options`.

В примере функции `VARIANT_DEMO` переменная `MyVar` описывается как трехсимвольная строка, выглядящая как число. Затем эта “строка” делится на два, и `MyVar` приобретает числовой тип данных. Потом `MyVar` добавляется к строке, что преобразует ее снова в строку. Функция возвращает строку: *Answer: 61,5*.

Использование констант

Значение переменной может — и часто так и происходит — изменяться в процессе выполнения процедуры (поэтому она и называется *переменной*). Иногда необходимо сослаться на имя переменной или строки, которая никогда не меняется; другими словами, сослаться на *константу*.

Константа описывается оператором `Const`. Вот несколько примеров объявления констант:

```
Const NumQuarters as Integer = 4  
Const Rate = .0725, Period = 12  
Const CompanyName as String = "Acme Snaopholytes"
```

Во второй строке две переменные объявляются одним оператором, но он не указывает их тип. Следовательно, обе константы имеют тип `variant`. Поскольку константа никогда не меняет свое значение, для нее нужно обязательно указывать тип. *Область действия* константы зависит от того, в каком месте модуля она описана:

- ◆ Чтобы константа использовалась только в одной процедуре, ее нужно объявить после операторов `Sub` и `Function`; при этом константа становится локальной.
- ◆ Чтобы константа была доступна для всех процедур в модуле, объявите ее перед первой процедурой модуля.
- ◆ Чтобы сделать константу доступной для всех модулей в рабочей книге, используйте ключевое слово `Public` и объявите константу перед первой процедурой модуля. Следующее выражение создает константу, используемую во всех модулях VBA в рабочей книге:

```
Public AppName As String = "Budget Tools"
```



Если в процедуре VBA значение константы попытаться изменить появится сообщение об ошибке, что, в общем-то, и должно происходить. Константа есть константа, а не переменная.

Хорошей привычкой является использование констант в коде для фиксированных значений или строк. Например, если процедура ссылается на определенное значение (скажем, на процентную ставку) несколько раз, то это значение лучше всего описать как константу и в выражениях вместо значения использовать ее имя. Так код легче читать и вносить в него изменения — при необходимости нужно изменить только один оператор, а не несколько.

Использование строк

Как и Excel, VBA может оперировать как с числами, так и с текстом (строками). В VBA поддерживается два типа строк:

- ◆ *Строки фиксированной длины.* Объявляются с указанием количества символов. Максимальная длина составляет 65 535 символов.
- ◆ *Строки переменной длины.* Теоретически могут содержать до 2 миллиардов символов.

Каждый символ строки занимает 1 байт памяти. Если длина строки известна заранее (т.е. строка с фиксированной длиной), то ее можно указать при объявлении с помощью оператора Dim; или VBA определит ее длину динамически (строка с переменной длиной). В некоторых случаях память используется более рационально при работе со строками фиксированной длины.

В следующем примере переменная MyString объявлена как string с фиксированной длиной 50 символов. Переменная YourString тоже объявлена типом string, но ее длина не указывается.

```
Dim MyString As String * 50
Dim YourString As String
```

Работа с датами

Строчковую переменную можно использовать для описания дат, но с помощью этой переменной нельзя вычислить ту или иную дату. Для работы с датами воспользуйтесь типом Date.

Под переменную, объявленную типом Date, отводится 8 байт памяти, и в ней содержатся данные в диапазоне от 1 января 0100 года по 31 декабря 9999 года. Это промежуток примерно в 10000 лет — более чем достаточно даже для самого долгосрочного финансового прогноза! Тип данных Date также используется для данных времени. Даты и время в VBA заключаются между двумя знаками решетки (#).



Диапазон дат, поддерживаемых в VBA, намного больше, чем собственный диапазон дат в Excel, который начинается с 1 января 1900. Поэтому будьте внимательны и не используйте дату, которая не входит в диапазон дат, приемлемых в Excel.

Далее приведены несколько примеров объявления переменных и констант типом Date:

```
Dim Today As Date
Dim StartTime As Date
Const FirstDay As Date = #1/1/2002#
Const Noon = #12:00:00#
```




Переменные, описанные типом Date, показывают даты в соответствии с кратким системным форматом даты, а время отображается в соответствии с системным форматом времени (либо 12-часовым, либо 24-часовым). Системные настройки можно изменить с помощью апплета Язык и региональные стандарты папки Панель управления в Windows.

Использование выражений присвоения

Выражением присвоения называется оператор VBA, в котором вычисляется результат и присваивается переменной или объекту. *Выражением* называется комбинация ключевых слов, операторов, переменных и констант, которая возвращает строки, числа и объекты. Выражение может выполнить вычисление, оперировать символами или проверять данные.

Если у вас не возникало трудностей с созданием формул в Excel, то при создании выражений в VBA тоже их не должно добавиться. При введении формулы на рабочем листе в Excel результат отображается в ячейке. Подобно этому выражение VBA можно присвоить переменной или использовать его в качестве значения операнда.

В VBA в качестве оператора присвоения используется знак равенства (=). Обратите внимание на следующие примеры присвоения (выражения расположены справа от знака равенства):

```
x = 1
x = x + 1
x = (y * 2) / (z * 2)
MultiSheets = True
```

В выражениях часто используются функции. Это могут быть встроенные функции VBA, функции рабочих листов Excel или пользовательские функции, созданные в VBA. О встроенных функциях VBA рассказывается далее в этой главе.

Главную роль в VBA играют операторы. Математические действия описываются уже знакомыми операторами, а именно: суммирования (+), умножения (*), деления (/), вычитания (-), возведения в степень (^) и объединение строк (&). Менее известный оператор, например, обратная косая (\), используется при целочисленном делении, а оператор Mod используется в модульной арифметике. Оператор Mod возвращает остаток деления двух целых чисел. Например, результат следующего выражения равняется 2:

```
17 Mod 8
```

Возможно, вам уже знакома функция MOD. Заметьте, что в VBA Mod является оператором, а не функцией.

VBA поддерживает операторы сравнения, используемые в формулах Excel: равно (=), больше чем (>), меньше чем (<), больше или равно (>=), меньше или равно (<=) и неравно (<>). Кроме того, VBA предлагает полный набор логических операторов, приведенных в табл. 24.2. Читайте справочную систему, чтобы получить дополнительную информацию и примеры по работе этих операторов.

Таблица 24.2. Логические операторы VBA

Оператор	Действие
Not	Выполняет логическое отрицание в выражении
And	Выполняет логическое объединение двух выражение
Or	Выполняет логическое вычитание двух выражений
Xor	Выполняет логическое исключение в двух выражениях

Оператор	Действие
<code>Eqv</code>	Выполняет логическое приравнивание двух выражений
<code>Imp</code>	Выполняет логическое включение двух выражений

Приоритет выполнения операторов в VBA такой же, как и в Excel. Но можно поставить круглые скобки и изменить его.

Использование массивов

Массивом называется группа элементов одного типа с одним именем; обращение к определенному элементу массива осуществляется с помощью его имени и индекса элемента. Например, определим массив из 12 строковых переменных, в котором каждая переменная является именем одного из 12 месяцев. Если массив назвать `MonthNames`, то к первому элементу можно обратиться, написав `MonthNames(0)`, ко второму — `MonthNames(1)` и так далее до `MonthNames(11)`.

Объявление массива

Массив объявляется оператором `Dim` или `Public`, как и обычная переменная. В массиве можно объявить сразу несколько элементов. Для этого нужно указать первый индекс, а затем ключевое слово `To` и последний индекс — все в круглых скобках. Например, вот как объявляется массив, состоящий из 100 целых чисел:

```
Dim MyArray(1 To 100) As Integer
```

При объявлении массива можно указать только первый его элемент, при этом в VBA (по умолчанию) предполагается, что 0 — первый элемент. Следовательно, выражения ниже приводят к одинаковому результату:

```
Dim MyArray(0 to 100) As Integer
Dim MyArray(100) As Integer
```

В обоих случаях массив состоит из 101 элемента.

Если первым элементом для всех массивов, в которых указан только их последний элемент, должна выступать единица, то перед любой процедурой в модуле введите следующее выражение:

```
Option Base 1
```

Если такое выражение присутствует в модуле, то следующие два выражения выдадут одинаковый результат (в обоих описывается массив из 100 элементов):

```
Dim MyArray(1 to 100) As Integer
Dim MyArray(100) As Integer
```

Описание многомерных массивов

В предыдущем разделе приведены примеры одномерных массивов. Массивы в VBA могут иметь до 60 размерностей, хотя редко используется больше 3-х (3-D массив). В следующем выражении описывается двумерный (2-D) массив, состоящий из 100 целых чисел:

```
Dim MyArray(1 To 10, 1 To 10) As Integer
```

Такой массив похож на матрицу размером 10x10. Чтобы обратиться к определенному элементу двумерного массива, следует указать два индекса. Вот как присвоить значение элементу вышеописанного массива:

```
MyArray(3,4) = 125
```

В динамическом массиве нет заранее заданного количества элементов. В случае динамического массива в круглых скобках ничего не ставится:

```
Dim MyArray() As Integer
```

Однако перед использованием динамического массива в коде необходимо воспользоваться оператором ReDim, чтобы указать VBA, сколько элементов находится в массиве (или ReDim Preserve, если в массиве нужно сохранить существующие значения). Оператор ReDim можно использовать неоднократно, изменяя размерность массива столько раз, сколько это необходимо.

Более подробно массивы рассмотрены далее в этой главе в разделах, в которых повествуется о циклах.

Использование встроенных функций VBA

В VBA есть множество встроенных функций, упрощающих вычисление и выполнение операций. Многие из функций VBA схожи (или идентичны) с функциями рабочих листов Excel. Например, функция VBA UCase, которая изменяет регистр строкового аргумента с нижнего на верхний, эквивалентен функции ПРОПИСН рабочего листа Excel.



Чтобы отобразить список функций VBA во время написания кода, введите VBA и поставьте точку. Редактор VBA отобразит список всех функций (рис. 24.1). Если ничего не произошло и список не отобразился, убедитесь, что активизирован параметр Auto List Members. Выберите Tools⇒Options и перейдите на вкладку Editor. Кроме функций в отображенный список входят встроенные константы. Функции VBA описаны в справочной системе. Чтобы просмотреть справку, подведите указатель к имени функции и нажмите клавишу <F1>.

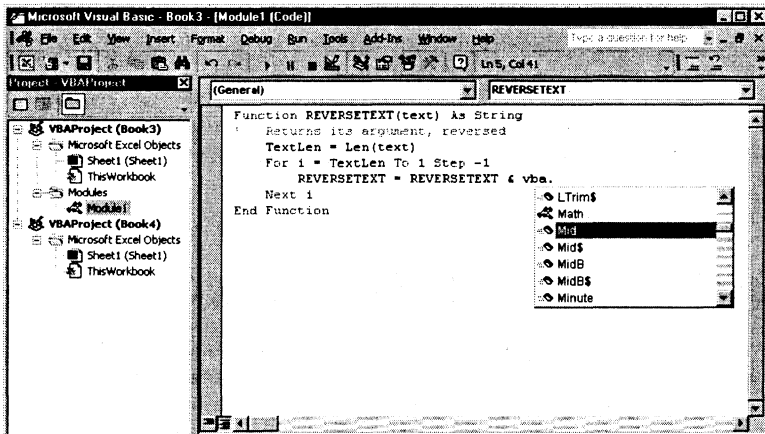


Рис. 24.1. Отображения списка функций VBA в редакторе VBA

Ниже приведено выражение, в котором вычисляется квадратный корень с помощью функции `Sqr`, а результат присваивается переменной `x`.

```
x = Sqr(MyValue)
```

Имея представление о функциях VBA, можно избавить себя от большого объема работы. Например, рассмотрим процедуру функции `REMOVESPACES`, о которой рассказывалось в начале этой главы. В этой функции для проверки каждого символа строки используется цикл `For-Next`, а затем создается новая строка. Намного проще (и более эффективно) использовать функцию `Replace`. Ниже приведена новая версия процедуры функции.

```
Function REMOVESPACES(cell) As String
    Удаляет пробелы из ячейки
    REMOVESPACES = Replace(cell, " ", "")
End Function
```



Функция `Replace` появилась в VBA в Excel 2000. Этой функции нет в более ранних версиях Excel.

В коде VBA можно использовать многие (но не все) функции рабочего листа Excel. Чтобы в выражении VBA использовать функцию рабочего листа, предверьте имя функции словом `WorksheetFunction` и точкой.



Для совместимости с более ранними версиями Excel вместо `WorksheetFunction` используйте `Application`. Объект `WorksheetFunction` появился в Excel 97, он не поддерживается в Excel 95. Следующие выражения являются эквивалентными:

```
Result = Application.Max(x, y, z)
Result = WorksheetFunction.Max(x, y, z)
```

Нижеприведенный код демонстрирует использование функции Excel в выражении VBA. Редко используемая в Excel функция `Римское` (`Roman`) преобразует десятичные числа в римские.

```
DemValue = 1999
RValue = WorksheetFunction.Roman(DecValue)
```

Переменная `RValue` содержит строку `MCMXCIX`. Пользователей приводит в замешательство тот факт, что в Excel нет функции преобразования римских чисел в их десятичный эквивалент. Конечно, можно создать такую функцию. Но стоит ли?

Важно понять, что функции Excel, для которых представлен эквивалент в VBA, нельзя использовать. Например, в VBA нет функции рабочего листа `SQRT`, потому что в VBA есть собственная версия этой функции — `Sqr`. Таким образом, следующее выражение вызовет ошибку:

```
x = Application.SQRT(123) 'ошибка
```

Управление выполнением кода

Некоторые процедуры VBA выполняются от начала и до конца кода. Однако иногда нужно контролировать процесс выполнения, пропуская одни выражения, выполняя другие несколько раз и проверяя условия, чтобы определить, что же будет выполнено далее.

В этом разделе рассмотрено несколько способов контроля за выполнением процедур VBA:

- ◆ Конструкция `If-Then`
- ◆ Конструкция `Select Case`

- ◆ Цикл For-Next
- ◆ Цикл Do While
- ◆ Цикл Do Until
- ◆ Выражение On Error

Конструкция If-Then

Вероятно, наиболее часто используемой конструкцией в VBA является If-Then. Эта конструкция является единственным случаем, где выражению предоставляется возможность выбора. В конструкции If-Then используется следующий синтаксис:

If условие Then выполняемые в случае истинности операторы [Else выполняемые в случае ложности операторы]

Конструкция If-Then выполняет один или несколько операторов в зависимости от справедливости условия. Оператор Else вводить необязательно. Если он добавлен, то выполняются специальные операторы даже в случае ложности условия.

В примере следующей процедуры функции структура If-Then представлена без оператора Else. Это пример со временем суток. Время суток выражено дробным значением — например, полдень записывается как 0,5. Функция Time в VBA возвращает значение, которое показывает время суток, задаваемое системными часами. В следующем примере функция начинается с присвоения пустой строки переменной GreetMe. Выражение If-Then проверяет время суток. Если время до полудня, то часть Then выполняется и функция возвращает значение *Good Morning*.

```
Function GreetMe()
    GreetMe = ""
    If Time < 0.5 Then GreetMe = "Good Morning"
End Function
```

В следующей функции используется две конструкции If-Then. Эта функция возвращает либо *Good Morning*, либо *Good Afternoon*:

```
Function GreetMe()
    If Time < 0.5 Then GreetMe = "Good Morning"
    If Time >= 0.5 Then GreetMe = "Good Afternoon"
End Function
```

Обратите внимание, что во второй раз в конструкции If-Then использовался символ >= (больше или равно). В этом случае при выполнении функции учитывается точное время 12:00.

Другим способом является использование в конструкции If-Then оператора Else. Например:

```
Function GreetMe()
    If Time < 0.5 Then GreetMe = "Good Morning" Else _
    GreetMe = "Good Afternoon"
End Function
```

Заметьте, что в этом примере используется символ продолжения строки (пробел и подчеркивание); If-Then-Else является одним выражением.

Далее приведен еще один пример использования конструкции If-Then. Процедура функции вычисляет скидку в зависимости от количества (предполагается целое число) товара. Она принимает один аргумент (количество) и, с учетом этого значения, возвращает соответствующую скидку.

```
Function Discount(quantity)
    If quantity <= 5 Then Discount = 0
```

```

If quantity >= 6 Then Discount = 0.1
If quantity >= 25 Then Discount = 0.15
If quantity >= 50 Then Discount = 0.2
If quantity >= 75 Then Discount = 0.25
End Function

```

Обратите внимание, что в этой процедуре выражение If-Then всегда выполняется, а значение Discount изменяется по мере выполнения функции. Возвращаемое значение является конечным.

Во всех приведенных выше примерах конструкции If-Then в выражении Then используется один оператор. Но часто бывает нужно выполнить несколько действий, если условие является верным. Воспользоваться конструкцией If-Then можно и в этом случае, но для обозначения последней операции в блоке Then нужно использовать выражение End If. Ниже приведен пример выполнения двух выражений в случае, когда условие справедливо.

```

If x > 0 Then
    y = 2
    z = 3
End If

```

Можно использовать много операторов в конструкции If-Then-Else. Следующий пример демонстрирует два выражения при истинном условии и два других выражения — при ложном:

```

If x > 0 Then
    y = 2
    z = 3
Else
    y = -2
    z = -3
End If

```

Конструкция Select Case

Конструкция Select Case используется для предоставления выбора из двух и более опций. Эта конструкция является альтернативой конструкции If-Then-Else. В ней используется следующий синтаксис:

```

Select Case логическое выражение
    [Case список выражений-n
        [операторы-n]]
    [Case Else
        [операторы_по_умолчанию]]
End Select

```

В следующем примере конструкции Select Case показан другой способ написания кода из примера GreetMe, представленный в предыдущем разделе:

```

Function GreetMe()
    Select Case Time
        Case Is < 0.5
            GreetMe = "Good Morning"
        Case 0.5 To 0.75
            GreetMe = "Good Afternoon"
        Case Else
            GreetMe = "Good Evening"
    End Select
End Function

```

А вот пример с функцией Discount, в этот раз с использованием конструкции Select Case:

```
Function Discount(quantity)
  Select Case quantity
    Case Is <= 5
      Discount = 0
    Case 6 To 24
      Discount = 0.1
    Case 25 To 49
      Discount = 0.15
    Case 50 To 74
      Discount = 0.2
    Case Is >=75
      Discount = 0.25
  End Select
End Function
```

Под оператором Case можно вводить любое количество операций; все они выполняются, если условие верно.

Создание цикла операторов

Циклическая обработка — это процесс повторения группы операторов VBA в процедуре. Количество повторений (итераций) можно задать самому или определить на основе значений переменных в программе. VBA предлагает несколько конструкций циклов:

- ◆ Цикл For-Next
- ◆ Цикл Do While
- ◆ Цикл Do Until

Цикл For-Next

В цикле For-Next используется следующий синтаксис:

```
For счетчик = начало To конец [Step шаг]
  [операторы]
[Exit For]
  [операторы]
Next [счетчик]
```

В приведенном ниже примере показано использование цикла For-Next, в котором нет обязательного оператора Step и выражения Exit For. Функция возвращает сумму всех целых чисел в диапазоне (включительно), заданном двумя аргументами.

```
Function SumInteger(first, last)
  total = 0
  For num = first To last
    total = total + num
  Next num
  SumInteger = total
End Function
```

А следующая строка возвращает число 55 — сумму всех целых чисел от 1 до 10.
=SumIntegers(1,10)

В этом примере `num` (счетчик переменных в цикле) начинается с первой переменной и увеличивается на 1 при каждом повторе цикла. Цикл завершается, когда `num` равно последней переменной. Переменная `total` складывает все значения `num`, которые оно принимало во время выполнения цикла.



При использовании цикла `For-Next` следует иметь в виду, что счетчик в цикле — не обычная переменная, а особый тип переменной. Поэтому значение счетчика в цикле в коде между операторами `For` и `Next` можно изменять. Однако это не очень хорошая идея — при ее реализации могут возникнуть проблемы. Будьте внимательны и следите, чтобы счетчик переменных в цикле кода не изменился.

Чтобы пропустить определенные итерации цикла, воспользуйтесь оператором `Step`. Ниже приведена функция вычисления суммы чисел *через* между первым и последним аргументами *через одно значение*.

```
Function SumIntegers2(first, last)
    total = 0
    For num = first To last Step 2
        total = total + sum
    Next num
    SumIntegers2 = Total
End Function
```

Следующая строка возвращает значение 25, которое является суммой 1, 3, 5, 7 и 9.
=`SumIntegers2(1,10)`

В цикле `For-Next` может содержаться один или несколько операторов `Exit For`. Когда встречается этот оператор, цикл тот час же завершается, как показано в следующем примере:

```
Function RowOfLargest(c)
    NumRows = Rows.Count
    MaxVal = WorksheetFunction.Max (Columns(c))
    For r = 1 To NumRows
        If Cells (r, c) = MaxVal Then
            RowOfLargest = r
        Exit If
    Next r
End Function
```

Функция `RowOfLargest` принимает номер столбца (от 1 до 256) в качестве аргумента и возвращает номер строки с наибольшим значением в этом столбце. Функция начинается с получения количества строк в таблице (значение отличается в зависимости от версии Excel). Это число присваивается переменной `NumRows`. Максимальное значение в столбце вычисляется функцией `MAX` Excel, и это значение присваивается переменной `MaxVal`.

Цикл `For-Next` проверяет каждую ячейку в столбце. Если обнаруживается ячейка, равная `MaxVal`, то номер строки (переменная `r`, счетчик в цикле) присваивается имени функции, а выражение `Exit For` завершает цикл. Если не написать выражение `Exit For`, цикл продолжит проверять все ячейки столбца, на что может потребоваться довольно много времени!

В предыдущем примере использовались сравнительно простые циклы. Но бывают ситуации, когда в цикле используется несколько сложных выражений, а иногда один цикл `For-Next` вкладывается в другой. Далее показан код VBA, в котором массив $10 \times 10 \times 10$ инициализируется значениями -1. Для этого используются вложенные циклы `For-Next`. Когда завершится выполнение трех циклов, каждому из 1000 элементов `MyArray` присваивается значение -1.

```
Dim MyArray(1 to 10, 1 to 10, 1 to 10)
For I = 1 To 10
    For j = 1 To 10
```



```

    For k = 1 To 10
        MyArray(I, j, k) = -1
    Next k
Next j
Next i

```

Цикл Do While

Другой циклической конструкцией, предлагаемой VBA, является Do While. В отличие от цикла For-Next, Do While выполняется, пока не выполнится определенное условие. В цикле Do While используется следующий синтаксис:

```

Do [While условие]
[операторы]
[Exit Do]
[операторы]
Loop

```

Или:

```

Do
[операторы]
[Exit Do]
[операторы]
Loop [While условие]

```

Как видно, в VBA можно поставить условие While в начале или в конце цикла. Разница между этими двумя синтаксисами заключается в выполнении условий. Цикл первого варианта синтаксиса может не выполниться ни разу. Во втором варианте синтаксиса условие всегда выполняется хотя бы один раз.

В следующем примере приведена функция RowOfLargest, записанная с использованием цикла Do While (первый вариант синтаксиса).

```

Function RowOfLargest(c)
    NumRows = Rows.Count
    MaxVal = Application.Max(Column(c))
    r = 1
    DoWhile Cells(r, c) <> MaxVal
        r = r + 1
    Loop
    RowOfLargest = r
End Function

```

Первым значением переменной r является 1, затем оно увеличивается в цикле Do While. Цикл продолжает выполняться, пока не встретится значение ячейки, равное MaxVal. Затем цикл прекращается и функции присваивается значение r. Обратите внимание, что если максимальное значение расположено в первой строке, то цикл выполняться не будет.

В приведенной ниже процедуре использован второй вариант синтаксиса Do While. Цикл всегда выполняется хотя бы один раз.

```

Function RowOfLargest(c)
    NumRows = Rows.Count
    MaxVal = Application.Max(Columns(c))
    r = 0
    Do
        r = r + 1
    Loop While Cells(r, c) <> MaxVal
    RowOfLargest = r
End Function

```

В цикле Do While может содержаться один или несколько операторов Exit Do. Когда встречается оператор Exit Do, цикл сразу прерывается.

Цикл Do Until

Структура цикла Do Until очень похожа на конструкцию Do While. Разница заметна только при выполнении условий. В цикле Do While операторы выполняются, *пока* условие верно. В цикле Do Until операторы выполняются *до тех пор, пока* условие не станет верным. В цикле Do Until используется такой синтаксис:

```
Do [Until условие]
    [операторы]
    [Exit Do]
    [операторы]
Loop
```

Или:

```
Do
    [операторы]
    [Exit Do]
    [операторы]
Loop [Until условие]
```

В следующем примере показано применение первого варианта синтаксиса цикла Do Until. В этом коде проще разобраться, поскольку в нем нет отрицательного сравнения, используемого в цикле Do While.

```
Function RowOfLargest(c)
    NumRows = Rows.Count
    MaxVal = Application.Max(Columns(c))
    r = 1
    Do Until (r, c) = MaxVal
        r = r + 1
    Loop
    RowOfLargest = r
End Function
```

И, наконец, эта же процедура с использованием второго варианта синтаксиса цикла Do Until.

```
Function RowOfLargest(c)
    NumRows = Rows.Count
    MaxVal = Application.Max(Column(c))
    r = 0
    Do
        r = r + 1
        Loop Until Calls(r, c) = MaxVal
    RowOfLargest = r
End Function
```

Оператор On Error

Несомненно, при использовании функций в формулах в Excel иногда случалось так, что формула возвращала значение ошибки (например, #ЗНАЧ!). Значение ошибки появляется в следующих случаях:

- ◆ Пропущено одно или несколько аргументов.
- ◆ Аргумент неправильного типа (например, текстовый вместо числового).
- ◆ Аргумент находится вне приемлемого числового диапазона (например, деление на ноль).

Во многих случаях ошибку можно просто проигнорировать. Если не указать корректное значение аргумента, то функция просто выдаст значение ошибки. Устранением ошибки занимается пользователь вручную. Так исправляются ошибки в рабочем листе Excel.

В любых случаях появившиеся ошибки нужно исправлять. Оператор `On Error` в Excel позволяет выявить и устранить их.

Чтобы просто проигнорировать ошибку, воспользуйтесь следующим выражением:

```
On Error Resume Next
```

При использовании этого выражения ошибки определяются в результате проверки свойства `Number` объекта `Err`. Если свойство равно нулю, ошибка не появляется. Если `Err.Number` равно другому значению, то появляется сообщение об ошибке.

В следующем примере написана функция, возвращающая имя ячейки или диапазона. Если у ячейки или диапазона нет имени, то появляется ошибка, а формула, в которой используется функция, возвращает #ЗНАЧ!.

```
Function RENGENAME(rng)
    RANGENAME = rng.Name.Name
End Function
```

В приведенном ниже листинге показана улучшенная версия этой же функции. Оператор `On Error Resume Next` сообщает VBA, что ошибку нужно проигнорировать. Оператор `If Err` проверяет наличие ошибок. Если они есть, то функция возвращает пустую строку.

```
Function RANGENAME(rng)
    On Error Resume Next
    RANGENAME = rng.Name.Name
    If Err.Number <> 0 Then RANGENAME = ""
End Function
```

Представленное ниже выражение предлагает VBA проверить существование ошибок, и, если ошибка найдена, продолжить выполнение с другого места; в данном случае, с оператора `ErrorHandler`.

```
On Error GoTo ErrorHandler
```

В следующей процедуре функции продемонстрировано практическое применение этого выражения. Функция `DIVIDETWO` принимает два аргумента (`num1` и `num2`) и возвращает результат — `num1`, деленный на `num2`.

```
Function DIVIDETWO(num1, num2)
    on Error GoTo ErrorHandler
    DIVIDETWO = num1 / num2
    Exit Function
ErrorHandler:
    DIVIDETWO = "ERROR"
End Function
```

В случае возникновения ошибки, выражение `On Error GoTo` предлагает VBA перейти к операциям, помеченным как `ErrorHandler`. В результате функция возвратит строку (ERROR), если при выполнении функции возникнет ошибка. Обратите внимание на оператор `Exit Function`. Без этого выражения выполнение кода продолжится, и всегда будет отображаться ошибка. Другими словами, функция всегда будет возвращать ERROR.

Важно понять, что функция DIVIDETWO нестандартная. Здесь при появлении ошибки (ERROR) в качестве результата возвращается строка, чего нет в Excel. В Excel обычно возвращается действительное значение ошибки.



В главе 25 приведены несколько примеров использования выражения On Error, в том числе пример, показывающий, что сделать, чтобы функция возвратила действительное значение ошибки.

Работа с диапазонами

Многие созданные пользовательские функции работают с данными, расположенными в ячейке или в диапазоне ячеек. Не забудьте, что диапазоном называется как одна ячейка, так и группа ячеек. В этом разделе описываются некоторые основные способы улучшенного выполнения задач. Приведенная в этом разделе информация рассчитана на практическое применение. Чтобы подробно ознакомиться с их функционированием, обратитесь к справочной системе.



В главе 25 приведено много практических примеров использования диапазонов. Изучение этих примеров поможет разобраться с информацией данного раздела.

Конструкция For Each-Next

Иногда процедуру функции нужно зациклить на определенном диапазоне ячеек. Например, можно написать функцию, которая в качестве аргумента принимала бы диапазон. В описанном случае код проверяет каждую ячейку диапазона и выполняет определенное действие. Конструкция For Each-Next очень удобна для операций такого рода. В ней используется следующий синтаксис:

```
For Each элемент In группа
    [операторы]
[Exit For]
    [операторы]
Next [элемент]
```

Нижеприведенная процедура Function принимает аргумент диапазона и возвращает сумму квадратов величин этого диапазона.

```
Function SUMOFSQUARES(rng as Range)
    Dim total as Double
    Dim cell as Range
    total = 0
    For Each cell In rng
        total = total + cell ^ 2
    Next cell
    SUMOFSQUARES = total
End Function
```

А вот формула Excel, в которой используется функция SumOfSquares:
=SumOfSquares(A1:C100)

В описанном случае аргументом функции является диапазон, состоящий из 100 ячеек.



В предыдущем примере *cell* и *rng* — имена переменных. В именах нет ничего необычного; их можно изменить на любое другое допустимое название.

Ссылка на диапазон

В коде VBA к диапазону можно обратиться несколькими способами:

- ◆ с помощью свойства *Range*
- ◆ с помощью свойства *Cells*
- ◆ с помощью свойства *Offset*

Свойство *Range*

Свойство *Range* можно использовать для прямого обращения к диапазону, используя адрес или имя ячейки. В приведенном ниже примере значение ячейки *A1* присваивается переменной, которая называется *Init*. Таким образом, организуется доступ к свойству *Value* диапазона.

```
Init = Range("A1").Value
```

Кроме *Value* VBA предлагает и другие свойства диапазона. Например, следующее выражение подсчитывает количество ячеек в диапазоне и присваивает это значение переменной *Cnt*.

```
Cnt = Range("A1:C300").Count
```

Свойство *Range* также удобно использовать для обращения к одной ячейке многоячеечного диапазона. Например, можно создать функцию, которая принимает в качестве аргумента одну ячейку. Если пользователь задает в качестве аргумента несколько ячеек, то с помощью свойства *Range* можно получить значение левой верхней ячейки диапазона. В следующем примере используется свойство *Range* (с аргументом *A1*) для получения значения левой верхней ячейки диапазона, представленного аргументом *cell*.

```
Function Square(cell as Range)  
    Dim CellValue as Double  
    CellValue = cell.Range("A1").Value  
    Square = CellValue ^ 2  
End Function
```

Предположим, что пользователь ввел следующую формулу:
`=Square(C5:C12)`

Функция *Square* обращается к левой верхней ячейке диапазона *C5:C12* (т.е. *C5*) и возвращает квадрат его значения.



Так работают многие функции Excel. Например, если в качестве первого аргумента функции *ЛЕВСИМВ* задать многоячеечный диапазон, то Excel примет левую верхнюю ячейку диапазона. Однако Excel — противоречивая программа. Если многоячеечный диапазон указать в качестве аргумента функции *SQRT*, то Excel возвратит ошибку.

Свойство *Cells*

Другим способом обращения к диапазону является свойство *Cells*. Оно принимает два аргумента (номер строки и столбца) и возвращает одну ячейку. В следующем выражении значение ячейки *A1* присваивается переменной *FirstCell*:

```
FirstCell = Cells(1,1).Value
```

А следующее выражение возвращает левую верхнюю ячейку диапазона C5 : C12:

```
UpperLeft = Range("C5:C12").Cells(1,1)
```



Если свойство Cells используется без аргумента, то оно возвращает диапазон, содержащий все ячейки рабочего листа. В приведенном ниже примере переменная TotalCells содержит общее количество ячеек рабочего листа:

```
TotalCells = Cells.Count
```

В следующем выражении используется функция Excel для определения количества непустых ячеек листа:

```
NonEmpty = WorksheetFunction.COUNTS(Cells)
```

Свойство Offset

Свойство Offset (как и свойства Range и Cells) также возвращает объект Range. Свойство Offset применяется к диапазону. Оно принимает два аргумента, соответствующие относительному расположению левой верхней ячейки определенного объекта Range. Аргументы могут быть положительными, отрицательными или равными нулю. В следующем примере возвращается значение ячейки, которая на одну ниже ячейки A1 (т.е. A2), и присваивается переменной NextCell:

```
NextCell = Range("A1").Offset(1, 0).Value
```

В следующей процедуре функции принимается аргумент одной ячейки и используется цикл For-Next для возвращения суммы 10 ячеек, находящихся под выбранной ячейкой:

```
Function SumBelow(cell as Range)
    Dim Total As Double
    Dim i as Integer
    Total = 0
    For i = 1 To 10
        Total = Total + cell.Offset(i, 0)
    Next i
    SubBelow = Total
End Function
```

Часто используемые свойства диапазона

В предыдущих разделах приводились примеры использования свойства Value диапазона. VBA предоставляет возможность работать со многими дополнительными свойствами диапазона. Некоторые наиболее полезные свойства кратко описаны в следующих разделах. Более полная информация по тому или иному свойству находится в справочной системе.

Свойство Formula

Свойство Formula возвращает формулу (как строку), которая находится в ячейке. При обращении к свойству Formula диапазона, содержащего несколько ячеек, возникает ошибка. Если в ячейке нет формулы, то свойство возвратит строку — значение ячейки в таком виде, в каком оно выглядит в строке формул. Следующая функция отображает формулу левой верхней ячейки диапазона:

```
Function CELLFORMULA(cell)
    CELLFORMULA = cell.Range("A1").Formula
End Function
```

Чтобы определить, есть ли в ячейке формула, используйте свойство HasFormula.

Свойство Address

Свойство Address возвращает адрес диапазона в виде строки. По умолчанию это свойство возвращает адрес в виде абсолютной ссылки (например, \$A\$1:\$C\$12). Следующая функция, используемая не так уж и часто, возвращает адрес диапазона.

```
Function RANGEADDRESS(rng)
    RANGEADDRESS = rng.Address
End Function
```

А следующая формула возвращает строку \$A\$1:\$C\$3:

```
=RANGEADDRESS(A1:C3)
```

Свойство Count

Свойство Count возвращает количество ячеек в диапазоне. Оно используется в следующем примере:

```
Function CELLCOUNT(rng)
    CELLCOUNT = rng.Count
End Function
```

Эта формула возвращает 9:

```
=CELLCOUNT(A1:C3)
```

Свойство Parent

Свойство Parent возвращает объект, соответствующий контейнеру текущего объекта. Для объекта Range свойство Parent возвращает объект WorkSheet (рабочий лист, содержащий диапазон).

В приведенной ниже функции используется функция Parent; она возвращает имя рабочего листа диапазона, используемого в качестве аргумента:

```
Function SHEETNAME(rng)
    SHEETNAME = rng.Parent.Name
End Function
```

Следующая формула возвращает строку Sheet1:

```
=SHEETNAME(Sheet1!A16)
```

Свойство Name

Свойство Name возвращает объект Name ячейки диапазона. Чтобы получить реальное имя ячейки или диапазона, нужно обратиться к свойству Name объекта Name. Если у ячейки или диапазона нет имени, то свойство Name возвратит ошибку.

В следующей процедуре функции определяется имя ячейки или диапазона, переданных в качестве аргумента. Если у диапазона или ячейки нет имени, функция возвращает пустую строку. Обратите внимание, что в ней используется выражение On Error Resume Next. Таким образом, учитывается ситуация, когда у диапазона нет имени.

```
Function RANGENAME(rng)
    On Error Resume Next
    RANGENAME = rng.Name.Name
    If Err.Number <> 0 Then RANGENAME = ""
End Function
```

Свойство Numberformat

Свойство Numberformat возвращает числовой формат ячейки (в виде строки), присвоенный самой ячейке или диапазону. В приведенной ниже функции показан формат номера левой верхней ячейки диапазона:

```
Function NUMBERFORMAT(cell)
    NUMBERFORMAT = cell.Range("A1").NumberFormat
End Function
```

Свойство Font

Свойство Font возвращает объект Font диапазона или ячейки. Чтобы осуществить какое-либо действие с помощью объекта Font, нужно обратиться к его свойствам. Например, у объекта Font есть такие свойства, как Bold, Italic, Name, Color и так далее. Следующая функция возвращает значение TRUE, если к левой верхней ячейке применено полужирное начертание:

```
Function ISBOLD(cell)
    ISBOLD = cell.Range("A1").Font.Bold
End Function
```

Свойства Columns и Rows

Свойства Columns и Rows используются для управления строками и столбцами диапазона. Например, следующая функция с помощью обращения к свойству Count возвращает количество столбцов в диапазоне:

```
Function COLUMNCOUNT(rng)
    COLUMNCOUNT = rng.Column.Count
End Function
```

Свойства Entirerow и Entirecolumn

Свойства Entirerow и Entirecolumn позволяют работать со всей строкой или столбцом, в которой(ом) расположена текущая ячейка. Следующая функция принимает в качестве аргумента одну ячейку и использует свойство EntireColumn, чтобы получить диапазон, содержащий целый столбец с заданной ячейкой. Далее используется функция Excel СЧЁТЗ, чтобы определить количество непустых ячеек в столбце.

```
Function NONEMPTY(cell)
    NONEMPTY = WorksheetFunction.CountA(cell.EntireColumn)
End Function
```

Свойство Hidden

Свойство Hidden используется для управления строками и столбцами. Оно возвращает значение TRUE, если строка или столбец скрыт. Если применить это свойство к диапазону, в который не входит целая строка или столбец, возникает ошибка. Приведенная ниже функция принимает аргумент одной ячейки и возвращает TRUE, если либо строка, либо столбец с указанной ячейкой скрыт:

```
Function CELLISHIDDEN(cell)
    If cell.EntireRow.Hidden Or cell.EntireColumn.Hidden
    Then
        CELLISHIDDEN = True
    Else
        CELLISHIDDEN = False
    End If
End Function
```


Эту функцию можно записать без использования конструкции If-Then-Else. В следующей функции выражение справа от знака равенства возвращает либо TRUE, либо FALSE — и это значение присваивается переменной, возвращаемой функцией.

```
Function CELLISHIDDEN(cell)
    CELLISHIDDEN = cell.EntireRow.Hidden Or _
        cell.EntireColumn.Hidden
End Function
```

Ключевое слово Set

Важным моментом VBA является возможность создавать новый объект Range и присваивать его переменной, а конкретнее, создавать *переменную объекта*. Эта задача выполняется с помощью ключевого слова Set. Следующее выражение создает переменную объекта MyRange:

```
Set MyRange = Range("A1:A10")
```

После выполнения этого выражения переменную MyRange можно использовать в коде для обращения к диапазону. Примеры следующих подразделов помогут разобраться в этой возможности.



Создание объекта Range отличается от создания именованного диапазона. Другими словами, имя объекта Range нельзя использовать в формулах.

Функция Intersect

Функция Intersect возвращает диапазон, образованный пересечением двух диапазонов. Например, возьмем два диапазона, представленных на рис. 24.2. У этих диапазонов, D3 : D10 и B5 : F5, есть одна общая ячейка (D5). Другими словами, D5 является пересечением D3 : D10 и B5 : F5.

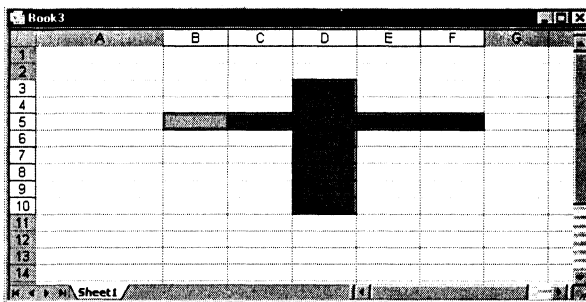


Рис. 24.2. Использование функции Intersect для управления пересечением двух диапазонов

Следующая процедура функции принимает два диапазона и подсчитывает, сколько у них общих ячеек.

```
Function CELLSINCOMMON(rng1, rng2)
    Dim CommonCells As Range
    On Error Resume Next
    Set CommonCells = Intersect(rng1, rng2)
    If Err.Number = 0 Then
```

```

CELLINCOMMON = CommonCells.Count
Else
CELLINCOMMON = 0
End If
End Function

```

Функция CELLINCOMMON использует функцию `Intersect` для создания объекта диапазона `CommonCells`. Обратите внимание на использование выражения `On Error Resume Next`. Оно необходимо, поскольку, если у диапазонов нет общих ячеек, функция `Intersect` возвращает ошибку. Если встречается ошибка и используется выражение `On Error Resume Next`, то ошибка игнорируется. Последнее выражение проверяет свойство `Number` объекта `Err`. Если оно равно 0, то ошибка не возникает, и функция возвращает значение свойства `Count` объекта `CommonCells`. Если возникает ошибка, то `Err.Number` принимает значение, отличное от нуля, и функция возвращает 0.

Функция Union

Функция `Union` объединяет два и более диапазона в один. В следующем выражении функция `Union` используется для создания объекта диапазона, содержащего первый и третий столбцы рабочего листа:

```
Set TwoCols = Union(Range("A:A"), Range("C:C"))
```

Функция `Union` может содержать любое количество аргументов.

Свойство UsedRange

Свойство `UsedRange` возвращает объект `Range`, характеризующий используемый диапазон рабочего листа. При нажатии клавиш `<Ctrl+End>` активизируется правая нижняя ячейка используемого диапазона. Свойство `UsedRange` может быть полезно при повышении эффективности выполнения функций.

Рассмотрим следующую процедуру функции. Эта функция принимает аргумент диапазона и возвращает количество ячеек с формулами в диапазоне.

```

Function FORMULACOUNT(rng As Range)
Dim cnt As Long
Dim cell As Range
cnt = 0
For Each cell In rng
If cell.HasFormula Then cnt = cnt + 1
Next cell
FORMULACOUNT = cnt
End Function

```

В большинстве случаев эта функция прекрасно работает. Но что если пользователь введет следующую формулу:

```
=FORMULACOUNT("A:C")
```

С аргументом, содержащим один или более столбцов, эта функция не работает, поскольку она в цикле просматривает каждую ячейку диапазона, даже те ячейки, которые лежат вне области использованного листа. Ниже приведена функция, исправляющая рассмотренный недостаток предыдущей функции:

```

Function FORMULACOUNT(rng As Range)
Dim cnt As Long
Dim cell As Range

```

```
cnt = 0
set WorkRange = Intersect(rng, rng.Parent.UsedRange)
For Each cell In WorkRange
    If cell.HasFormula Then cnt = cnt + 1
Next cell
FORMULACOUNT = cnt
End Function
```

Эта функция создает объект Range с именем WorkRange, в котором сохраняется пересечение диапазона, принимаемого в качестве аргумента, и используемого диапазона рабочего листа. Другими словами, WorkRange состоит из подмножества аргументов диапазона, в которое входят ячейки используемого диапазона рабочего листа.

Резюме

В этой главе вы познакомились с элементами языка VBA, а именно с переменными, типами данных, константами и массивами. Здесь также рассматривались различные методы регулирования выполнения процедур функций. Было представлено несколько примеров функций, которые продемонстрировали управление массивами и использование встроенных функций VBA.

В следующей, последней главе приведены примеры пользовательских функций.

Примеры пользовательских функций VBA

В этой главе...

- ◆ Простые функции
- ◆ Определение типа данных в ячейке
- ◆ Многофункциональная функция
- ◆ Генерирование случайных чисел
- ◆ Вычисление комиссионных от продаж
- ◆ Функции управления текстом
- ◆ Функции подсчета и суммирования
- ◆ Функции управления датами
- ◆ Возвращение последней непустой ячейки столбца или строки
- ◆ Функции для работы в нескольких листах
- ◆ Дополнительные пользовательские функции
- ◆ Резюме

В этой главе приведены различные примеры полезных (или потенциально полезных) пользовательских функций VBA. Большинство функций можно использовать в таком виде, в каком они приведены. Но некоторые функции нужно немного изменить, чтобы они отвечали конечным задачам. Для повышения производительности и эффективности выполнения таких процедур функций в этой главе описываются все используемые в них переменные.

Простые функции

Относительно простые функции, представленные в этой главе, могут оказаться очень полезными. Многие из них основаны на том факте, что VBA получает доступ к большому количеству полезной информации, которая обычно не используется в формулах. Например, код VBA может обратиться к свойству ячейки `HasFormula`, чтобы определить, содержится ли в ячейке формула. Странно, но в Excel нет такой встроенной функции.

Есть ли в ячейке формула?

Приведенная ниже функция `CELLHASFOMULA` принимает аргумент в виде одной ячейки и возвращает значение `TRUE`, если в ячейке содержится формула.

```
Function CELLHASFOMULA(cell) As Boolean
' Возвращает TRUE если в ячейке введена формула
CELLHASFOMULA = cell.Range("A1").HasFormula
End Function
```

Если в функции использовать многоячеечный диапазон, то она проверит значение только в левой верхней ячейке диапазона.

Возвращение формулы ячейки

Следующая функция CELLFORMULA возвращает формулу, содержащуюся в ячейке, в виде строки. Если в ячейке нет формулы, функция возвращает пустую строку.

```
Function CELLFORMULA(cell) As String
' Возвращает формулу ячейки или
' пустую строку, если формулы в ячейке нет
Dim UpperLeft As Range
Set UpperLeft = cell.Range("A1")
If UpperLeft.HasFormula Then
    CELLFORMULA = UpperLeft.Formula
Else
    CELLFORMULA = ""
End If
End Function
```

Эта функция создает переменную объекта Range, которая называется UpperLeft. Эта переменная представляет левую верхнюю ячейку аргумента функции.

Скрыта ли ячейка?

Следующая функция CELLISHIDDEN принимает аргумент в виде одной ячейки и возвращает значение TRUE, если ячейка скрыта. Ячейка считается скрытой, если скрыта ее строка или столбец.

Использование функций этой главы

Если в этой главе вам встретится функция, полезная для решения конкретных задач, то используйте ее в своем рабочем листе. Все процедуры этой главы приведены на Web-узле. Откройте соответствующую рабочую книгу (см. Приложение Д с описанием файлов), активизируйте редактор Visual Basic и вставьте листинг функции в модуль VBA вашей рабочей книги. При необходимости воспользуйтесь несколькими функциями или создайте надстройку (см. главу 23).

Невозможно предугадать, какая функция понадобится в том или ином случае. Однако примеры этой главы охватывают различные темы, что поможет определиться с выбором функции и применить ее в собственном коде.

```
Function CELLISHIDDEN(cell) As Boolean
' Возвращает TRUE если ячейка скрыта
Dim UpperLeft As Range
Set UpperLeft = cell.Range("A1")
CELLISHIDDEN = UpperLeft.EntireRow.Hidden Or _
    UpperLeft.EntireColumn.Hidden
End Function
```

Возвращение имени рабочего листа

Приведенная ниже функция SHEETNAME принимает один аргумент (диапазон) и возвращает имя рабочего листа с этим диапазоном. В функции используется свойство Parent объекта Range. Свойство Parent возвращает объект, в котором содержится объект Range.

```
Function SHEETNAME(rng) As String
' Возвращает имя рабочей книги для rng
```

```

    SHEETNAME = rng.Parent.Name
End Function

```

А вот еще один вариант этой же функции. В ней не используется аргумент; действие этой функции основано на том, что она может определить ячейку, из которой она вызвана с помощью свойства `Application.Caller`.

```

Function SHEETNAME2() As String
    ' Возвращает имя книги для ячейки,
    ' содержащей функцию
    SHEETNAME2 = Application.Caller.Parent.Name
End Function

```

Понятие родительских объектов

Объекты в Excel выстроены в иерархическом порядке. На верхнем уровне иерархии расположен объект `Application` (сама Excel). В Excel есть и другие объекты; эти объекты, в свою очередь, состоят из других объектов и т.д. Ниже показано, на каком месте в этой иерархии находится объект `Range`.

```

Объект Application (Excel)
    Объект Workbook
        Объект Worksheet
            Объект Range

```

В объектно-ориентированном программировании родительским для объекта `Range` является объект `Worksheet`, в котором находится объект `Range`. Родительским для объекта `Worksheet` является `Workbook`. И, наконец, родителем объекта `Workbook` является объект `Application`. Разобравшись в такой структуре, используйте свойство `Parent`, чтобы создавать различные полезные функции.

В этой функции свойство `Application.Caller` возвращает объект `Range`, который соответствует ячейке с функцией. Предположим, что в ячейке `A1` содержится такая формула: `=SHEETNAME()`

Когда выполняется функция `SHEETNAME`, свойство `Application.Caller` возвращает объект `Range`, соответствующий ячейке, содержащей функцию. Свойство `Parent` возвращает объект `Worksheet`, а свойство `Name` возвращает имя рабочего листа.

Возвращение имени рабочей книги

Приведенная ниже функция `WORKBOOKNAME` возвращает имя рабочей книги. Обратите внимание, что в этой функции свойство `Parent` используется дважды. Первый раз свойство `Parent` возвращает объект `Worksheet`, а во второй раз — объект `Workbook`; свойство `Name` возвращает имя рабочей книги.

```

Function WORKBOOKNAME() As String
    ' Возвращает имя книги для ячейки,
    ' содержащей функцию
    WORKBOOKNAME = Application.Caller.Parent.Parent.Name
End Function

```

Возвращение имени приложения

Следующая функция, в принципе, редко используемая, переносит рассуждения о родительских объектах на следующий логический уровень, обращаясь к свойству `Parent` три раза. Эта функция возвращает имя объекта `Application`, которое всегда является строкой *Microsoft Excel*.

```
Function APPNAME() As String
'   Возвращает имя книги для ячейки,
'   содержащей функцию
    APPNAME = Application.Caller.Parent.Parent.Parent.Name
End Function
```

Возвращение версии Excel

Следующая функция возвращает номер версии Excel. Например, если используется Excel 2002, то она возвратит текстовую строку 10.0.

```
Function EXCELVERSION() As String
'   Возвращает номер версии Excel
    EXCELVERSION = Application.Version
End Function
```

Обратите внимание, что функция `EXCELVERSION` возвращает строку, а не значение. Приведенная ниже функция возвращает значение `TRUE`, если используемым приложением является Excel версии 97 или выше (Excel 97 соответствует версии 8). В этой функции используется функция `Val` для преобразования текстовой строки в значение.

```
Function EXCEL97ORLATER() As Boolean
    EXCEL97ORLATER = Val(Application.Version) >= 8
End Function
```

Возвращение информации о форматировании ячейки

В этом разделе приведено несколько пользовательских функций, которые возвращают информацию о форматировании, примененном в ячейке. Такие функции используются при сортировке данных по форматированию (например, по ячейкам с полужирным начертанием).



В функциях этого раздела используется следующее выражение:

```
Application.Volatile True
```

Это выражение заставляет функцию проводить перерасчет формул в рабочей книге. Но, как обнаружится позднее, эти функции не всегда возвращают правильные значения. Это происходит из-за того, что Excel, скажем, не выполняет перерасчет ячеек при изменении форматирования ячеек. Чтобы все же выполнить общий перерасчет (и обновить все пользовательские функции), нажмите <Ctrl+Alt+F9>.

Следующая функция возвращает значение `TRUE`, если к ячейке-аргументу применено полужирное начертание.

```
Function ISBOLD(cell) As Boolean
'   Возвращает TRUE, если шрифт ячейки имеет полужирное начертание
    ISBOLD = cell.Range("A1").Font.Bold
End Function
```

А эта функция возвращает значение TRUE, если к аргументу одной ячейки применено курсивное начертание.

```
Function ISITALIC(cell) As Boolean
' Возвращает TRUE, если шрифт ячейки имеет курсивное начертание
  ISITALIC = cell.Range("A1").Font.Italic
End Function
```

В обоих приведенных выше функциях есть небольшое упущение: если к ячейке применено смешанное форматирование, то функции возвращают ошибку. Например, если полужирным начертанием форматированы только некоторые символы. Следующая функция возвратит значение TRUE только в том случае, если все символы в ячейке имеют полужирное начертание. В ней используется VBA-функция IsNull, определяющая, возвращает ли свойство Bold объекта Font значение Null. Если да, то в ячейке содержится смешанное форматирование.

```
Function ALLBOLD(cell) As Boolean
' Возвращает TRUE, если весь текст в ячейке
' имеет полужирное начертание
  Dim UpperLeft As Range
  Set UpperLeft = cell.Range("A1")
  ALLBOLD = False
  If UpperLeft.Font.Bold Then ALLBOLD = True
End Function
```

Следующая функция FILLCOLOR возвращает целочисленное значение, которое соответствует индексу цвета заливки ячейки. Если в ячейке нет заливки, функция возвращает значение -4142.

```
Function FILLCOLOR(cell) As Integer
' Возвращает целое число, соответствующее
' цвету заливки ячейки
  FILLCOLOR = cell.Range("A1").Interior.ColorIndex
End Function
```

Приведенная ниже функция возвращает в виде строки формат цифр в ячейке.

```
Function NUMBERFORMAT(cell) As String
' Возвращает строку, представляющую
' числовой формат ячейки
  Application.Volatile True
  NUMBERFORMAT = cell.Range("A1").NUMBERFORMAT
End Function
```

Если в ячейке используется формат цифр, заданный по умолчанию, то функция возвратит строку *General*.

Определение типа данных в ячейке

В Excel предложено несколько встроенных функций определения типа данных, содержащихся в ячейке. Эти функции называются ЕТЕКСТ, ЕЛОГИЧ и ЕОШИБКА. Кроме этих функций в VBA есть такие функции, как ЕПУСТО, ЕДАТА и ЕЧИСЛО.

Следующая функция принимает аргумент диапазона и возвращает строку (*Blank, Text, Logical, Error, Date, Time* или *Value*), которая описывает тип данных левой верхней ячейки диапазона.

```
Function CELLTYPE(cell)
' Возвращает тип левой верхней
' ячейки диапазона
  Dim UpperLeft As Range
  Application.Volatile
```



```

Set UpperLeft = cell.Range("A1")
Select Case True
    Case UpperLeft.NumberFormat = "@"
        CELLTYPЕ = "Text"
    Case IsEmpty(UpperLeft)
        CELLTYPЕ = "Blank"
    Case WorksheetFunction.IsText(UpperLeft)
        CELLTYPЕ = "Text"
    Case WorksheetFunction.IsLogical(UpperLeft)
        CELLTYPЕ = "Logical"
    Case WorksheetFunction.IsErr(UpperLeft)
        CELLTYPЕ = "Error"
    Case IsDate(UpperLeft)
        CELLTYPЕ = "Date"
    Case InStr(1, UpperLeft.Text, ":") <> 0
        CELLTYPЕ = "Time"
    Case IsNumeric(UpperLeft)
        CELLTYPЕ = "Value"
End Select
End Function

```

На рис. 25.1 показана функция CELLTYPЕ в действии. В столбце В содержатся формулы, обработанные функцией CELLTYPЕ с аргументами из столбца А. Например, в ячейке В1 содержится такая формула:

=CELLTYPЕ(A1)

	A	B	C
1	1	Value	A simple value
2	8,6	Value	Formula that returns a value
3	Budget Sheet	Text	Simple text
4	ЛОЖЬ	Logical	Logical formula
5	ИСТИНА	Logical	Logical value
6	#ДЕЛ/0!	Error	Formula error
7	07.06.2002	Date	Formula that returns a date
8	4:00 PM	Time	A time
9	'143	Text	Value preceded by apostrophe
10	434	Text	Cell formatted as Text
11	A1:C4	Text	Text with a colon
12		Blank	Empty cell
13		Text	Cell with a single space
14			
15			
16			

Рис. 25.1. Функция CELLTYPЕ возвращает строку с описанием типа содержимого ячейки

Многофункциональная функция

В этом разделе показаны способы создания одной функции рабочего листа, которая работает как несколько функций. Такая функция VBA, называемая STATFUNCTION, работает с двумя аргументами — диапазоном (rng) и оператором (op). В зависимости от значения op функция возвращает значение, вычисленное с использованием одной из функций рабочего листа: СРЗНАЧ, СЧЁТ, МАКС, МЕДИАНА, МИН, МОДА и т.д. Эту функцию можно использовать на рабочем листе:

=STATFUNCTION(B1:B24;A24)

Результат формулы зависит от содержимого ячейки A24, которая содержит строку, например, Average, Count, Max и т.д. Таким образом можно работать и с другими типами функций.

```

Function STATFUNCTION(rng, op)
    Select Case UCase(op)
        Case "SUM"
            STATFUNCTION = Application.Sum(rng)
        Case "AVERAGE"
            STATFUNCTION = Application.Average(rng)
        Case "MEDIAN"
            STATFUNCTION = Application.Median(rng)
        Case "MODE"
            STATFUNCTION = Application.Mode(rng)
        Case "COUNT"
            STATFUNCTION = Application.Count(rng)
        Case "MAX"
            STATFUNCTION = Application.Max(rng)
        Case "MIN"
            STATFUNCTION = Application.Min(rng)
        Case "VAR"
            STATFUNCTION = Application.Var(rng)
        Case "STDEV"
            STATFUNCTION = Application.StDev(rng)
        Case Else
            STATFUNCTION = CVErr(xlErrNA)
    End Select
End Function

```

На рис. 25.2 представлена функция STATFUNCTION, используемая вместе с раскрывающимся списком, созданным с помощью команды Excel Данные⇒Проверка. Формула находится в ячейке C14:

```
=STATFUNCTION(C1:C12;B14)
```

Следующая функция STATFUNCTION2 имеет более простой вид, но работает так же, как и функция STATFUNCTION. В этой функции для вычисления выражения используется оператор Evaluate.

```

Function STATFUNCTION2(rng, op)
    STATFUNCTION2 = Evaluate(op & "(" & _
        rng.Address(external:=True) & ")")
End Function

```

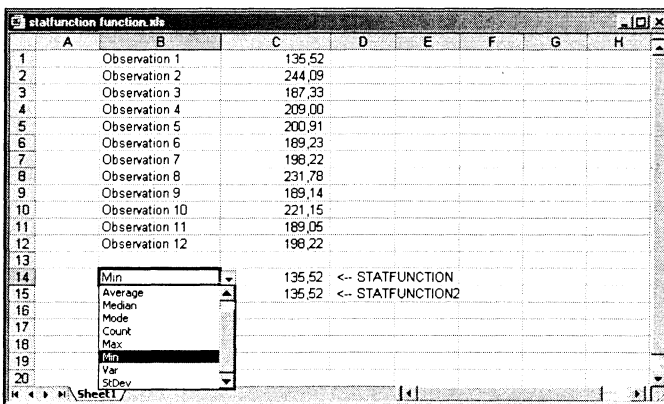


Рис. 25.2. Результат выбора операции из списка отображается в ячейке B14

Предположим, что аргумент `rng` соответствует `C1:C12`, а аргумент `op` является строковым значением `SUM`. Выражение, используемое в качестве аргумента оператора `Evaluate`, имеет следующий вид:

```
SUM(C1:C12)
```

Оператор `Evaluate` подсчитывает аргументы и возвращает результат. Кроме того, что это довольно короткое выражение, преимущество этой версии функции `STATFUNCTION` заключается еще и в том, что необязательно перечислять все возможные функции.

Генерирование случайных чисел

В этом разделе представлены две функции, работающие со случайными числами. Одна из них генерирует случайные числа, которые не изменяются. Другая выбирает из диапазона случайную ячейку.

Генерирование неизменяемых случайных чисел

Чтобы быстро создать диапазон ячеек, состоящих из случайных чисел, воспользуйтесь в Excel функцией `СЛЧИС`. Но, как выяснится позднее, функция `СЛЧИС` генерирует новое случайное число каждый раз при пересчете рабочего листа. Если нужно создать диапазон случайных чисел, которые не изменяются при каждом пересчете, используйте процедуру `STATICRAND`:

```
Function STATICRAND()  
' Возвращает произвольное число, не  
' изменяющееся при перерасчете данных  
    STATICRAND = Rnd  
End Function
```

В функции `STATICRAND` используется функция `Rnd` VBA, которая, как и функция `СЛЧИС`, возвращает случайное число от 0 до 1. Но при использовании `STATICRAND` случайные числа не меняются при пересчете листа.



При нажатии клавиши `<F9>` новые значения функции `STATICRAND` не генерируются, это можно сделать с помощью сочетания клавиш `<Ctrl+Alt+F9>` (клавиатурное сокращение “глобального пересчета” в Excel).

Чтобы сгенерировать серии случайных целых чисел между 1 и 1000, используйте формулу `=ЦЕЛОЕ(STATICRAND()*1000)+1`

Выбор случайной ячейки

Следующая функция, которая называется `DRAWONE`, выбирает случайным образом одну ячейку из введенного диапазона и возвращает ее содержимое.

```
Function DRAWONE(rng)  
' Выбор одного из чисел диапазона произвольным образом  
    DRAWONE = rng(Int((rng.Count) * Rnd + 1))  
End Function
```

Используя эту функцию, обратите внимание, что при пересчете рабочего листа она не пересчитывается. Другими словами, эта функция не является временной (более подробно об управлении процессом пересчета рассказывается во врезке “Управление пересчетом функции” далее в этой главе). Функцию можно сделать изменяемой, добавив следующее выражение:

```
Application.Volatile True
```

Управление пересчетом функции

Когда пересчитывается пользовательская функция, используемая в формуле рабочего листа?

Пользовательские функции ведут себя как встроенные функции рабочего листа Excel. Обычно пользовательская функция пересчитывается только тогда, когда она должна быть пересчитана, т.е. когда изменяются аргументы функции. Но пересчет функции можно делать чаще. Добавив следующее выражение в процедуру, в функции выполнится пересчет при изменении любой ячейки.

```
Application.Volatile True
```

Оператор `Volatile` в объекте `Application` принимает один аргумент (либо `True`, либо `False`). Атрибут `volatile` у функции заставляет ее пересчитываться каждый раз при пересчете любой ячейки рабочего листа.

Например, пользовательскую функцию `STATICRAND`, представленную в этой главе, можно заменить на ее эмулятор, функцию `Excel СЛЧИС()`, используя оператор `Volatile`:

```
Function NONSTATICRAND()  
' Возвращает произвольное число, не  
' изменяющегося при перерасчете  
Application.Volatile True  
NONSTATINCRAND = Rnd  
End Function
```

Использование в операторе `Volatile` аргумента `False` заставляет функцию выполнить пересчет только в том случае, если изменяются один или несколько ее аргументов (если в функции нет аргументов, то этот метод неэффективен). По умолчанию все функции действуют так, как будто для них задано выражение `Application.Volatile False`.

После этого функция `DRAWONE` будет отображать новые случайные значения каждый раз при пересчете листа.



Далее в этой главе представлены еще две функции управления случайными числами.

Вычисление комиссионных от продаж

Менеджерам по продажам часто необходимо просчитать комиссионные, полученные от продаж. Расчеты в примере функции, представленной ниже, основаны на подвижной шкале: чем больше работник продал, тем больше комиссионных он получил (табл. 25.1). Например, менеджер с продажами в диапазоне от \$10000 до \$19999 получает 10,5% комиссионных.

Таблица 25.1. Ставка комиссионных для продаж за месяц

Продажа за месяц	Процентная ставка
Менее \$10000	8,0%
\$10000-\$19999	10,5%
\$20000-\$39999	12,0%
\$40000 и более	14,0%

Комиссионные можно вычислить для различных объемов продаж, указанных на рабочем листе. Можно использовать сложную формулу со вложенными функциями IF, например:

```
=IF (A1<0;0; IF (A1<10000; A1*0,08; IF (A1<20000; A1*0,105; IF (A1<40000; A1*0,12; A1*0,14))) )
```

Но это не наилучший способ по нескольким причинам. Во-первых, формула получается слишком громоздкой и запутанной, что затрудняет ее понимание. Во-вторых, значения в формуле указываются жестко, что затрудняет внесение изменений в формулу. И если есть больше семи процентных ставок, то запись функции выйдет за ограничения по вложенным функциям Excel.

Лучший способ описать эти условия — воспользоваться функцией таблицы Lookup для подсчета комиссионных, а именно:

```
=ПРОСМОТР (A1; Table; 2) *A1
```

Использование функции ПРОСМОТР является хорошей альтернативой, но она не выполняется, если структура начисления комиссионных слишком сложная. (См. подраздел “Функция сложной структуры начисления комиссионных”). Еще одной альтернативой является создание пользовательской функции.

Функция простой структуры начисления КОМИССИОННЫХ

Следующая функция COMMISSION принимает один аргумент (Sales) и вычисляет ставку комиссионных.

```
Function COMMISSION(Sales) As Single
    ' Вычисляет комиссионные с продаж
    Const Tier1 As Double = 0.08
    Const Tier2 As Double = 0.105
    Const Tier3 As Double = 0.12
    Const Tier4 As Double = 0.14
    Select Case Sales
        Case Is >= 40000
            COMMISSION2 = Sales * Tier4
        Case Is >= 20000
            COMMISSION2 = Sales * Tier3
        Case Is >= 10000
            COMMISSION2 = Sales * Tier2
        Case Is < 10000
            COMMISSION2 = Sales * Tier1
    End Select
End Function
```

Приведенная ниже формула рабочего листа возвращает значение 3000 (объем продаж 25000 соответствует ставке 12%):

```
=COMMISSION(25000)
```

В этой функции легко разобраться. В ней используется константа для хранения ставки комиссионных и структура Select Case для определения того, какую ставку следует брать.



Когда структура Select Case пересчитывается, программа выходит из нее, как только выполнится необходимый оператор Case.

Функция сложной структуры начисления КОМИССИОННЫХ

Если структура начисления комиссионных более сложная, следует воспользоваться дополнительными аргументами в функции COMMISSION. Представьте, что вышеупомянутый менеджер по продажам внедряет новую политику, чтобы уменьшить текучесть кадров: общие комиссионные повышаются на 1% в год, пока агент по продажам работает на эту компанию.

Ниже приведена измененная функция COMMISSION (она называется COMMISSION2). Теперь эта функция принимает два аргумента: объем продажи за месяц (Sales) и количество отработанных в компании лет (Years).

```
Function COMMISSION2(Sales, Years) As Single
' Вычисляет комиссионные с продаж на основе
' рабочего стажа
Const Tier1 As Double = 0.08
Const Tier2 As Double = 0.105
Const Tier3 As Double = 0.12
Const Tier4 As Double = 0.14
Select Case Sales
Case Is >= 40000
COMMISSION2 = Sales * Tier4
Case Is >= 20000
COMMISSION2 = Sales * Tier3
Case Is >= 10000
COMMISSION2 = Sales * Tier2
Case Is < 10000
COMMISSION2 = Sales * Tier1
End Select
COMMISSION2 = COMMISSION2 + (COMMISSION2 * Years / 100)
End Function
```

На рис. 25.3 представлена функция COMMISSION2. Формула в ячейке D2 такая: =COMMISSION2(B2;C2)

1	A	B	C	D	E
	Sales Rep	Amount Sold	Years Employed	Commission	
2	Adams, Robert	5 010,54	1	404,85	
3	Baker, Sheila	9 833,91	0	786,71	
4	Clarke, Edward	12 500,32	2	1 338,78	
5	Davis, Don	35 988,22	3	4 448,14	
6	Elfin, Bill	41 822,99	3	6 030,88	
7	Franklin, Ben	8 090,32	1	653,70	
8	Gomez, Chris	11 098,32	2	1 188,63	
9	Harley, Mary	48 745,23	5	7 165,55	
10					
11					
12					
13					

Рис. 25.3. Пересчет комиссионных на основе объема продаж и количества отработанных в компании лет

Функции управления текстом

Управление текстом с помощью функций выполняется самыми разными способами. В этом разделе мы рассмотрим операции обратного порядка следования символов (реверсирования строки), перетасовки и извлечения символов из строки. В примерах этого раздела вы найдете множество функций управления текстовыми строками.

Реверсирование строки

Следующая функция REVERSETEXT возвращает текст ячейки в обратном порядке.

```
Function REVERSETEXT(text) As String
'   Возвращает аргумент в обратном порядке
    REVERSETEXT = StrReverse(text)
End Function
```

В этой функции используется функция VBA StrReverse. Приведенная ниже формула возвращает *tfosorciM*.

```
=REVERSETEXT("Microsoft")
```

Функции StrReverse нет в Excel до версии 2000. Таким образом, если эта функция вам нужна в ранних версиях Excel, необходимо написать собственную аналогичную функцию. Функция REVERSETEXT2 работает так же, как и функция REVERSETEXT.

```
Function REVERSETEXT2(text) As String
'   Возвращает аргумент в обратном порядке
'   Используется в версиях до Excel 2000
    Dim TextLen As Integer
    Dim i As Integer
    TextLen = Len(text)
    For i = TextLen To 1 Step -1
        REVERSETEXT2 = REVERSETEXT2 & Mid(text, i, 1)
    Next i
End Function
```

В функции используется цикл For-Next с отрицательным значением Step. Буквы объединены (с помощью символа &, который является оператором конкатенации) для формирования строки в обратном порядке.

Перетасовка текста

Следующая функция возвращает содержимое аргумента, но его символы расположены в случайном порядке. Например, если использовать *Microsoft* в качестве аргумента, то в результате может получиться *oficMicro* или строка с другим порядком представления символов.

```
Function SCRAMBLE(text)
    Dim TextLen As Integer
    Dim i As Integer
    Dim RandPos As Integer
    Dim Char As String * 1
    Set text = text.Range("A1")
    TextLen = Len(text)
    For i = 1 To TextLen
        Char = Mid(text, i, 1)
        RandPos = Int((TextLen - 1 + 1) * Rnd + 1)
        Mid(text, i, 1) = Mid(text, RandPos, 1)
        Mid(text, RandPos, 1) = Char
    Next i
    SCRAMBLE = text
End Function
```

Эта функция обрабатывает каждый символ, а затем заменяет его другим случайно выбранным символом.

Действие функции Mid может показаться немного странным. Обратите внимание, что код Mid стоит справа от оператора присвоения, то это функция. А когда Mid стоит слева от оператора присвоения, то это выражение. Обратитесь к справочной системе, чтобы подробно ознакомиться с назначением Mid.

Возвращение аббревиатуры

Функция ACRONYM возвращает первую букву каждого слова аргумента (в верхнем регистре). Например, следующая формула возвратит *IBM*:

```
=ACRONYM(International Business Machines)
```

Листинг процедуры Function ACRONYM следующий:

```
Function ACRONYM(text) As String
    ' Возвращает аббревиатуру
    Dim TextLen As Integer
    Dim i As Integer
    text = Application.Trim(text)
    TextLen = Len(text)
    ACRONYM = Left(text, 1)
    For i = 2 To TextLen
        If Mid(text, i, 1) = Chr(32) Then
            ACRONYM = ACRONYM & Mid(text, i + 1, 1)
        End If
    Next i
    ACRONYM = UCase(ACRONYM)
End Function
```

В этой функции используется функция TRIM, чтобы удалить лишние пробелы в аргументе. Первый символ аргумента всегда является первым символом результата. Цикл For-Next проверяет каждый символ. Если символ является пробелом, то символ *после* пробела добавляется к результату. Затем результат преобразуется в верхний регистр с помощью функции VBA UCase.

Соответствие текста шаблону?

Следующая функция возвращает TRUE, если строка совпадает с шаблоном, составленным из текста и специальных символов. Функция ISLIKE довольно проста и обычно используется вместе с оператором VBA Like.

```
Function ISLIKE(text As String, pattern As String) As Boolean
    ' Возвращает TRUE, если первый аргумент подобен второму
    If text Like pattern Then ISLIKE = True Else ISLIKE = False
End Function
```

В VBA поддерживаются следующие специальные символы:

- ? один любой символ
- * ноль или несколько символов
- # любая цифра (0-9)
- [list] один любой символ в списке
- [!list] один любой символ вне списка

Следующая формула возвращает значение TRUE, поскольку вопросительный знак (?) соответствует любому символу. Если бы первый аргумент равнялся Unit12, то функция в результате возвратила бы FALSE.

```
=ISLIKE ("Unit1"; "Unit?")
```


Функция ISLIKE работает также и со значениями. Например, нижеприведенная формула возвращает значение TRUE, если ячейка A1 содержит значение, которое начинается с 1 и состоит из трех цифр.

```
=ISLIKE(A1;"1##")
```

Следующая формула возвращает значение TRUE, поскольку первый аргумент является символом, который находится в списке символов, определенном во втором аргументе.

```
=ISLIKE("a";"[aeiou]")
```

Если список символов начинается с восклицательного знака (!), то сравниваются символы вне списка. Например, следующая формула возвращает значение TRUE, поскольку первый аргумент является символом вне списка второго аргумента.

```
=ISLIKE("g";"[!aeiou]")
```

Использование оператора Like очень разносторонне. Более подробная информация о работе с оператором Like приведена в справочной системе.

Содержится ли в ячейке текст?

В главе 5 рассказывалось, как некоторые функции рабочего листа Excel становились временно ненадежными при управлении текстом в ячейках. Приведенная ниже функция CELLHASTEXT возвращает значение TRUE, если в аргументе ячейки содержится текст или значение, форматированное как Text.

```
Function CELLHASTEXT(cell) As Boolean
'   Возвращает TRUE, если ячейка содержит строку
'   или имеет текстовый тип данных
  Dim UpperLeft As Range
  CELLHASTEXT = False
  Set UpperLeft = cell.Range("A1")
  If UpperLeft.NumberFormat = "@" Then
    CELLHASTEXT = True
    Exit Function
  End If
  If Not IsNumeric(UpperLeft) Then
    CELLHASTEXT = True
    Exit Function
  End If
End Function
```

Следующая формула возвращает значение TRUE, если в ячейке A1 содержится строка или если ячейка отформатирована как Text.

```
=CELLHASTEXT(A1)
```

Извлечение n-ого элемента строки

Функция EXTRACTELEMENT является пользовательской функцией рабочего листа, которая извлекает элемент из текстовой строки, основываясь на определенном символе-разделителе. Предположим, что в ячейке A1 содержится такой текст:

```
123-456-789-9133-8844
```

Например, следующая формула возвращает строку 9133, что соответствует четвертому элементу строки. В строке в качестве разделителя используется дефис (-).

```
=EXTRACTELEMENT(A1;4;"-")
```

В функции `EXTRACTELEMENT` используется три аргумента:

- ◆ `Txt`. Строка текста, из которой извлекается элемент. Это может быть строка из букв или ссылка на ячейку.
- ◆ `n`. Целое число, показывающее номер извлекаемого элемента.
- ◆ `Separator`. Символ, используемый в качестве разделителя.



Если в качестве символа-разделителя определить пробел, то несколько пробелов могут восприниматься как один пробел (почти всегда это так, как и должно быть). Если `n` превышает количество элементов в строке, то функция возвращает пустую строку.

В VBA-коде функция `EXTRACTELEMENT` выглядит так:

```
Function EXTRACTELEMENT(Txt, n, Separator) As String
' Возвращает n-й элемент текстовой строки при разделении
' ее определенным символом
Dim AllElements As Variant
AllElements = Split(Txt, Separator)
EXTRACTELEMENT = AllElements(n - 1)
End Function
```

В этой функции использовалась функция VBA `Split`, которая возвращает массив `variant`, содержащий все элементы текстовой строки. Массив начинается с 0 (а не с 1), поэтому для обращения к нужному элементу следует писать `n-1`.

Функция `Split` начала использоваться с Excel 2000. Если вами используется более ранняя версия Excel, то следует создать следующую функцию:

```
Function EXTRACTELEMENT2(Txt, n, Separator) As String
' Возвращает n-й элемент текстовой строки при разделении
' ее определенным символом
' Используется в версиях до Excel 2000

Dim Txt1 As String, TempElement As String
Dim ElementCount As Integer, i As Integer

Txt1 = Txt
' Если разделитель – пробел, удаляются повторяющиеся символы
If Separator = Chr(32) Then Txt1 = Application.Trim(Txt1)

' Добавление разделителя в конец строки
If Right(Txt1, Len(Txt1)) <> Separator Then _
    Txt1 = Txt1 & Separator

' Инициализация
ElementCount = 0
TempElement = ""

' Выделение каждого элемента
For i = 1 To Len(Txt1)
    If Mid(Txt1, i, 1) = Separator Then
        ElementCount = ElementCount + 1
        If ElementCount = n Then
            ' При нахождении выход
            EXTRACTELEMENT2 = TempElement
            Exit Function
        End If
    End If
Next i

Else
```

```

TempElement = ""
End If
Else
TempElement = TempElement & Mid(Txt1, i, 1)
End If
Next i
EXTRACTELEMENT2 = ""
End Function

```

Написание цифры прописью

Функция SPELLDOLLARS возвращает найденную в тексте цифру, написанную прописью — как это делается на банковских чеках. Например, приведенная ниже формула возвращает строку *One hundred twenty-three and 45/100 dollars*.

=SPELLDOLLARS (123 , 45)

На рис. 25.4 показаны примеры функции SPELLDOLLARS. В столбце С содержатся формулы, в которых используется эта функция. Например, формула в ячейке С1 такая:

=SPELLDOLLARS (A1)

Обратите внимание, что отрицательные значения записываются прописью и заключаются в круглые скобки.

	A	B	C
1		32	Thirty-Two and 00/100 Dollars
2		37,56	Thirty-Seven and 56/100 Dollars
3		-32	(Thirty-Two and 00/100 Dollars)
4		-26,44	(Twenty-Six and 44/100 Dollars)
5		-4	(Four and 00/100 Dollars)
6		1,56	One and 56/100 Dollars
7		1	One and 00/100 Dollars
8		6,56	Six and 56/100 Dollars
9		12,12	Twelve and 12/100 Dollars
10		1000000	One Million and 00/100 Dollars
11		1000000000	Ten Billion and 00/100 Dollars
12			
13			
14			
15			

Рис. 25.4. Примеры функции SPELLDOLLARS

Функции подсчета и суммирования

В главе 7 приведено много примеров формул подсчета и суммирования ячеек по различным критериям. Если ни одна из формул для подсчета и суммирования не подходит, значит нужно создать собственную пользовательскую функцию. В этом разделе предлагаются три функции, выполняющие операции подсчета и суммирования.

Подсчет ячеек с определенными значениями

Предположим, что нужно посчитать количество значений в промежутке между 6 и 12 в диапазоне A1:A100. Для этого воспользуемся формулой

=СЧЁТЕСЛИ(A1:A100;"<=12")-СЧЁТЕСЛИ(A1:A100;"<6")

Эта формула корректно работает, но ее написание может вызвать затруднения. Формула подсчитывает количество ячеек со значениями, меньше или равно 12, а затем вычитает количество ячеек со значениями, меньше 6.

Функция COUNTBETWEEN обычно используется вместо приведенной выше формулы:

Function COUNTBETWEEN(rng, num1, num2)

' Вычисление количества значений между num1 и num2

Dim CellCount As Integer

Dim cell As Range

Set rng = Intersect(rng.Parent.UsedRange, rng)

CellCount = 0

```

For Each cell In rng
    If cell.Value >= num1 And cell.Value <= num2 Then _
        CellCount = CellCount + 1
Next cell
COUNTBETWEEN = CellCount
End Function

```

Функция COUNTBETWEEN принимает три аргумента:

- ◆ rng. Диапазон
- ◆ num1. Нижний предел
- ◆ num2. Верхний предел

В этой функции используется функция Excel СЧЁТЕСЛИ, которая возвращает количество ячеек в rng, которое больше или равно num1 и меньше или равно num2.

Подсчет видимых ячеек диапазона

Следующая функция COUNTVISIBLE принимает аргумент диапазона и возвращает количество непустых отображаемых ячеек диапазона. Ячейка считается скрытой, если она находится в скрытой строке или столбце.

```

Function COUNTVISIBLE(rng)
    ' Подсчет отображаемых ячеек
    Dim CellCount As Long
    Dim cell As Range
    Application.Volatile
    CellCount = 0
    Set rng = Intersect(rng.Parent.UsedRange, rng)
    For Each cell In rng
        If Not IsEmpty(cell) Then
            If Not cell.EntireRow.Hidden And _
                Not cell.EntireColumn.Hidden Then _
                CellCount = CellCount + 1
        End If
    Next cell
    COUNTVISIBLE = CellCount
End Function

```

Эта функция обрабатывает каждую ячейку диапазона, проверяя, является ли она пустой. Если ячейка непустая, то функция проверяет свойство Hidden строки и столбца ячейки. Если ни столбец, ни строка не являются скрытыми, то переменная CellCount увеличивается на 1.

При работе со средством Автофильтр или в режиме структуры вы можете обратиться к более удобной функции Excel ПРОМЕЖУТОЧНЫЕ.ИТОГИ (с первым аргументом 2 или 3). Однако функция ПРОМЕЖУТОЧНЫЕ.ИТОГИ работает некорректно, если ячейки скрыты вручную с помощью команд Формат⇒Строка⇒Скрыть или Формат⇒Столбец⇒Скрыть. В таком случае функция COUNTVISIBLE является единственной альтернативой.

Суммирование видимых ячеек диапазона

Функция SUMVISIBLE основана на функции COUNTVISIBLE, о которой говорилось в предыдущем разделе. Эта функция принимает аргумент диапазона и возвращает сумму видимых ячеек диапазона. Ячейка считается скрытой, если она находится в скрытом столбце или строке.

```

Function SumVisible(rng)
' Суммирование только отображенных значений
Dim CellSum As Long
Dim cell As Range
Application.Volatile
CellSum = 0
Set rng = Intersect(rng.Parent.UsedRange, rng)
For Each cell In rng
    If IsNumeric(cell) Then
        If Not cell.EntireRow.Hidden And _
            Not cell.EntireColumn.Hidden Then _
            CellSum = CellSum + cell
    End If
Next cell
SumVisible = CellSum
End Function

```

Скрывание и отображение строк и столбцов не приводит к выполнению пересчета рабочего листа. Поэтому, чтобы сделать пересчет, нужно нажать клавиши <Ctrl+Alt+F9>.

Функция Excel ПРОМЕЖУТОЧНЫЕ.ИТОГИ (с первым аргументом 9) также применяется для суммирования видимых ячеек в списке автофильтра. Однако функция ПРОМЕЖУТОЧНЫЕ.ИТОГИ работает некорректно, если в неотфильтрованном списке присутствуют скрытые ячейки.

Функции управления датами

В главе 6 предложено несколько полезных функций и формул Excel для вычисления дат, времени и временных периодов. В этом разделе приведены дополнительные функции для работы с датами.

Расчет следующего понедельника

Приведенная ниже функция NEXTMONTH принимает дату в качестве аргумента и возвращает дату следующего понедельника.

```

Function NEXTMONDAY(d As Date) As Date
' Возвращение следующего понедельника (Monday)
NEXTMONDAY = d + 8 - WeekDay(d, vbMonday)
End Function

```

В этой функции используется функция VBA WeekDay, которая возвращает целое число, соответствующее дню недели для текущей даты (1=Sunday, 2=Monday и т.д.). Также в этой функции используется заранее заданная константа vbMonday.

Приведенная ниже формула возвращает 31/12/2001, что является первым понедельником после Рождества 2001 года (Рождество выпадает на вторник):

```
=NEXTMONDAY(DATE(2001;12;25))
```



Функция возвращает порядковый номер даты. Чтобы отобразить порядковый номер в виде действительной даты, нужно изменить формат номера в ячейке.

Если аргумент, обработанный функцией NEXTMONDAY, выступает понедельником, функция возвратит значение *следующего* понедельника. Чтобы функция возвратила значение этого понедельника, введите следующий код:

```

Function NEXTMONDAY2(d As Date) As Date
'   Возвращение следующего понедельника или текущей даты, если
'   d равно aMonday
  If WeekDay(d) = 2 Then
    NEXTMONDAY2 = d
  Else
    NEXTMONDAY2 = d + 8 - WeekDay(d, vbMonday)
  End If
End Function

```

Вычисление следующего дня недели

Следующая функция NEXTDAY является вариацией функции NEXTMONDAY. Эта функция принимает два аргумента: дату и целое число в диапазоне от 1 до 7, соответствующее дню недели (1=Sunday, 2=Monday и т.д.). Функция NEXTDAY возвращает дату следующего дня недели.

```

Function NEXTDAY(d As Date, day As Integer) As Variant
'   Возвращение следующего определенного дня недели
'   Убедитесь что день задан в диапазоне 1-7
  If day < 1 Or day > 7 Then
    NEXTDAY = CVErr(xlErrNA)
  Else
    NEXTDAY = d + 8 - WeekDay(d, day)
  End If
End Function

```

В функции NEXTDAY используется оператор If, обеспечивающий приемлемость аргумента дня (принадлежит диапазону от 1 до 7). Если аргумент дня является неприемлемым числом, функция возвращает значение #Н/Д. Поскольку функция может вернуть другое значение, а не дату, ее тип описан как variant.

Неделя месяца

Приведенная ниже функция MONTHWEEK возвращает целое число, соответствующее номеру недели месяца.

```

Function MONTHWEEK(d As Date) As Integer
'   Возвращение недели месяца для указанной даты
  Dim FirstDay As Integer

'   Проверка правильности аргумента
  If Not IsDate(d) Then
    MONTHWEEK = CVErr(xlErrNA)
    Exit Function
  End If

'   Получение первого дня месяца
  FirstDay = WeekDay(DateSerial(Year(d), Month(d), 1))

'   Вычисление недели месяца
  MONTHWEEK = Application.RoundUp((FirstDay + day(d) - 1) / 7, 0)
End Function

```

Управление датами до 1900 года

Многие пользователи удивляются, когда выясняется, что Excel не умеет работать с датами до 1900 года. Чтобы исправить этот недостаток, я создал надстройку, которая называется Extended Date Functions (Функции продолжения дат). Эта надстройка позволяет работать с датами начиная с 0100 года и заканчивая 9999 годом.

Установленная настройка Extended Date Functions дает возможность работать с восемью дополнительными функциями рабочего листа:

- ◆ **XDATE(y,m,d,fmt)**. Возвращает дату определенного года, месяца и дня. В качестве параметра можно задать строку формата даты.
- ◆ **XDATEADD(xdate1,days,fmt)**. Добавляет определенное число дней к дате. В качестве параметра можно задать строку формата даты.
- ◆ **XDATEIF(xdate1,xdate2)**. Возвращает количество дней между двумя датами.
- ◆ **XDATEYEARDIF(xdate1,xdate2)**. Возвращает количество полных лет между двумя датами (используется при вычислении возраста).
- ◆ **XDATEYEAR(xdate1)**. Возвращает год указанной даты.
- ◆ **XDATEMONTH(xdate1)**. Возвращает месяц указанной даты.
- ◆ **XDATEDAY(xdate1)**. Возвращает день указанной даты.
- ◆ **XDATEDOW(xdate1)**. Возвращает день недели указанной даты (в виде целого числа от 1 до 7).



Функции не подстраиваются под изменения, вносимые в календарь в 1582 году. Поэтому результаты управления датами до 15 октября 1582 года могут быть неправильными.

Возвращение последней непустой ячейки столбца или строки

В этом разделе представлено две функции: **LASTINCOLUMN**, которая возвращает содержимое последней непустой ячейки в столбце, и **LASTINROW**, которая возвращает содержимое последней непустой ячейки в строке. В главе 13 описаны формулы массивов, используемые с той же целью, но для решения этой задачи можно также использовать и пользовательские функции.

Каждая из этих функций в качестве одного аргумента принимает диапазон. Аргументом может быть ссылка на столбец (для **LASTINCOLUMN**) или строку (для **LASTINROW**). Если принимаемый аргумент соответствует ссылке на неполную строку или столбец (например, 3:3 или D:D), то в функции используется столбец или строка левой верхней ячейки диапазона. Например, следующая формула возвращает содержимое последней непустой ячейки из столбца B:

```
=LASTCOLUMN(B5)
```

А следующая формула возвращает содержимое последней непустой ячейки в строке 7:

```
=LASTROW(C7:D9)
```

Функция LASTINCOLUMN

Ниже приведена функция LASTINCOLUMN:

```
Function LASTINCOLUMN(rng As Range)
' Возвращение содержимого последней непустой ячейки столбца
Application.Volatile
With rng.Parent
    With .Cells(.Rows.Count, rng.Column)
        If Not IsEmpty(.Value) Then
            LASTINCOLUMN = .Value
        ElseIf IsEmpty(.End(xlUp)) Then
            LASTINCOLUMN = ""
        Else
            LASTINCOLUMN = .End(xlUp).Value
        End If
    End With
End With
End Function
```

Обратите внимание на ссылку Parent диапазона. Она позволяет функции работать с аргументами, которые ссылаются на различные рабочие листы и книги.

Функция LASTINROW

Следующая функция LASTINROW:

```
Function LASTINROW(rng As Range)
' Возвращение содержимого последней непустой ячейки строки
Application.Volatile
With rng.Parent
    With .Cells(rng.Row, .Columns.Count)
        If Not IsEmpty(.Value) Then
            LASTINROW = .Value
        ElseIf IsEmpty(.End(xlToLeft)) Then
            LASTINROW = ""
        Else
            LASTINROW = .End(xlToLeft).Value
        End If
    End With
End With
End Function
```

Функции для работы в нескольких листах

Иногда нужно создать функцию для работы с данными, которые содержатся на нескольких рабочих листах одной рабочей книги. В этом разделе предлагается две функции VBA, которые позволяют работать с данными из различных листов, в том числе функции, которые обходят ограничения Excel на копирование формул в другой лист.

Возвращение максимального значения из рабочих листов

Чтобы определить максимальное значение (например, B1) на нескольких рабочих листах, используйте такую формулу:

```
=МАКС (Лист1:Лист4!B1)
```

Эта формула возвращает максимальное значение ячейки B1 листов Лист1, Лист4 и всех листов, находящихся между ними. Но что делать, если после Лист4 нужно добавить новый лист (Лист5)? Формула автоматически не обновляется, поэтому нужно отредактировать ее и добавить ссылку на новый лист:

```
=МАКС (Лист1:Лист5!B1)
```

Следующая функция принимает аргумент одной ячейки и возвращает максимальное значение этой ячейки на всех рабочих листах рабочей книги. Например, приведенная ниже формула возвращает максимальное значение ячейки B1 на всех листах рабочей книги.

```
=MAXALLSHEETS (B1)
```

При добавлении нового листа редактировать формулу не нужно.

```
Function MAXALLSHEETS(cell As Range)
    Dim MaxVal As Double
    Dim Addr As String
    Dim Wksht As Object
    Application.Volatile
    Addr = cell.Range("A1").Address
    MaxVal = -9.9E+307
    For Each Wksht In cell.Parent.Parent.Worksheets
        If Wksht.Name = cell.Parent.Name And _
            Addr = Application.Caller.Address Then
            Else
                If IsNumeric(Wksht.Range(Addr)) Then
                    If Wksht.Range(Addr) > MaxVal Then _
                        MaxVal = Wksht.Range(Addr).Value
                End If
            End If
        Next Wksht
    If MaxVal = -9.9E+307 Then MaxVal = 0
    MAXALLSHEETS = MaxVal
End Function
```

Оператор For Each используется для обращения к рабочей книге:

```
cell.Parent.Parent.Worksheets
```

Родителем ячейки является рабочий лист, а родителем рабочего листа — рабочая книга. Таким образом, цикл Each-Next обрабатывает все рабочие листы в рабочей книге. Первый оператор If внутри цикла выполняет проверку на наличие функции в проверяемой ячейке. Если функция обнаружена, ячейка игнорируется во избежание ошибки циклической ссылки.



Функцию MAXALLSHEETS можно настроить для выполнения других вычислений по нескольким листам: Мин, СРЗНАЧ, СУММ и т.д.

Функция SHEETOFFSET

Неоднократные недовольства приложением Excel (в том числе и Excel 2002) заключаются в недостаточной поддержке относительных ссылок на листы. Например, представим, что есть рабочая книга с несколькими листами, и на листе Лист2 вводится следующая формула:

```
=Лист1!A1+1
```

Такая формула прекрасно работает. Однако если копировать эту формулу на следующий лист (Лист3), то формула также будет ссылаться на Лист1. Если вставить лист между Лист1 и Лист2, формула все равно продолжит ссылаться на Лист1 (скорее всего она должна ссылаться на только что вставленный лист). Фактически, нельзя создать формулу, которая бы ссылалась на рабочие листы относительным образом. Однако можно воспользоваться функцией SHEETOFFSET, чтобы избежать этих ограничений:

Функция SHEETOFFSET. Первый шаг

Приведенная ниже процедура функции VBA называется SHEETOFFSET.

```
Function SHEETOFFSET(offset As Integer, Ref As Range)
' Возвращение содержимого листа для Ref при изменении листа
Dim WksIndex As Integer
Application.Volatile
WksIndex = WorksheetIndex(Application.Caller.Parent)
SHEETOFFSET = Worksheets(WksIndex + offset).Range(Ref.Address)
End Function
```

Функция SHEETOFFSET принимает два аргумента:

- ♦ offset. Смещение листа, который может быть положительным, отрицательным или равным 0.
- ♦ ref. Ссылка на одну ячейку. Если аргумент offset равен 0, то ссылка на ячейку не должна быть такой же, что и на ячейку, содержащую формулу. Иначе получится циклическая ошибка.

Следующая формула возвращает значение ячейки A1 листа, который расположен перед листом с формулой:

```
=SHEETOFFSET(-1;A1)
```

Следующая формула возвращает значение ячейки A1 листа, стоящего после листа с формулой:

```
=SHEETOFFSET(1;A1)
```

Функция корректно работает в большинстве случаев. Например, можно скопировать формулу на другой лист, и относительные ссылки повлияют на все скопированные формулы. А если вставить рабочий лист, то ссылка на лист отрегулируется автоматически.

Однако у этой функции имеется недостаток: если в рабочей книге есть чужеродный рабочий лист (т.е. лист с диаграммой или диалоговые листы Excel 5), то функция может работать некорректно, поскольку попытается сослаться на ячейку на листе, которой там нет.

Функция SHEETOFFSET. Второй шаг

Представленную выше версию функции SHEETOFFSET можно улучшить. В этой версии функции используется вторая функция, которая называется WorksheetIndex. Эта функция возвращает индекс рабочего листа объекта Worksheet, принимаемого в качестве аргумента. Затем в этой функции используется специальное значение для определения другого рабочего

листа. Ниже приведена версия SHEETOFFSET, которая обычно игнорирует все чужеродные рабочие листы в рабочей книге.

```
Private Function WorksheetIndex(x As Worksheet) As Integer
' Возвращение индекса Worksheets (не Sheets)
Dim wks As Worksheet, WksNum As Integer
WksNum = 1
For Each wks In x.Parent.Worksheets
    If x.Name = wks.Name Then
        WorksheetIndex = WksNum
        Exit Function
    End If
    WksNum = WksNum + 1
Next wks
End Function
```

Обратите внимание, что поскольку функция WorksheetIndex не предназначена для использования в формуле, она описана ключевым словом Private. Таким образом предотвращается ее появление в диалоговом окне вставки функции.

Дополнительные пользовательские функции

В этом разделе рассматриваются дополнительные функции. Примеры демонстрируют некоторые специальные методы, которые можно реализовать в пользовательских функциях.

- ◆ Возвращение ошибки из функции
- ◆ Возвращение массива из функции
- ◆ Использование необязательных аргументов функции
- ◆ Использование неопределенного количества аргументов функции
- ◆ Использование функций Windows API

Возвращение значения Error

В некоторых случаях пользовательская функция должна возвращать значение, соответствующее ошибке. Предположим, что это функция REVERSETEXT, которая представлена ранее в этой главе.

```
Function REVERSETEXT(text) As String
' Возвращение аргумента в обратном порядке
REVERSETEXT = StrReverse(text)
End Function
```

Функция возвращает обратное значение содержимого ячейки-аргумента (может быть текстом или значением). Если аргумент является многозначным диапазоном, то функция возвращает #ЗНАЧ!

Предположим, что эта функция должна работать только со строками. Если в аргументе содержится не строка, то функция должна вернуть значение ошибки (#N/D). Можно просто присвоить строку, которая выглядела бы как значение ошибки формулы Excel. Например:

```
REVERSETEXT = "#N/D"
```

Хотя строка выглядит как значение ошибки, она не воспринимается другими формулами. Чтобы действительно вернуть значение ошибки, воспользуйтесь функцией VBA CVerr, которая преобразует номер ошибки в действительную ошибку.

К счастью, в VBA есть встроенные константы для ошибок, которые должны вернуться пользовательской функцией. Эти константы перечислены ниже:

- ◆ xlErrDiv0
- ◆ xlErrNA
- ◆ xlErrName
- ◆ xlErrNull
- ◆ xlErrNum
- ◆ xlErrRef
- ◆ xlErrValue

Ниже приведена функция REVERSETEXT:

```
Function REVERSETEXT(text) As Variant
' Возвращение аргумента в обратном порядке
If Application.ISNONTEXT(text) Then
REVERSETEXT = CVerr(xlErr)
Else
REVERSETEXT = StrReverse(text)
End If
End Function
```

В этой функции используется функция Excel ISNONTEXT для определения того, является ли аргумент текстовой строкой. Если аргумент не является текстовой строкой, то функция возвращает ошибку #Н/Д. В противном случае она возвращает символы в обратном порядке.



Тип исходной функции REVERSETEXT обозначен как String, поскольку функция возвращает текстовую строку. В исправленной версии функция описана как Variant, поскольку теперь она возвращает не строку.

Возвращение массива из функции

Большинство функций, разработанных в VBA, возвращают одно значение. Однако можно написать функцию, которая бы возвращала несколько значений в виде массива.



В части 3 рассматриваются массивы и используемые в них формулы. В этих главах приводятся примеры формулы, которая возвращает несколько значений в отдельных ячейках. Как показано далее, можно также создать пользовательские функции, возвращающие массивы.

В VBA есть весьма полезная функция, которая называется Array. Эта функция возвращает массив данных типа variant. Важно понимать, что возвращаемый массив и обычный массив, составленный из элементов типа Variant, — не одно и то же. Другими словами, массив variant и массив из значений variants — это не одно и то же.

Если формулы массива в Excel вам уже известны, тогда можно приступить к ознакомлению с функцией Array. Формула массива вводится в ячейку после нажатия клавиш <Ctrl+Shift+Enter>. Excel вокруг формулы автоматически добавляет скобки, чтобы показать, что это формула массива. Более подробно о формулах массива рассказано в главе 12.



Нижняя граница массива, созданного с помощью функции Array, по умолчанию равна 0. Но ее можно изменить, воспользовавшись оператором Option Base.

Следующая функция MONTHNAMES показывает, как вернуть массив процедурой функции.

```
Function MONTHNAMES() As Variant
    MONTHNAMES = Array( _
        "Jan", "Feb", "Mar", "Apr", _
        "May", "Jun", "Jul", "Aug", _
        "Sep", "Oct", "Nov", "Dec")
End Function
```

На рис. 25.5 показан рабочий лист с использованием функции MONTHNAMES. Функция вводится после выбора диапазона A4:L4 и введения следующей формулы:

```
{=MONTHNAMES() }
```

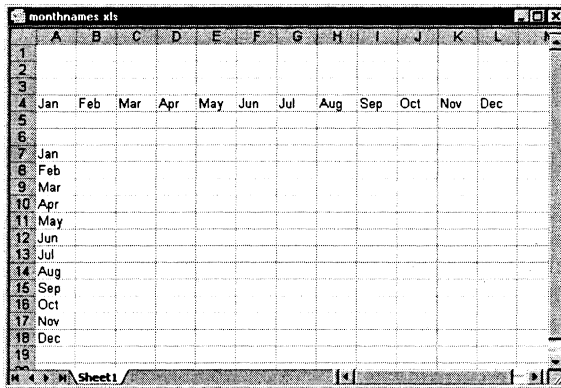


Рис. 25.5. Функция MONTHNAMES, введенная формулой массива



Как и при работе с формулой массива, чтобы ввести формулу, нужно нажать клавиши <Ctrl+Shift+Enter>. Не ставьте скобок — Excel добавит их сама.

Функция MONTHNAMES, как уже говорилось, возвращает горизонтальный массив размером в одну строку. Чтобы отобразить массив вертикального диапазона в один столбец (как на рис. 25.5 A7:A18), выделите диапазон и введите следующую формулу:

```
{=ТРАНСП(MONTHNAMES() ) }
```

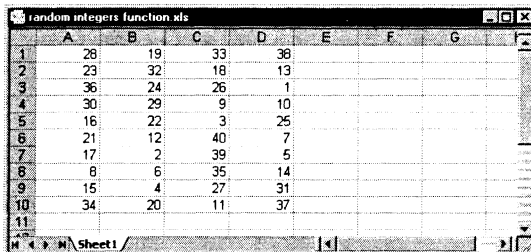
Чтобы сделать перегруппировку, можно изменить функцию. В следующей функции для возвращения вертикального массива используется функция Excel ТРАНСП.

```
Function VMONTHNAMES() As Variant
    VMONTHNAMES = Application.Transpose(Array( _
        "Jan", "Feb", "Mar", "Apr", _
        "May", "Jun", "Jul", "Aug", _
        "Sep", "Oct", "Nov", "Dec"))
End Function
```

Возвращение массива из неповторяющихся случайных целых чисел

Функция `RANDOMINTEGERS` возвращает массив, состоящий из неповторяющихся целых чисел. Эта функция используется в формулах многоячеечных массивов. На рис. 25.6 показан рабочий лист, в котором в диапазоне `A1:D10` используется следующая формула.

```
{=RANDOMINTEGERS() }
```



	A	B	C	D	E	F	G
1	28	19	33	38			
2	23	32	18	13			
3	36	24	26	1			
4	30	29	9	10			
5	16	22	3	25			
6	21	12	40	7			
7	17	2	39	5			
8	8	6	35	14			
9	15	4	27	31			
10	34	20	11	37			

Рис. 25.6. Формула массива генерирует неповторяющиеся целые числа, организованные в случайном порядке

Эта формула вводится в диапазон при нажатии `<Ctrl+Shift+Enter>`. Формула возвращает массив неповторяющихся случайных целых чисел. Поскольку формула охватывает 40 ячеек, то диапазон целых чисел лежит в интервале от 1 до 40. Далее приведен код формулы `RANDOMINTEGERS`:

```
Function RANDOMINTEGERS()  
    Dim FuncRange As Range  
    Dim V() As Integer, ValArray() As Integer  
    Dim CellCount As Double  
    Dim i As Integer, j As Integer  
    Dim r As Integer, c As Integer  
    Dim Temp1 As Variant, Temp2 As Variant  
    Dim RCount As Integer, CCount As Integer  
    Randomize  
  
    ' Создание объекта Range  
    Set FuncRange = Application.Caller  
  
    ' Возвращение ошибки, если FuncRange слишком большое  
    CellCount = FuncRange.Count  
    If CellCount > 1000 Then  
        RANDOMINTEGERS = CVErr(xlErrNA)  
        Exit Function  
    End If  
  
    ' Назначение переменных  
    RCount = FuncRange.Rows.Count  
    CCount = FuncRange.Columns.Count  
    ReDim V(1 To RCount, 1 To CCount)  
    ReDim ValArray(1 To 2, 1 To CellCount)  
  
    ' Заполнение массива произвольными числами  
    ' и целочисленными значениями
```

```

For i = 1 To CellCount
    ValArray(1, i) = Rnd
    ValArray(2, i) = i
Next i

' Сортировка ValArray
For i = 1 To CellCount
    For j = i + 1 To CellCount
        If ValArray(1, i) > ValArray(1, j) Then
            Temp1 = ValArray(1, j)
            Temp2 = ValArray(2, j)
            ValArray(1, j) = ValArray(1, i)
            ValArray(2, j) = ValArray(2, i)
            ValArray(1, i) = Temp1
            ValArray(2, i) = Temp2
        End If
    Next j
Next i

' Заполнение произвольными значениями массива V
i = 0
For r = 1 To RCount
    For c = 1 To CCount
        i = i + 1
        V(r, c) = ValArray(2, i)
    Next c
Next r
RANDOMINTEGERS = V
End Function

```

Перетасовка массива

Приведенная ниже функция RANGERANDOMIZE принимает аргумент диапазона и возвращает введенный массив, элементы которого располагаются в случайном порядке.

```

Function RANGERANDOMIZE(rng)
    Dim V() As Variant, ValArray() As Variant
    Dim CellCount As Double
    Dim i As Integer, j As Integer
    Dim r As Integer, c As Integer
    Dim Temp1 As Variant, Temp2 As Variant
    Dim RCount As Integer, CCount As Integer
    Randomize

' Возвращение ошибки, если rng слишком большое
CellCount = rng.Count
If CellCount > 1000 Then
    RANGERANDOMIZE = CVErr(xlErrNA)
    Exit Function
End If

' Назначение переменных
RCount = rng.Rows.Count
CCount = rng.Columns.Count
ReDim V(1 To RCount, 1 To CCount)
ReDim ValArray(1 To 2, 1 To CellCount)

```

```

' Заполнение ValArray произвольными числами
' и значениями из rng
For i = 1 To CellCount
    ValArray(1, i) = Rnd
    ValArray(2, i) = rng(i)
Next i

' Сортировка ValArray
For i = 1 To CellCount
    For j = i + 1 To CellCount
        If ValArray(1, i) > ValArray(1, j) Then
            Temp1 = ValArray(1, j)
            Temp2 = ValArray(2, j)
            ValArray(1, j) = ValArray(1, i)
            ValArray(2, j) = ValArray(2, i)
            ValArray(1, i) = Temp1
            ValArray(2, i) = Temp2
        End If
    Next j
Next i

' Занесение произвольных значений в массив V
i = 0
For r = 1 To RCount
    For c = 1 To CCount
        i = i + 1
        V(r, c) = ValArray(2, i)
    Next c
Next r
RANGERANDOMIZE = V
End Function

```

Этот код немного похож на код функции `RANDOMINTEGERS`. На рис. 25.7 показана эта функция в действии. Формула массива в диапазоне `C2:C11` такая:

```
{=RANGERANDOMIZE(A2:A11)}
```

Эта формула возвращает содержимое ячеек `A2:A11`, но в случайном порядке.

Использование необязательных аргументов

Во многих встроенных функциях рабочих листов Excel используются необязательные аргументы. Например, функция `ЛЕВСИМВ` возвращает символы левой части строки. В этой функции используется следующий синтаксис:

```
ЛЕВСИМВ (text; num_chars)
```

Первый аргумент обязателен, а второй задается по желанию. Если опустить обязательный аргумент, то Excel примет предполагаемое значение, в данном случае 1.

В пользовательских функциях, которые создаются в VBA, тоже есть необязательные аргументы. Перед именем таких аргументов стоит ключевое слово `Optional`. Ниже приведена простая функция, которая возвращает имя пользователя.

```
Function USER()
USER = Application.UserName
Enf Function

```


Представим, что в некоторых случаях имя пользователя должно отображаться в верхнем регистре. В следующей функции используется необязательный аргумент.

```
Function USER(Optional UpperCase As Boolean)
If IsMissing(UpperCase) Then UpperCase = False
If UpperCase = True Then
USER = Ucase(Application.UserName)
Else
USER = Application.UserName
End If
End Function
```

Если аргумент FALSE пропущен, то имя пользователя возвращается без каких-либо изменений. Если же аргумент TRUE, то имя пользователя преобразуется в верхний регистр (при помощи функции VBA UCase) и только после этого отображается на экране. Обратите внимание, что в первом операторе в процедуре используется функция VBA IsMissing для определения того, указывался ли аргумент. Если аргумента нет, то оператор устанавливает переменную UpperCase равной FALSE (по умолчанию).

Все приведенные ниже формулы являются правильными (и первые две выдают одинаковый результат):

```
=USER ()
=USER (FALSE)
=USER (TRUE)
```

	A	B	C	D	E	F
1	Original		Randomized			
2	Alan		Jack			
3	Bob		Hilda			
4	Cindy		Alan			
5	Dave		Frank			
6	Ellen		Dave			
7	Frank		Ellen			
8	Ginny		Cindy			
9	Hilda		Irvin			
10	Irvin		Bob			
11	Jack		Ginny			
12						
13						

Рис. 25.7. Функция RANGERANDOMIZE возвращает диапазон, элементы которого расположены в случайном порядке

Использование неопределенного количества аргументов

В некоторых функциях рабочего листа Excel используется заданное количество аргументов. Примером такой функции может быть функция СУММ, в которой применяется следующий синтаксис:

```
СУММ(число1; число2...)
```

Первый аргумент обязателен, а после него можно указать вплоть до 29 аргументов. Вот пример формулы, в которой используется функция СУММ с четырьмя диапазонами в качестве аргументов:

```
=СУММ(A1:A5; C1:C5; E1:E5; G1:G5)
```

Типы аргументов могут быть одинаковыми и разными. Например, в следующем примере используется три аргумента — диапазон, значение и оператор.

```
=СУММ(A1:A5; 12; 24*3)
```

Можно создать процедуру с неопределенным количеством аргументов. Суть в том, чтобы использовать массив в качестве последнего (и единственного) аргумента с ключевым словом ParamArray.



ParamArray можно применить только к последнему аргументу процедуры. У него всегда тип variant и это всегда необязательный аргумент (хотя ключевое слово Optional не используется).

Простой пример неопределенных аргументов

Ниже приведена процедура функции, в которой может использоваться любое количество однозначных аргументов. Она возвращает сумму аргументов.

```
Function SIMPLESUM(ParamArray arglist() As Variant) As Double
    Dim arg as Variant
    For Each arg In arglist
        SIMPLESUM = SIMPLESUM + arg
    Next arg
End Function
```

Приведенная ниже формула возвращает сумму аргументов-ячеек:

```
=SIMPLESUM(A1;A5;12)
```

Наиболее серьезное ограничение функции SIMPLESUM заключается в том, что она не работает с несколькими диапазонами. А следующая версия функции работает:

```
Function SIMPLESUM(ParamArray arglist() As Variant) As Double
    Dim arg as Variant
    Dim cell as Range
    For Each arg In arglist
        If TypeName(arg) = «Range» Then
            For Each cell In arg
                SIMPLESUM = SIMPLESUM + cell
            Next cell
        Else
            SIMPLESUM = SIMPLESUM + arg
        End If
    Next arg
End Function
```

Эта функция проверяет каждую запись массива Arglist. Если запись является диапазоном, то в коде используется цикл For Each-Next, чтобы выполнить суммирование ячеек диапазона.

Даже эта версия не является заменой функции Excel СУММ. Проверьте эту функцию в действии, используя различные типы аргументов, и вы увидите, что функция не работает, если все аргументы не являются значениями или ссылкой на диапазон. Также если аргумент состоит из целого столбца, то функция будет работать очень медленно, поскольку она просчитывает все ячейки, в том числе пустые.

Имитирование функции СУММ

В этом разделе представлена процедура функции, которая называется MYSUM. В отличие от функции SIMPLESUM, о которой рассказывалось в предыдущем разделе, MYSUM прекрасно заменяет функцию Excel СУМ.

Перед тем как взглянуть на код функции MYSUM, давайте проанализируем функцию Excel СУММ. В этой функции может быть любое количество аргументов (даже “пропущенные” аргументы); аргументами могут быть цифры, ячейки, диапазоны, текстовое выражение цифр, логические значения и даже внедренные функции. Например, посмотрите на следующую формулу:

```
=SUM(A1;5;"6";;ИСТИНА;SQRT(4);B1:B5)
```

Эта формула — правильная — она содержит все типы аргументов, перечисленные ниже в порядке появления в формуле:

- ◆ ссылка на одну ячейку (A1)
- ◆ константа (5)
- ◆ строка в виде точного значения ("6")


```

        If args(i) = "True" Then MySum = MySum + 1
    Case "Date"
        MySum = MySum + args(i)
    Case Else
        MySum = MySum + args(i)
    End Select
End If
Next i
End Function

```

При написании кода с использованием функции `MYSUM`, помните о следующем:

- ◆ Пропущенные аргументы (определяются функцией `IsMissing`) просто игнорируются.
- ◆ Для определения типа аргумента в процедуре используется функция `VBA TypeName` (`Range`, `Error` или др.). Каждый тип аргумента обрабатывается по-своему.
- ◆ Для аргумента диапазона функция обрабатывает каждую ячейку диапазона и добавляет ее значение к общему значению.
- ◆ Тип функции `variant`, поскольку функция должна возвращать ошибку, если один из аргументов является значением ошибки.
- ◆ Если аргумент содержит ошибку (например, `#ДЕЛО!`), то функция `MYSUM` возвратит ошибку, — как и функция `Excel СУММ`.
- ◆ Функция `Excel СУММ` предполагает, что текстовая строка имеет нулевое значение, если не указать другую константу (т.е. значение, а не переменная). Следовательно, `MySum` добавляет значение ячейки, только если его можно вычислить численно (для этого используется функция `Excel IsNumeric`).
- ◆ При работе с логическими аргументами нужно быть очень внимательным. Чтобы `MySum` могла заменить `СУММ`, нужно проверить наличие `ИСТИНА` в списке аргументов и компенсировать разность (т.е., если прибавить 2 к `-1`, то получится 1).
- ◆ Для аргументов диапазона функции используется оператор `Intersect`, чтобы создать временный диапазон, состоящий из пересечения диапазонов на одном и на разных листах. При этом учитываются случаи, когда аргумент диапазона состоит из целой строки или столбца, для вычисления которых потребуется целая вечность.

Возможно, возникнут вопросы по поводу относительной производительности `СУММ` и `MySum`. Конечно, `MySum` работает намного медленнее, но насколько медленно, зависит от производительности системы и самих формул. В моей системе рабочий лист с 1000 формулам `СУММ` вычисляется мгновенно. После того как я заменил функцию `СУММ` на `MySum`, на пересчет потребовалось 12 секунд. `MySum` у вас, может быть, работает немного быстрее, но скорости этих двух функций несравнимы.

Надеюсь также, что вы понимаете, что целью этого примера является не создание новой функции `СУММ`. На самом деле здесь демонстрировалось, как создавать пользовательские функции рабочего листа, которые работают и выглядят как встроенные функции `Excel`.

Резюме

В этой главе представлено множество примеров пользовательских процедур функций, которые можно использовать в формулах рабочих листов. Большинство этих функций можно просто скопировать из приведенных примеров. Некоторые нужно настроить так, чтобы они соответствовали требованиям задачи.

Приложения

ПРИЛОЖЕНИЕ А

Работа с импортированными
файлами Lotus 1-2-3

ПРИЛОЖЕНИЕ Б

Описание функции Excel

ПРИЛОЖЕНИЕ В

Пользовательские форматы чисел

ПРИЛОЖЕНИЕ Г

Дополнительные ресурсы

Приложение А

Работа с импортированными файлами Lotus 1-2-3

Когда-то Lotus 1-2-3 было основным приложением для работы с электронными таблицами. Теперь этого удостоена Excel, что и использует в 90 случаев из 100 для управления таблицами. Конечно, некоторые пользователи до сих пор используют Lotus 1-2-3, и может возникнуть ситуация, когда нужно импортировать файл, созданный в Lotus 1-2-3. Поэтому информация этого приложения может быть полезной.

О файлах Lotus 1-2-3

На сегодняшний день существует много версий Lotus 1-2-3, и файлы Lotus 1-2-3 могут представляться в различных форматах. В табл. А.1 приведены наиболее часто встречаемые файлы Lotus 1-2-3.

Таблица А.1. Типы файлов Lotus 1-2-3

Расширение файла	Описание
WKS	Создан в Lotus 1-2-3 for DOS Release 1.0 и 1.0a. В этих файлах содержится один лист. Excel может получать и записывать такие файлы.
WK1	Создан в Lotus 1-2-3 for DOS Release 2.x. В этих файлах содержится один лист и может содержаться приложение *.FMT или *.ALL с информацией о форматировании. В Excel можно получать такие файлы, но сохранять только активный лист.
WK3	Создан в Lotus 1-2-3 for DOS Release 3.x и Lotus 1-2-3 для Windows версии 1.0. В этих файлах может содержаться несколько листов и может находиться приложение *.FM3 с информацией о форматировании. В Excel такие файлы можно получать и сохранять.
WK4	Создан в Lotus 1-2-3 for Windows Release 4.0. В этих файлах содержится несколько листов. В Excel можно записывать и читать такие файлы.
123	Создан Lotus 1-2-3 for Windows Release 5 и Millennium Edition. В Excel такие не читаются и не записываются.

Увеличение размера файла

Импортировав файл Lotus 1-2-3 и сохранив его как файл Excel можно обнаружить, что размер файла значительно увеличился, что заметно замедляет его открытие и сохранение. Наиболее вероятной причиной этого является то, что импортированный файл Lotus 1-2-3 содержит целые предварительно форматированные столбцы. Когда Excel импортирует такой файл, он преобразует все в форматированные ячейки — даже если они пусты. Чтобы выйти из подобной ситуации, выделите все пустые строки под последней непустой ячейкой рабочего листа и удалите их. Снова сохраните рабочую книгу, и размер файла должен стать более правдоподобным.

При импортировании или экспортировании файлов Lotus 1-2-3 не следует ожидать идеального преобразования. В справочной системе Excel описаны возможные ограничения.



Excel поддерживает связи файлов с рабочими книгами Lotus 1-2-3. Однако это средство ограничено работой с файлами WKS, WK1, WK3 и WK4 (не самые последние форматы файлов Lotus 1-2-3). В некоторых случаях ссылки нужно обновить явно. Для этого воспользуйтесь командой Правка⇒Связи и щелкните на кнопке Обновить.

Формулы Lotus 1-2-3

В одних случаях формулы в импортированном файле Lotus 1-2-3 прекрасно работают в Excel. В других случаях формулы некорректно преобразуются, и их нужно редактировать и переписывать.

Некоторые формулы Excel высчитывает иначе, чем в Lotus 1-2-3. Эти формулы можно разделить на три категории:

- ◆ С использованием текста в вычислениях
- ◆ С использованием логических значений (ИСТИНА или ЛОЖЬ)
- ◆ С использованием критериев баз данных

Чтобы в Excel воспользоваться методами вычисления формул, принятыми в Lotus 1-2-3, выберите команду Сервис⇒Параметры. В диалоговом окне Параметры перейдите на вкладку Переход и выставьте флажок опции Производить вычисления по правилам Lotus 1-2-3.



При открытии файла Lotus 1-2-3 параметр Производить вычисления по правилам Lotus 1-2-3 активизируется автоматически, чтобы указать Excel, что вычисления формул будут выполняться по правилам Lotus 1-2-3.

Пусть Excel научит

Если переходить от старых DOS-версий Lotus 1-2-3 к более новым, то можно обнаружить, что Excel помогает осуществлять такой переход. Секрет таится в команде Справка⇒Справка по Lotus 1-2-3. При выборе этой команды отображается диалоговое окно с перечисленными в левой части командами 1-2-3.

Выберите команду Lotus 1-2-3, и Excel отобразит инструкции или даже покажет соответствующую команду меню. Например, если вы уже давно работаете с Lotus 1-2-3 для DOS, то знаете, что для удаления имени нужно воспользоваться командой /rnd (от Range Name Delete). Если эту команду ввести в диалоговом окне Справка для пользователей Lotus 1-2-3, то появятся инструкции с описанием того, как выполнить это действие в Excel.

Если вами планируется повсеместное использование импортированных файлов Lotus 1-2-3, то следует подумать о преобразовании формул, которые некорректно вычисляются, и отключить параметр Производить вычисления по правилам Lotus 1-2-3. Это поможет избежать путаницы тем, кто не знаком с Lotus 1-2-3.

В следующих разделах предложены советы по преобразованию формул Lotus 1-2-3 так, чтобы они корректно работали в Excel (без активизации опции Производить вычисления по правилам Lotus 1-2-3).

Порядок вычислений

Вычисление формул в Excel и в Lotus 1-2-3 отличаются. Во время вычислений в Lotus 1-2-3 сначала выполняется действие с оператором возведения в степень (^), а затем вычитание (-). В Excel принят обратный порядок. Рассмотрим следующую формулу:

=-3^2

Lotus 1-2-3 возвратит значение -9, а Excel — число +9. Чтобы получить одинаковые результаты, в Lotus 1-2-3 формулу нужно изменить таким образом:

=(3^2)

Текст в вычислениях

В Lotus 1-2-3 считается, что ячейки, содержащие текст, имеют нулевое значение, если они используются в формуле с математическими операторами. Excel в таких случаях возвращает ошибку.



Если параметр Производить вычисления по правилам Lotus 1-2-3 активизирован, то Excel воспринимает текст как нулевое значение.

Следующая формула корректно работает в Lotus 1-2-3 (и возвращает 12). В Excel эта формула возвращает значение #ЗНАЧ!.

="Dog"+12

Подобно этому, если в ячейке A1 содержится текст Dog, а в ячейке A2 содержится значение 12, то следующая формула корректно работает в Lotus 1-2-3, а в Excel возвращает ошибку:

=A1+A2

Тем не менее, Excel позволяет использовать ссылки на ячейки с текстом в аргументах функций, и обрабатывает такие ссылки. Например, следующая формула прекрасно работает как в Lotus 1-2-3, так и в Excel, даже если в диапазоне A1:A10 содержится текст:

=СУММ(A1:A10)

Этот факт можно использовать для преобразования формул Lotus 1-2-3, таких как =A1+A2, в следующий вид:

=СУММ(A1;A2)

Логические значения

В результате вычисления логических выражений в Lotus 1-2-3 возвращается 1 или 0. Excel отображает эти значения в виде ИСТИНА и ЛОЖЬ. Значение ИСТИНА эквивалентно 1 в Lotus 1-2-3, а ЛОЖЬ — 0.



Если параметр Производить вычисления по правилам Lotus 1-2-3 активизирован, то вместо результата ЛОЖЬ Excel отобразит 0, а вместо ИСТИНА — 1.

Например, в Lotus 1-2-3 следующая формула возвратит либо 1, либо 0, в зависимости от содержимого ячеек A1 и A2. В Excel эта формула возвратит либо ИСТИНА, либо ЛОЖЬ.

=A1<A2

Такое различие важно, если в рабочем листе для проверки 0 или 1 используются функции ЕСЛИ. Например, следующая формула возвратит различные результаты в Lotus 1-2-3 и в Excel:

=ЕСЛИ (A1<A2=1 ; B1 ; B2)

Чтобы эта функция корректно работала в Excel, измените ее следующим образом:

=ЕСЛИ (A1<A2 ; B1 ; B2)

В Lotus 1-2-3 используются такие логические операторы: #AND #, #NOT# и #OR#. В Excel вместо них используются логические функции (И, НЕ и ИЛИ). Например, следующая формула Lotus 1-2-3 возвратит строку yes, если ячейка A1=12 и A2=12, и строку no, если обе ячейки не равны 12:

@IF (A1=12#AND#A2=12, "yes", "no")

Эквивалент этой формулы в Excel выглядит так:

=ЕСЛИ (И (A1=12 ; A2=12) ; "yes" ; "no")

Проблемы с датами

Если в импортированных рабочих листах содержатся даты с дефисами-разделителями (например, 12-31-98), то такие даты могут остаться нераспознанными в Excel. Скажем, Excel интерпретирует содержимое ячейки как формулу:

=12-31-98

Такие ячейки нужно отредактировать, чтобы они корректно отображались.

Критерии базы данных

Если в импортированном рабочем листе используются диапазоны с критерием базы данных (например, дополнительная фильтрация), следует быть особенно внимательным. При извлечении данных, поиске данных и использовании функций баз данных диапазоны с критерием баз данных вычисляются по-другому. Например, в 1-2-3 критерию "Ben" соответствуют только те строки, в ячейках которых содержится значение *Ben*. В Excel критерию "Ben" удовлетворяют строки, в которых содержимое ячеек начинается с *Ben* — в том числе *Benjamin*, *Benny* и *Benito*.

Как быть с макросами Lotus 1-2-3?

Некоторые макросы Lotus 1-2-3 выполняются в Excel — макросы, активизируемые с клавиатуры, разработаны в ранних версиях Lotus 1-2-3. Они хранятся непосредственно в рабочем листе и соответствуют сочетаниям клавиш. Идеальной совместимости ожидать не следует.

Обычно для таких макросов используют имена диапазонов, например, \t. Этот макрос выполняется при нажатии клавиш <Ctrl+T>. Таким образом, специальные имена приемлемы, если они содержатся в импортированном файле. Но такое имя нельзя создать в Excel.

Если файлы Lotus 1-2-3 конвертировать в Excel, то наилучшим выходом является восстановить макрос с помощью VBA. Таким образом можно увеличить производительность и упростить сохранение макросов.



Если параметр Производить вычисления по правилам Lotus 1-2-3 активизирован, то использование критерия баз данных в Excel подобно его использованию в Lotus 1-2-3.

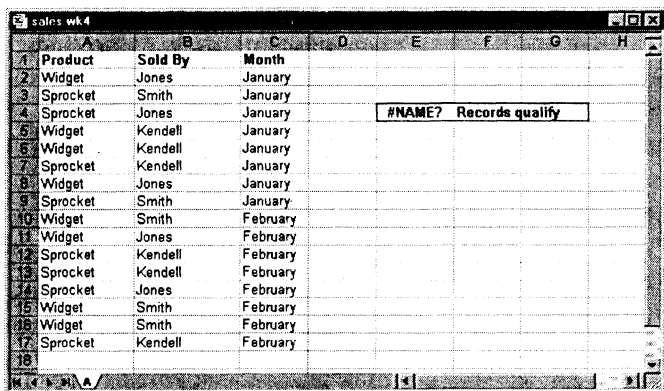
Совместимость функций Lotus 1-2-3

У большинства функций рабочего листа в Lotus 1-2-3 есть эквиваленты в Excel. В некоторых случаях соответствие не совсем точное. К счастью, справка по Excel предоставляет полное описание различий между функциями рабочего листа Lotus 1-2-3 и Excel.

Функции конвертирования баз данных

При работе с функциями баз данных Lotus 1-2-3 (например, @DSUM и @DCOUNT) всегда возникает одна и та же проблема. Lotus 1-2-3 позволяет в качестве аргумента задать критерий. Обратимся к рис. А.1, на котором показан файл Lotus 1-2-3, импортированный в Excel. Следующая формула (в ячейке E4) была некорректно преобразована, из-за чего отобразилась ошибка:

=БСЧЁТА (A1 : C17 ; " Product " ; И (PRODUCT = "Widget " ; MONTH = "January "))



Product	Sold By	Month
Widget	Jones	January
Sprocket	Smith	January
Sprocket	Jones	January
Widget	Kendell	January
Widget	Kendell	January
Sprocket	Kendell	January
Widget	Jones	January
Sprocket	Smith	January
Widget	Smith	February
Widget	Jones	February
Sprocket	Kendell	February
Sprocket	Kendell	February
Sprocket	Jones	February
Widget	Smith	February
Widget	Smith	February
Sprocket	Kendell	February

Рис. А.1. В импортированном файле Lotus 1-2-3 используется функция @DCOUNT с аргументом критерием, не поддерживаемым в Excel

Исходная формула Lotus 1-2-3 должна возвращать количество записей, в которых Product является Widget, а Month — January. Исходная формула (до конвертирования в Excel) была такой: @DCOUNT (A1 : C17 , " Product " , PRODUCT = "Widget " #AND#MONTH = "January ")

К сожалению, функции баз данных Excel не позволяют в качестве аргумента задавать критерий. Вместо этого необходимо выполнить одно из нижеприведенных действий:

- ◆ Установить особый диапазон критерия для функции ДСЧЁТЗ
- ◆ Воспользоваться другой функцией — в данном случае функцией СЧЁТЕСЛИ

На рис. А.2 показан диапазон критериев в E1 : E2. Следующая формула возвращает количество записей, в которых Product является Widget, а Month — January:

=ДСЧЁТА (A1 : C17 , " Product " , E1 : F2)

Product	Sold By	Month	Product	Month
Widget	Jones	January	Widget	January
Sprocket	Smith	January		
Sprocket	Jones	January		
Widget	Kendell	January		
Widget	Kendell	January		
Sprocket	Kendell	January		
Widget	Jones	January		
Sprocket	Smith	January		
Widget	Smith	February		
Widget	Jones	February		
Sprocket	Kendell	February		
Sprocket	Kendell	February		
Sprocket	Jones	February		
Widget	Smith	February		
Widget	Smith	February		
Sprocket	Kendell	February		

Рис. А.2. Использование диапазона критерия для формулы ДСЧЁТЗ

Можно также воспользоваться формулой массива, в которой не нужен диапазон критерия. Следующая формула является эквивалентом Excel некорректно преобразованной формулы @DCOUNT:

`{=СУММ((A2:A17="Widget")*(C2:C17="January"))}`



Введите формулу массива с помощью клавиш <Ctrl+Shift+Enter>. Не ставьте скобок — Excel поставит их сам. Более подробно о формулах массивов рассказано в части III.

Приложение Б

Описание функций Excel

В этом приложении приведен полный список функций, используемых на рабочих листах Excel. Функции расположены в алфавитном порядке и отсортированы в таблицах по категориям, которые приведены в диалоговом окне Мастер функций.

Функции Excel по категориям

В таблицах Б.1-Б.10 приведены следующие категории функций Excel: баз данных, даты и времени, инженерные, финансовые, статистические, логические, просмотра, математические, информационные и управления текстом. Некоторые из этих функций доступны только после установки надстройки “Пакет анализа”.

Чтобы ознакомиться более подробно с информацией о тех или иных функциях, в том числе и об их аргументах, выберите функцию в диалоговом окне Мастер функций и нажмите клавишу <F1>.

В табл. Б.1 приведены функции баз данных Excel.

Таблица Б.1. Функции категории баз данных

Функция	Описание
ДСРЗНАЧ	Возвращает среднее значение из выделенных элементов базы данных
ДСЧЁТ	Подсчитывает количество ячеек, содержащих числа определенных баз данных и удовлетворяющие определенным критериям
ДСЧЁТА	Подсчитывает непустые ячейки определенной базы данных и критерии
ДИЗВЛЕЧЬ	Извлекает из базы данных одну запись, которая удовлетворяет определенному критерию
ДМАКС	Возвращает максимальное значение из выбранных элементов базы данных
ДМИН	Возвращает минимальное значение из выбранных элементов базы данных
ВДПРОИЗВЕД	Умножает значения в определенном поле записей, которые удовлетворяют критерию базы данных
ДСТАНДОТКЛ	Вычисляет стандартное отклонение на примере выбранных элементов базы данных
ДСТАНДОТКЛП	Вычисляет стандартное отклонение из всей совокупности выбранных элементов базы данных
БДСУММ	Суммирует числа в столбце полей с записями в базе данных, которые удовлетворяют критерию
БДДИСП	Вычисляет дисперсию у выбранных элементов базы данных
БДДИСПП	Вычисляет дисперсию от совокупности выбранных элементов базы данных

В табл. Б.2 приведены функции управления датой и временем в Excel.

Таблица Б.2. Функции категории даты и времени

Функция	Описание
ДАТА	Возвращает числовой формат определенной даты
ДАТАЕСЛИ	Вычисляет количество дней, месяцев или лет между двумя датами. Эта функция доступна только в Excel 2000.
ДАТАЗНАЧ	Преобразует дату в виде текста в числовой формат записи
ДЕНЬ	Преобразует последовательный номер в день месяца
ДНЕЙ360	Вычисляет количество дней между двумя датами с учетом года в 360 дней
ДАТАМЕС*	Возвращает порядковый номер даты, который является заданным номером месяца перед или после начальной даты
КОНМЕСЯЦА*	Возвращает серийный номер последнего дня месяца перед или после определенного количества месяцев
ЧАС	Преобразует числовой формат записи даты в часы
МИНУТЫ	Преобразует числовой формат записи даты в минуты
МЕСЯЦ	Преобразует числовой формат записи в месяцы
ЧИСТРАБДНИ*	Возвращает количество всех рабочих дней между двумя датами
ТДАТА	Возвращает числовой формат записи текущей даты и времени
СЕКУНДЫ	Преобразует числовой формат записи в секунды
ВРЕМЯ	Возвращает числовой формат записи определенного времени
ВРЕМЗНАЧ	Преобразует время в виде текста в серийный номер
СЕГОДНЯ	Возвращает числовой формат записи сегодняшней даты
ДЕНЬНЕД	Преобразует числовой формат записи в день недели
НОМНЕДЕЛИ*	Возвращает номер недели в году
РАБДЕНЬ*	Возвращает числовой формат записи даты перед или после определенного количества рабочих дней
ГОД	Преобразует числовой формат записи в год
ДОЛЯГОДА*	Возвращает часть года в виде количества полных дней между начальной датой и конечной датой

* Доступны только при установленной надстройке “Пакет анализа”.

В табл. Б.3 перечислены инженерные функции Excel.

Таблица Б.3. Инженерные функции, доступные при установке надстройки “Пакет анализа”

Функция	Описание
БЕССЕЛЬ. I	Возвращает измененную функцию Бесселя $\ln(x)$
БЕССЕЛЬ. J	Возвращает функцию Бесселя $J_n(x)$
БЕССЕЛЬ. K	Возвращает измененную функцию Bessel $K_n(x)$
БЕССЕЛЬ. Y	Возвращает функцию $Y_n(x)$
ДВ. В. ДЕС	Преобразует двоичное число в десятичное
ДВ. В. ШЕСТН	Преобразует двоичное число в шестнадцатеричное

Функция	Описание
ДВ . В . ВОСЬМ	Преобразует двоичное число в восьмеричное
КОМПЛЕКСН	Преобразует действительный и мнимый коэффициенты в комплексное значение
ПРЕОБР	Преобразует число из одной системы исчисления в другую
ДЕС . В . ДВ	Преобразует десятичное число в двоичное
ДЕС . В . ШЕСТН	Преобразует десятичное число в шестнадцатеричное
ДЕС . В . ВОСЬМ	Преобразует десятичное число в восьмеричное
ДЕЛЬТА	Проверяет, равны ли два значения
ФОШ	Возвращает функцию ошибки
ДФРШ	Возвращает дополнительную функцию ошибки
ПОРОГ	Проверяет, больше ли число, чем пороговое значение
ШЕСТН . В . ДВ	Преобразует шестнадцатеричное значение в двоичное
ШЕСТН . В . ДЕС	Преобразует шестнадцатеричное значение в десятичное
ШЕСТН . В . ВОСЬМ	Преобразует шестнадцатеричное значение в восьмеричное
МНИМ . ABS	Возвращает абсолютное значение (модуль) комплексного числа
МНИМ . ЧАСТЬ	Возвращает мнимую часть комплексного числа
МНИМ . АРГУМЕНТ	Возвращает аргумент — угол, выраженный в радианах
МНИМ . СОПРЯЖ	Возвращает сопряженное комплексное число
МНИМ . COS	Возвращает косинус комплексного числа
МНИМ . ДЕЛ	Возвращает частное двух комплексных чисел
МНИМ . EXP	Возвращает экспоненту комплексного числа
МНИМ . LN	Возвращает натуральный логарифм комплексного числа
МНИМ . LOG10 –	Возвращает логарифм с основанием 10 комплексного числа
МНИМ . LOG2	Возвращает логарифм с основанием 2 комплексного числа
МНИМ . СТЕПЕНЬ	Возвращает комплексное число, возведенное в степень целого числа
МНИМ . ПРОИЗВЕД	Возвращает произведение двух комплексных чисел
МНИМ . ВЕЩ	Возвращает действительный коэффициент комплексного числа
МНИМ . SIN	Возвращает синус комплексного числа
МНИМ . КОРЕНЬ	Возвращает квадратный корень комплексного числа
МНИМ . РАЗН	Возвращает разность двух комплексных чисел
МНИМ . СУММ	Возвращает сумму комплексных чисел
ВОСЬМ . В . ДВ	Преобразует восьмеричное число в двоичное
ВОСЬМ . В . ДЕС	Преобразует восьмеричное число в десятичное
ВОСЬМ . В . ШЕСТН	Преобразует восьмеричное число в шестнадцатеричное

В табл. Б.4 перечислены финансовые функции Excel.

Таблица Б.4. Функции финансовой категории

Функция	Описание
НАКОПДОХОД*	Возвращает начисленные проценты с периодически выплачиваемым процентом
НАКОПДОХОДПОГАЗ*	Возвращает начисленные проценты в конце срока
АМОКУМ*	Возвращает величину амортизации для каждого периода, используя коэффициент амортизации
АМОКУМ*	Возвращает величину амортизации для каждого отчетного периода
ДНЕЙКУПОНДО*	Возвращает количество дней от начала действия облигации до расчетного дня по фьючерсному контракту
ДНЕЙКУПОН*	Возвращает количество дней действия облигации с учетом расчетного дня по фьючерсному контракту
ДНЕЙКУПОНПОСЛЕ*	Возвращает количество дней от расчетного дня по фьючерсному контракту до начала действия следующей облигации
ДАТАКУПОНПОСЛЕ*	Возвращает дату действия следующей облигации после расчетного дня по фьючерсному контракту
ЧИСЛКУПОН*	Возвращает количество облигаций, подлежащих оплате, между расчетным днем по фьючерсному контракту и окончанию срока
ДАТАКУПОНДО*	Возвращает дату действия предыдущей облигации перед расчетным днем по фьючерсному контракту
ОБЩПЛАТ*	Возвращает суммарный процент, выплаченный между двумя периодами
ОБЩДОХОД*	Возвращает кумулятивную сумму основного долга между двумя периодами
ФУО	Возвращает снижение активов за определенный период, используя метод постоянного учета амортизации
ДДОБ	Возвращает снижение активов за определенный период при использовании метода двукратного учета амортизации или иного явно указанного метода
СКИДКА*	Возвращает норму скидки для ценных бумаг
РУБЛЬ.ДЕС*	Преобразует цену в рублях (в виде дроби) в цену в рублях (в виде десятичного числа)
РУБЛЬ.ДРОБЬ*	Преобразует цену в рублях (в виде десятичного числа) в цену в рублях (в виде дроби)
ДЛИТ*	Возвращает годовой срок действия ценной бумаги с периодически выплачиваемым процентом
ЭФФЕКТ*	Возвращает действительную годовую процентную ставку
БС	Возвращает будущую стоимость вложения
ВЗРАСПИС*	Возвращает будущую стоимость исходного капитала после реализации серий сложных процентных ставок
ИНОРМА*	Возвращает процентную ставку полностью вложенных ценных бумаг
ПРПЛАТ	Возвращает сумму выплат процентов от инвестирования за определенный период
ВСД	Возвращает внутреннюю процентную ставку после серии потоков денежных средств
ПРОЦПЛАТ	Возвращает процент, связанный с уплатой определенного займа
МДЛИТ*	Возвращает измененный срок действия Масаулеу за ценные бумаги с предположительной стоимостью \$100
МВСД	Возвращает внутреннюю норму доходности, где положительные и отрицательные денежные потоки учитываются по разным ставкам

Функция	Описание
НОМИНАЛ*	Возвращает годовую номинальную процентную ставку
КПЕР	Возвращает количество периодов инвестирования
ЧПС	Возвращает точную текущую стоимость инвестирования на основе серий периодических денежных потоков и дисконтных ставок
ЦЕНАПЕРВНЕРЕГ*	Возвращает цену ценных бумаг номинальной стоимостью \$100 с нерегулярным первым периодом
ДОХОДПЕРВНЕРЕГ*	Возвращает доход от ценных бумаг с нерегулярным первым периодом
ЦЕНАПОСЛНЕРЕГ*	Возвращает цену ценных бумаг номинальной стоимостью \$100 с нерегулярным последним периодом
ДОХОДПОСЛНЕРЕГ*	Возвращает доход от ценных бумаг с нерегулярным последним периодом
ПЛТ	Возвращает величину выплаты за один период годовой ренты
ОСПЛТ	Возвращает платеж по капиталу за инвестирование за данный период
ЦЕНА*	Возвращает цену ценных бумаг номинальной стоимостью \$100 с периодической выплатой процентов
ЦЕНАСКИДКА*	Возвращает цену дисконтных ценных бумаг номинальной стоимостью \$100
ЦЕНАПОГАШ*	Возвращает цену ценных бумаг номинальной стоимостью \$100 с выплатой процентов по окончании срока
ПС	Возвращает текущую стоимость инвестиций
СТАВКА	Возвращает процентную ставку за период дохода
ПОЛУЧЕНО*	Возвращает сумму, полученную по истечении срока за полностью инвестированные ценные бумаги
АПЛ	Возвращает прямое уменьшение активов за один период
АСЧ	Возвращает уменьшение активов за определенный период, рассчитанное методом "суммы (годовых) чисел".
РАВНОКЧЕК*	Возвращает доходность облигаций по казначейскому векселю
ЦЕНАКЧЕК*	Возвращает цену за номинальную стоимость \$100 по казначейскому векселю
ДОХОДКЧЕК*	Возвращает доход по казначейскому векселю
VDB	Возвращает уменьшение активов за определенный период или часть периода, используя метод сокращающегося баланса
ПУО*	Возвращает величину амортизации имущества для явно указанного или соответствующего периода при использовании метода разового учета амортизации
ЧИСТВНДОХОД*	Возвращает внутреннюю ставку доходности запланированных неперiodических денежных потоков
ДОХОД*	Возвращает доход от ценных бумаг с периодической выплатой процентов
ДОХОДСКИДКА*	Возвращает годовой доход от дисконтных ценных бумаг, например, казначейских векселей
ДОХОДПОГАШ*	Возвращает годовой доход от ценных бумаг с выплатой процентов по окончании срока

*Доступны только при установленной надстройке "Пакет анализа".

В табл. Б.5 приведены информационные функции в Excel.

Таблица Б.5. Информационные функции

Функция	Описание
ЯЧЕЙКА	Возвращает информацию о форматировании, расположении и содержимом ячейки
ТИП.ОШИБКИ	Возвращает число, соответствующее типу ошибки
ИНФОРМ	Возвращает информацию о текущей операционной среде
ЕПУСТО	Возвращает ИСТИНА, если значение пустое
ЕОШ	Возвращает ИСТИНА, если значением является любое значение ошибки, кроме #Н/Д
ЕОШИБКА	Возвращает ИСТИНА, если значением является любое значение ошибки
ЕЧЕТН*	Возвращает ИСТИНА, если число четное
ЕЛОГИЧ	Возвращает ИСТИНА, если значение логическое
ЕНД	Возвращает ИСТИНА, если значение является значением ошибки #Н/Д
ЕНЕТЕКСТ	Возвращает ИСТИНА, если значение не является текстом
ЕЧИСЛО	Возвращает ИСТИНА, если значение является числом
ЕНЕЧЕТ*	Возвращает ИСТИНА, если число нечетное
ЕССЫЛКА	Возвращает ИСТИНА, если значение является ссылкой
ЕТЕКСТ	Возвращает ИСТИНА, если значение является текстом
Ч	Возвращает значение, преобразованное в число
НД	Возвращает значение ошибки #Н/Д
ТИП	Возвращает число, которое обозначает тип значения

* Доступны только при установленной надстройке “Пакет анализа”.

В табл. Б.6 перечислены логические функции Excel.

Таблица Б.6. Логические функции

Функция	Описание
И	Возвращает ИСТИНА, если все его аргументы ИСТИНА
ЛОЖЬ	Возвращает логическое значение ЛОЖЬ
ЕСЛИ	Определяет для выполнения логическую проверку
НЕ	Реверсирует логическое значение аргумента
ИЛИ	Возвращает ИСТИНА, если один из аргументов ИСТИНА
ИСТИНА	Возвращает логическое значение ИСТИНА

В табл. Б.7 перечислены функции просмотра.

Таблица Б.7. Функции просмотра

Функция	Описание
АДРЕС	Возвращает ссылку на одну ячейку рабочего листа в виде текста
ОБЛАСТИ	Возвращает количество областей в ссылке
ВЫБОР	Выбирает значение из списка значений

Функция	Описание
СТОЛБЕЦ	Возвращает номер столбца ссылки
ЧИСЛСТОЛБ	Возвращает количество столбцов в ссылке
ГПР	Обращается к верхней строке массива и возвращает значение указанной ячейки
ГИПЕРССЫЛКА	Создает пиктограмму, щелчок на которой открывает документ на жестком диске, сервере или в Internet
ИНДЕКС	Использует индекс для выбора значения из ссылки или массива
ДВССЫЛ	Возвращает ссылку, указанную текстовым значением
ПРОСМОТР	Ищет значения в векторе или массиве
ПОИСКПОЗ	Ищет значения в ссылке или массиве
СМЕЩ	Возвращает смещение ссылки относительно данной ссылки
СТРОКА	Возвращает номер строки ссылки
ЧСТРОК	Возвращает количество строк в ссылке
ДРВ*	Извлекает реальные данные из программы, поддерживающей COM-автоматизацию
ТРАНСП	Возвращает транспонированный массив
ВПР	Обращается к первому столбцу массива и перемещается по строкам, чтобы вернуть значение ячейки

* Эта функция доступна только в Excel 2002.

В табл. Б.8 приведены арифметические и тригонометрические функции Excel.

Таблица Б.8. Математические функции

Функция	Описание
ABS	Возвращает абсолютное значение числа
ACOS	Возвращает арккосинус числа
ACOSH	Возвращает гиперболический косинус числа
ASIN	Возвращает арксинус числа
ASINH	Возвращает гиперболический синус числа
ATAN	Возвращает арктангенс числа
ATAN2	Возвращает арктангенс от координат x и y
ATANH	Возвращает гиперболический тангенс числа
ОКРВВЕРХ	Округляет число до ближайшего целого или до ближайшего кратного
ЧИСЛКОМБ	Возвращает количество комбинаций для данного числа объектов
COS	Возвращает косинус числа
COSH	Возвращает гиперболический косинус числа
ГРАДУСЫ	Преобразует радианы в градусы
ЧЁТН	Округляет число до ближайшего четного целого числа
EXP	Возвращает e в заданной степени
ФАКТР	Возвращает факториал числа

Функция	Описание
ОКРВНИЗ	Округляет число в меньшую сторону
НОД*	Возвращает наибольший общий делитель
ЦЕЛОЕ	Округляет число в меньшую сторону до ближайшего целого
НОК*	Возвращает наименьший общий множитель
LN	Возвращает натуральный логарифм числа
LOG	Возвращает логарифм числа по определенному основанию
LOG10	Возвращает логарифм числа по основанию 10
МОПРЕД	Возвращает определитель матрицы массива
МОБР	Возвращает обратную матрицу массива
МУМНОЖ	Возвращает результат умножения двух массивов
ОСТАТ	Возвращает остаток от деления
ОКРУГЛТ*	Возвращает число, округленное до указанного множителя
МУЛЬТИНОМ*	Возвращает многомерный набор чисел
НЕЧЁТ	Округляет число до ближайшего нечетного числа
ПИ	Возвращает число пи
СТЕПЕНЬ	Возвращает результат возведенного в степень числа
ПРОИЗВЕД	Умножает аргументы
ЧАСТНОЕ*	Возвращает целую часть после деления
РАDIАНЫ	Преобразует градусы в радианы
СЛУЧИС	Возвращает случайное число между 0 и 1
СЛУЧМЕЖДУ*	Возвращает случайное число между указанными числами
РИМСКОЕ	Преобразует арабские числа в римские в тексте
ОКРУГЛ	Округляет число до определенного количества цифр
УКРУГЛВНИЗ	Округляет число в меньшую сторону
ОКРУГЛВВЕРХ	Округляет число в большую сторону
РЯД.СУММ*	Возвращает сумму степенного ряда, заданную формулой
ЗНАК	Возвращает знак числа
SIN	Возвращает синус данного угла
SINH	Возвращает гиперболический синус числа
КОРЕНЬ	Возвращает положительный квадратный корень
КОРЕНЬПИ*	Возвращает квадратный корень от π
ПРОМЕЖУТОЧ- НЫЕ.ИТОГИ	Возвращает подсумму списка или базы данных
СУММ	Суммирует аргументы
СУММЕСЛИ	Суммирует ячейки, заданные определенным критерием
СУММПРОИЗВ	Возвращает сумму результата соответствующих компонентов массива
СУММКВ	Возвращает сумму квадратов аргументов

Функция	Описание
СУММРАЗНКВ	Возвращает сумму разности квадратов соответствующих значений в двух массивах
СУММСУММКВ	Возвращает сумму суммы квадратов соответствующих значений двух массивов
СУММКВРАЗН	Возвращает сумму квадратов разностей соответствующих значений двух массивов
TAN	Возвращает тангенс числа
TANH	Возвращает гиперболический тангенс числа
ОТВР	Урезает число до целого

* Доступны только при установленной надстройке “Пакет анализа”.

В табл. Б.9 перечислены функции статистики Excel.

Таблица Б.9. Статистические функции

Функция	Описание
СРОТКЛ	Возвращает среднее от абсолютных отклонений в данных, определенных пользователем
СРЗНАЧ	Возвращает среднее из аргументов
СРЗНАЧА	Возвращает среднее из аргументов и учитывает текстовые и логические значения
БЕТАРАСП	Возвращает интегральную функцию плотности бета-вероятности
БЕТАОБР	Возвращает обратную функцию к интегральной функции плотности бета-вероятности
БИНОМРАСП	Возвращает отдельное значение биномиального распределения
ХИ2РАСП	Возвращает одностороннюю вероятность распределения хи-квадрат
ХИ2ОБР	Возвращает обратное значение односторонней вероятности распределения хи-квадрат
ХИ2ТЕСТ	Возвращает проверку на независимость
ДОВЕРИТ	Возвращает доверительный интервал для совокупности значений
КОРРЕЛ	Возвращает коэффициент корреляции между двумя наборами данных
СЧЁТ	Вычисляет, сколько чисел в списке аргументов
СЧЁТЗ	Вычисляет, сколько значений в списке аргументов
СЧЁТЕСЛИ	Вычисляет количество ячеек, которые удовлетворяют заданному в аргументе критерию
КОВАР	Возвращает ковариацию — среднее двух результатов парного отклонения
КРИТБИНОМ	Возвращает наименьшее значение, для которого кумулятивное биномиальное распределение меньше или равно значению, заданному в критерии
КВАДРОТКЛ	Возвращает сумму квадратов отклонений
ЭКСПРАСП	Возвращает экспоненциальное распределение
FРАСП	Возвращает распределение F-вероятности
FРАСПОБР	Возвращает обратное распределение F-вероятности
ФИШЕР	Возвращает преобразование Фишера
ФИШЕРОБР	Возвращает обратное преобразование Фишера
ПРЕДСКАЗ	Возвращает значение по линейной тенденции
ЧАСТОТА	Возвращает частоту распределения в виде вертикального массива

Функция	Описание
ФТЕСТ	Возвращает результат F-теста
ГАММАРАСП	Возвращает гамма-распределение
ГАММАОБР	Возвращает обратное гамма-распределение
ГАММАНЛОГ	Возвращает натуральный логарифм гамма-функции
СРГЕОМ	Возвращает среднее геометрическое значение
РОСТ	Возвращает значения по экспоненциальному тренду
СРГАРМОН	Возвращает среднее гармоническое значение
ГИПЕРГЕОМЕТ	Возвращает гипергеометрическое распределение
ОТРЕЗОК	Возвращает отрезок линии регрессии
ЭКСЦЕСС	Возвращает критическое значение в наборе данных
НАИБОЛЬШИЙ	Возвращает k-ое наибольшее значение в наборе данных
ЛИНЕЙН	Возвращает параметры линейного тренда
ЛГРФПРИБЛ	Возвращает параметры экспоненциального тренда
ЛОГНОРМОБР	Возвращает инверсию логарифмически нормального распределения
ЛОГНОРМРАСП	Возвращает интегральное логарифмическое нормальное распределение
МАКС	Возвращает максимальное значение из списка аргументов, пропуская логические и текстовые значения
МАКСА	Возвращает максимальное значение из списка аргументов, учитывая логические и текстовые значения
МЕДИАНА	Возвращает медиану из заданных значений
МИН	Возвращает минимальное значение из списка аргументов, пропуская логические и текстовые значения
МИНА	Возвращает минимальное значение из списка аргументов, учитывая логические и текстовые значения
МОДА	Возвращает наиболее частое значение в наборе данных
ОТРЕБНОРРАСП	Возвращает отрицательное биномиальное распределение
НОРМРАСП	Возвращает нормальное интегральное распределение
НОРМОБР	Возвращает инверсию нормального интегрального распределения
НОРМСТРАСП	Возвращает стандартное нормальное интегральное распределение
НОРМСТОБР	Возвращает инверсию стандартного нормального интегрального распределения
ПИРСОН	Возвращает коэффициент корреляции Пирсона
ПРОЦЕНТРАНГ	Возвращает процентную ставку значения в наборе данных
ПЕРЕСТ	Возвращает количество перестановок данного числа объектов
ПУАССОН	Возвращает распределение Пуассона
ВЕРОЯТНОСТЬ	Возвращает вероятность попадания значения в диапазон между двумя числами
КВАРТИЛЬ	Возвращает квартиль набора данных
РАНГ	Возвращает ранг числа в списке чисел
КВПИРСОН	Возвращает квадрат корреляции коэффициента Пирсона

Функция	Описание
СКОС	Возвращает асимметрию распределения
НАКЛОН	Возвращает наклон линейной регрессии
НАИМЕНЬШИЙЦ	Возвращает k-ое наименьшее значение набора данных
НОРМАЛИЗАЦИЯ	Возвращает нормированное значение
СТАНДОТКЛОН	Вычисляет стандартное отклонение на основе выборки, пропуская текстовые и логические значения
СТАНДОТКЛОНА	Вычисляет стандартное отклонение на основе выборки, учитывая текстовые и логические значения
СТАНДОТКЛОНП	Вычисляет стандартное отклонение по генеральной совокупности
СТАНДОТКЛОНПА	Вычисляет стандартное отклонение по генеральной совокупности, включая числа, текст и логические значения
СТОШУХ	Возвращает стандартную ошибку заданного у-значения для каждого x в регрессии
СТЬЮДРАСП	Возвращает распределение Стьюдента
СТЬЮДРАСПОВР	Возвращает обратное распределение Стьюдента
ТЕНДЕНЦИЯ	Возвращает значения линейного тренда
УРЕЗСРЕДНЕЕ	Возвращает среднее внутренности множества данных
ТТЕСТ	Возвращает вероятность, соответствующую коэффициенту Стьюдента
ДИСП	Вычисляет дисперсию на основе выборки, пропуская логические значения и текст
ДИСПА	Вычисляет дисперсию на основе выборки, учитывая логические значения и текст
ДИСПР	Вычисляет дисперсию на основе всей совокупности, пропуская логические значения и текст
ДИСПРА	Вычисляет дисперсию на основе всей совокупности, учитывая логические значения и текст
ВЕЙБУЛЛ	Возвращает распределение Вейбулла
ZТЕСТ	Возвращает двустороннее P-значение z-теста

В табл. Б.10 приведены функции управления текстом в Excel.

Таблица Б.10. Текстовые функции

Функция	Описание
СИМВОЛ	Возвращает символ, указанный номером кода
ПЕЧСИМВ	Удаляет в тексте все непечатаемые символы
КОДСИМВ	Возвращает цифровой код первого символа в текстовой строке
СЦЕПИТЬ	Объединяет несколько элементов текста в один текстовый элемент
РУБЛЬ	Преобразует число в текст с помощью денежного формата
СОВПАД	Проверяет, совпадают ли два текстовых значения
НАЙТИ	Находит одно текстовое значение в другом (чувствителен к регистру)
ФИКСИРОВАННЫЙ	Форматирует число как текст с фиксированным количеством десятичных цифр
ЛЕВСИМВ	Возвращает самый левый символ из текстового значения

Функция	Описание
ДЛСТР	Возвращает количество символов в текстовой строке
СТРОЧН	Преобразует текст в нижний регистр
ПСТР	Возвращает определенное количество символов из текстовой строки, начиная с указанного места
ПРОПНАЧ	Пишет первую букву каждого слова текста в верхнем регистре
ЗАМЕНИТЬ	Заменяет символы в тексте
ПОВТОР	Повторяет текст заданное количество раз
ПРАВСИМВ	Возвращает самый правый символ текстового значения
ПОИСК	Находит одно текстовое значение в другом (не чувствителен к регистру)
ПОДСТАВИТЬ	Заменяет старый текст новым в текстовой строке
Т	Преобразует аргументы в текст
ТЕСКТ	Форматирует числа и Преобразует их в текст
СЖПРОБЕЛЫ	Удаляет пробелы из текста
ПРОПИСН	Преобразует текст в верхний регистр
ЗНАЧЕН	Преобразует аргумент текста в число

Пользовательские форматы чисел

Большой заслугой Excel является возможность создавать пользовательские форматы числа. Хотя в Excel предлагает значительное разнообразие встроенных форматов числа, ни один из них вам может и не подойти. В этом приложении описано, как создать собственный формат числа.

О форматировании числа

По умолчанию во всех ячейках используется общий числовой формат. Часто этот формат интерпретируется как “что вводится, то и получается”. Но если ячейка не достаточно широкая, чтобы вместить все число, то формат **Общий** округляет его, используя десятичную запятую и экспоненциальную запись значения. В большинстве случаев ячейку следует отформатировать с использованием другого формата чисел.

При форматировании чисел главное помнить, что оно влияет только на *отображение* числа в ячейке. Само число остается неизменным, и в формулах, использующих форматированное число, берется его реальное значение.



Исключения из этого правила возникают, если активизировать параметр **Точность** как на экране на вкладке **Вычисления** диалогового окна **Параметры**. Если этот параметр активизирован, то в формулах используются значения, реально отображенные в ячейках.

Автоматическое форматирование цифр

Excel достаточно умна, чтобы выполнить некоторое форматирование автоматически. Например, если в ячейку ввести 12,3%, то Excel знает, что нужно применять процентный формат, и автоматически изменяет форматирование. А если перед числом ввести знак доллара, Excel отформатирует ячейку денежным форматом (в английской версии).



В Excel 2000 используется параметр отключения автоматического форматирования процентных значений. Выберите **Сервис**⇒**Параметры** и перейдите на вкладку **Правка**. Если напротив параметра **Автоматический ввод процентов** выставлен флажок, то обычные вводимые значения будут отображаться в виде процентов (например, введите 12,5 и получится 12,5%). Если этот параметр не активизирован, то значения нужно вводить в виде десятичных чисел (например, ,125 для 12,5%).

Excel автоматически применяет встроенный формат цифр в ячейках, основываясь на следующих критериях:

- ◆ Если число содержит косую (/), его можно преобразовать в формат даты или дробей.
- ◆ Если число содержит дефис (-), его можно преобразовать в формат даты.

- ◆ Если число содержит двоеточие (:) или после числа стоит пробел или буква А или Р, то его можно преобразовать в формат времени.
- ◆ Если число содержит букву Е (в верхнем или нижнем регистре), то его можно преобразовать в экспоненциальное представление.



Чтобы отменить автоматическое форматирование числа при вводе значения, задайте заранее желаемый формат ячейки или поставьте перед числом апостроф (вы получите текстовый формат).

Форматирование чисел с помощью кнопок панели инструментов

На панели инструментов Форматирование находятся кнопки, с помощью которых можно быстро изменить форматы чисел. При щелчке на одной из этих кнопок к выбранному ячейкам применяется определенный формат цифр. В табл. В.1 приведены форматы, активизируемые кнопками панели инструментов Форматирование.



Эти пять кнопок панели инструментов применяют заранее определенные стили в выделенных ячейках. Список стилей в Excel можно открыть, выбрав команду Формат⇒Стиль.

Таблица В.1. Кнопки применения стилей на панели инструментов Форматирование

Кнопка	Описание
Денежный формат	Добавляет знак р в конце числа, разделяет тысячи пробелом и отображает два значения после десятичной запятой
Процентный формат	Отображает значение в виде процентов без десятичной запятой
Формат с разделителями	Разделяет тысячи пробелами и отображает два значения после десятичной запятой
Увеличить разрядность	Увеличивает количество цифр справа от десятичной запятой на одну
Уменьшить разрядность	Уменьшает количество цифр справа от десятичной запятой на одну

Использование клавиатурных сокращений для форматирования чисел

Еще один способ применять форматирование к числам — это использование клавиатурных сокращений. В табл. В.2 приведены клавиатурные сокращения, которые используются для применения форматирования к выделенному диапазону ячеек.

Таблица В.2. Клавиатурные сокращения для форматирования цифр

Комбинация клавиш	Форматирование
<Ctrl+Shift+`>	Общий формат цифр (т.е. неформатированные значения)
<Ctrl+Shift+\$>	Денежный формат с двумя значениями после запятой (отрицательные значения заключаются в скобки)
<Ctrl+Shift+%>	Процентный формат без значений после запятой
<Ctrl+Shift+^>	Экспоненциальное представление чисел без значений после десятичной запятой
<Ctrl+Shift+#>	Формат даты с днем, месяцем и годом
<Ctrl+Shift+@>	Формат времени с часами и минутами
<Ctrl+Shift+!>	Точность до сотых, разделитель тысяч и знак минус перед отрицательными значениями

Использование диалогового окна Формат ячеек для форматирования чисел

Оптимальный способ форматировать ячейки — это использование диалогового окна Форматирование ячеек и вкладки Число. Выделите ячейки, формат которых нужно изменить, и выберите Формат⇒Ячейки. На вкладке Число диалогового окна Формат ячейки задаются 12 категорий форматов чисел. При выборе в списке категории правая сторона окна изменяется и отображает параметры выбранной категории. Например, на рис. В.1 показано, как выглядит диалоговое окно при выборе категории Денежный и знака \$ в поле Обозначение.

Ниже приведено описание категорий форматов чисел:

- ♦ **Общий.** Применяется по умолчанию; отображает числа в виде целых, дробных или использует экспоненциальное представление, если значения слишком длинные и не помещаются в ячейке.
- ♦ **Числовой.** Позволяет задавать количество цифр между пробелами при разделении тысяч и указывать вид отрицательных чисел.
- ♦ **Денежный.** Позволяет задавать количество значений после запятой, выбирать денежный знак и отображать отрицательные числа. В этом формате всегда используется разделитель тысяч (например, пробел).
- ♦ **Финансовый.** В отличие от формата Денежный, денежные символы всегда выровнены по вертикали.
- ♦ **Дата.** Позволяет выбирать формат даты. Excel 2002 также позволяет указывать язык формата даты.
- ♦ **Время.** Позволяет выбирать формат времени. Excel 2002 также предоставляет возможность выбирать язык формата времени.

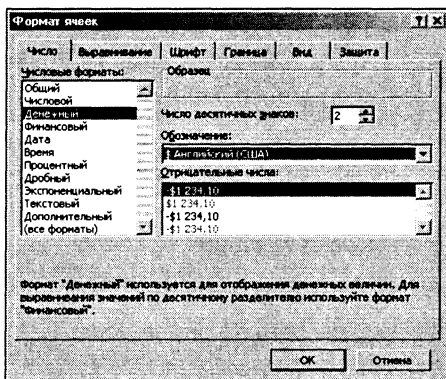


Рис. В.1. Параметры категории Денежный

- ◆ **Процентный.** Позволяет выбирать количество значений после запятой; всегда отображает знак процента.
- ◆ **Дробный.** Предлагает выбрать один из девяти форматов дробей.
- ◆ **Экспоненциальный.** Отображает числа в экспоненциальном представлении (с E): $2,00E = 200\ 000$, $2,05E+05 = 205\ 000$. Также можно выбрать количество значений после запятой слева от E.
- ◆ **Текстовый.** Применение этого формата сообщает Excel о необходимости обрабатывать значение как текст (даже если оно выглядит как число). Этот формат используется при нумерации.
- ◆ **Дополнительный.** Содержит дополнительные форматы чисел. Список изменяется в зависимости от выбранного языка (местоположения). Для Английский (США) предлагаются форматы Zip Code, Zip Code +4, Phone Number и Social Security Number.
- ◆ **Все форматы.** Позволяет определить пользовательский формат числа, которого нет ни в одной из выше перечисленных категорий.



Если в ячейке отображаются знаки решетки (#####), то обычно это значит, что столбец не достаточно широк, чтобы отобразить значение с использованием выбранного формата. Сделайте шире столбец или измените формат ячейки. Символы решетки также могут значить, что в ячейке содержится неверное время или дата.

Форматирование чисел на диаграммах

При создании диаграммы форматирование чисел на ней связано с ячейками рабочего листа, в которых заданы используемые данные. Например, в значениях, отображенных на осях или надписях диаграммы, используется тот же формат данных, что и у значений, на основе которых эта диаграмма создана. При необходимости, формат чисел (в том числе пользовательский формат числа) можно применить к любым значениям на диаграмме.

На любом значении в диаграмме можно дважды щелкнуть мышью. При этом откроется соответствующее диалоговое окно **Формат оси**. Перейдите на вкладку **Число** и укажите нужный формат числа. Можно воспользоваться одним из встроенных форматов или применить пользовательский.

Чтобы переопределить связи между форматами числа на диаграмме и рабочем листе, выставьте флажок опции **Связь с источником** на вкладке **Число** диалогового окна **Формат оси**.

Создание пользовательского формата числа

На рис. В.2 показано, как выглядит диалоговое окно **Формат ячейки** при выборе категории **Все форматы**. Эта категория позволяет создавать формат числа, которого нет ни в одной другой категории. Как видно на рисунке, Excel позволяет создать много самых разнообразных форматов числа.



Пользовательские форматы числа сохраняются вместе с рабочим листом. Чтобы пользовательский формат был доступен в других рабочих книгах, скопируйте ячейку, в которой используется пользовательский формат, в другую рабочую книгу.



Чтобы форматирование ячеек применялось автоматически (как цвет текста или фона) в зависимости от содержимого ячейки, воспользуйтесь средством Условное форматирование (доступное в Excel 97 и выше). Об этом средстве рассказывается в главе 19.

Предварительное форматирование ячейки

Обычно формат числа применяется к ячейке, уже содержащей значение. Но ячейки также можно отформатировать и *перед* вводом значения. Тогда, после ввода информации, значения принимают указанное форматирование. Отформатировать заранее можно определенные ячейки, целые строки или столбцы или весь рабочий лист.

Однако, чтобы не форматировать весь рабочий лист, можно изменить формат числа в стиле Обычный (если не указан другой стиль, то для всех ячеек используется стиль Обычный). Измените стиль, выбрав команду **Формат**⇒**Стиль**. В диалоговом окне **Стиль** щелкните на кнопке **Изменить** и выберите новый формат числа, который будет использоваться в стиле Обычный.

Коды пользовательского формата числа

В табл. В.3 перечислены коды форматирования, доступные для пользовательских форматов, и их краткое описание. Большинство этих кодов используется в примерах, приведенных далее в этом приложении.

Таблица В.3. Коды, используемые для создания пользовательских форматов чисел

Код	Описание
Основной	Отображает число в формате Общий
#	Заполнитель цифры
0 (ноль)	Заполнитель цифры
?	Заполнитель цифры
,	Десятичная запятая
%	Проценты
пробел	Разделитель тысяч
E- E+ e- e+	Экспоненциальное представление чисел
\$ - + / () : пробел	Отображает этот знак
\	Отображает следующий символ
*	Повторяет следующий символ, чтобы заполнить ширину столбца
_ (подчеркивание)	Оставляет отступ, равный ширине следующего символа
"text"	Отображает текст внутри двойных кавычек
@	Заполнитель текста
[цвет]	Отображает символы в заданном цвете
[цвет n]	Отображает соответствующий цвет на палитре цветов, где n – номер от 0 до 56
[условие]	Позволяет устанавливать собственный критерий для каждого раздела в формате числа

В табл. В.4 перечислены коды, используемые для создания пользовательских форматов для дат и времени.

Таблица В.4. Коды, используемые при создании пользовательских форматов даты и времени

Код	Описание
М	Отображает месяц в виде числа без нуля впереди (1-12)
ММ	Отображает месяц в виде числа с нулем впереди (01-12)
МММ	Отображает месяц в виде аббревиатуры (Янв-Дек)
ММММ	Отображает полное название месяца (Январь-Декабрь)
МММММ	Отображает первую букву месяца (Я-Д)
Д	Отображает день в виде числа без нуля впереди (1-31)
ДД	Отображает день в виде числа с нулем впереди (01-31)
ДДД	Отображает день в виде аббревиатуры (Пн-Вс)
ДДДД	Отображает полное название дня (Понедельник-Воскресенье)
ГГ или ГГГГ	Отображает год в виде двухзначного (00-99) или в виде четырехзначного числа (1900-9999)
ч или чч	Отображает час в виде числа без нуля впереди (0-23) или с нулем впереди (00-23)
м или мм	Отображает минуты в виде числа без нуля впереди (0-59) или с нулем впереди (00-59)
с или сс	Отображает секунды в виде числа без нуля впереди (0-59) или с нулем впереди (00-59)
[]	Отображает часы, больше 24, или минуты и секунды, больше 60
AM/PM	Отображает часы с использованием 12-часовых часов; если индикатор AM/PM не используется, то в часах используется 24-часовые часы.

Откуда взялись такие форматы числа?

Excel может создавать пользовательские форматы числа без ведома пользователя. При использовании кнопок Увеличить отступ или Уменьшить отступ на панели инструментов Форматирование создается новый формат числа, который отображается на вкладке Число диалогового окна Формат ячейки. (Чтобы открыть это диалоговое окно, выберите команду Ячейки в меню Формат). Например, если щелкнуть на кнопке Увеличить отступ пять раз, то создадутся такие пользовательские форматы числа:

0,0
 0,000
 0,0000
 0,000000

Пользовательский формат для двух значений после запятой не создается, поскольку он является встроенным.

Примеры пользовательских форматов

В этом приложении приведены примеры пользовательских форматов. Большинство этих форматов можно почерпнуть из приведенных ниже примеров. Но в некоторых случаях необходимо внести незначительные изменения, чтобы представленные форматы удовлетворяли соответствующим требованиям.

Масштаб

Пользовательский формат числа можно использовать для масштабирования числа. Например, при работе с очень большими числами иногда нужно отобразить числа в тысячах, а не единицах (т.е. отобразить число 1 000 000 как 1 000). В расчетах, конечно же, будет использоваться реальное число, которое указано в ячейке. Форматирование влияет только на вид этого числа.

Отображение значений в тысячах

Следующая строка формата отображает значения без последних трех цифр слева от десятичной запятой и удаляет саму запятую. Другими словами, значение выглядит так, как после деления на 1 000 и округления до целого.

, ###пробел

А вот вариант этой строки формата. Значение с этим форматом номера выглядит, как после деления на 1 000 и округления до двух значений после десятичной запятой.

, 00пробел

В табл. В.5 показан пример этих форматов:

Таблица В.5. Примеры отображения значений в тысячах

Значение	Формат числа	Отображение
123456	# ###пробел	123
1234565	# ###пробел	1,235
-323434	# ###пробел	-323
123123,123	# ###пробел	123
499	# ###пробел	(пусто)
500	# ###пробел	1
123456	# ### , 00пробел	123,46
1234565	# ### , 00пробел	1,234,57
-323434	# ### , 00пробел	-323,43
123123,123	# ### , 00пробел	123,12
499	# ### , 00пробел	,50
500	# ### , 00пробел	,50

Отображение значений в сотнях

Приведенная ниже строка формата отображает значения в сотнях с двумя значениями после десятичной запятой. Значение в этом формате выглядит так, как будто оно разделено на 100 и округлено до двух цифр после запятой.

0" , "00

В табл. В.6 перечислены примеры применения этого формата:

Таблица В.6. Примеры отображения значения в сотнях

Значение	Формат числа	Отображение
546	0", "00	5,46
100	0", "00	1,00
9890	0", "00	98,90
500	0", "00	5,00
-500	0", "00	-5,00
0	0", "00	0,00

Отображение значений в миллионах

Следующая строка формата отображает значения в миллионах без десятичной запятой. Значение в этом формате выглядит деленным на 1 000 000 и округленным до целого.

###пробелпробел

Ниже приведен вариант этой строки с десятичными значениями. Значение выглядит поделенным на 1 000 000 и округленным до двух цифр после запятой.

###, 00пробелпробел

Еще один вариант. В этом случае добавляется буква М в конец значения.

М

Следующая строка формата немного сложнее. Она позволяет добавлять букву М в конец значения и отображает отрицательные значения и нули в скобках.

###, 0 "М"_) ; (# ###, 0 "М) " ; 0, 0"М"_)

В табл. В.7 приведены примеры этих строк форматов.

Таблица В.7. Примеры отображения значений в миллионах

Значение	Формат числа	Отображение
123456789	# ###пробелпробел	123
1,234457E+11	# ###пробелпробел	123 457
1000000	# ###пробелпробел	1
5000000	# ###пробелпробел	5
-5000000	# ###пробелпробел	-5
0	# ###пробелпробел	(пусто)
123456789	# ###, 00пробелпробел	123,46
1,234457E+11	# ###, 00пробелпробел	123,456.7
1000000	# ###, 00пробелпробел	1.00
5000000	# ###, 00пробелпробел	5,00
-5000000	# ###, 00пробелпробел	-5,00
0	# ###, 00пробелпробел	.00
123456789	# ### "М"	123М

Значение	Формат числа	Отображение
1.234457E+11	# ### "М"	123 457M
1000000	# ### "М"	1M
5000000	# ### "М"	5M
-5000000	# ### "М"	-5M
0	# ### "М"	M
123456789	# ###,0 "М"_);# ###, 0 "М)";0,0"М"_)	123,5M
1,234457E+11	# ###,0 "М"_);# ###, 0 "М)";0,0"М"_)	123 456,8M
1000000	# ###,0 "М"_);# ###, 0 "М)";0,0"М"_)	1,0M
5000000	# ###,0 "М"_);# ###, 0 "М)";0,0"М"_)	5,0M
-5000000	# ###,0 "М"_);# ###, 0 "М)";0,0"М"_)	(5,0M)
0	# ###,0 "М"_);# ###, 0 "М)";0,0"М"_)	0,0M

Добавление в значение нулей

Следующая строка формата отображает значение с тремя дополнительными нулями и без десятичной запятой. Значение в этом формате выглядит округленным до целого числа и умноженным на 1 000.

#" 000"

В табл. В.8 показаны примеры этого формата и варианты добавления шести нулей.

Таблица В.8. Примеры отображения значений с дополнительными нулями

Значение	Формат числа	Отображение
1	#" 000"	1 000
1,5	#" 000"	2 000
43	#" 000"	43 000
-54	#" 000"	-54 000
5,5	#" 000"	6 000
0,5	#" 000 000"	1 000 000
0	#" 000 000"	000 000
1	#" 000 000"	1 000 000
1,5	#" 000 000"	2 000 000
43	#" 000 000"	43 000 000
-54	#" 000 000"	-54 000 000
5,5	#" 000 000"	6 000 000
0,5	#" 000 000"	1 000 000

Скрытие нуля

В следующей строке формата третий элемент строки пустой; это обуславливает отображение пустых ячеек с нулевыми значениями:

```
Основной;Основной; ;@
```

В этой строке формата используется формат Основной для положительных и отрицательных значений. При необходимости его можно заменить на другой код формата.

Отображение нулей впереди

Чтобы отобразить нули впереди, создайте пользовательский формат числа, в котором используется символ 0. Например, чтобы все числа отображались 10 цифрами, воспользуйтесь приведенной ниже строкой формата числа. Значения с количеством цифр, меньшим 10, отображаются с нулем впереди.

```
0000000000
```

Можно также сделать, чтобы все числа отображались с фиксированным количеством нулей впереди. Ниже приведена строка, которая добавляет три нуля в начало каждого числа:

```
"000"#
```

В следующем примере в строке формата используется повторяющийся код для символа (звездочки), чтобы заполнить нулями впереди всю ширину ячейки:

```
*00
```

Форматирование процентных значений

Использование символа процентов (%) в строке формата приводит к тому, что содержимое ячейки отображается в процентном формате. Обратите внимание, что знак процента также отображается в строке формулы.

Следующая строка форматирует значения, которые меньше либо равны 1, форматом Процентный. Значения больше 1 и текст форматируются форматом Общий.

```
[<=1]0,00%;Основной
```

Если в столбце смешаны ячейки с процентным и обычным форматированием, то можно представить непроцентные значения с отступом справа так, чтобы они были выровнены. Для этого следует применить следующий формат к непроцентным ячейкам. В этой строке формата используется подчеркивание с символом процента. Результатом является пробел, равный ширине символа процента.

```
#,00_%
```

Отображение дробей

Excel поддерживает встроенный формат дробных чисел (выберите категорию Дробный). Например, чтобы отобразить значение ,125 в виде дроби с знаменателем 8, выберите в списке Тип параметр Восьмыми долями (4/8).

Пользовательский формат можно использовать для создания других форматов дробей. Например, следующая строка формата отображает значение в 50-ых частях от целого:

```
# " ??/50
```

Следующая строка формата отображает значение в виде дробного доллара. Например, значение 154,87 отобразится как *154 and 87/100 Dollars*.

0 "and "??/100 "Dollars"

В следующем примере представлено значение в шестнадцатеричном виде с кавычками справа. Эта строка формата используется при работе с дюймами (например, 2/16").

" ??/16\"

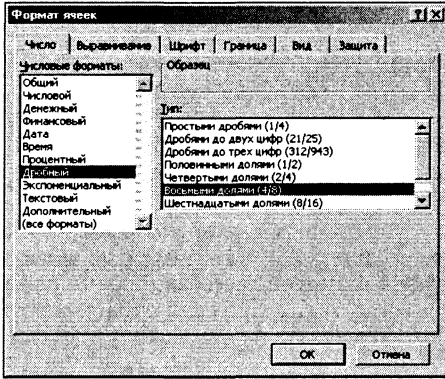


Рис. В.3. Выбор формата числа для отображения значения в виде дроби

	A	B	C	D
1	Value	Number Format	Display	
2	1.5	0.00_-0.00-	1.50	
3	0	0.00_-0.00-	0.00	
4	-3.4	0.00_-0.00-	3.40-	
5	9	0.00_-0.00-	9.00	
6	-0.5	0.00_-0.00-	0.50-	
7	-2.5	0.00_-0.00-	2.50-	
8	-2.25	0.00_-0.00-	2.25-	
9	0	0.00_-0.00-	0.00	
10	5.8	0.00_-0.00-	5.80	
11	-5.8	0.00_-0.00-	5.80-	
12	2	0.00_-0.00-	2.00	
13				

Рис. В.4. Использование пользовательского формата числа, который задает знак минус справа

Отображение текста в кавычках

В следующей строке формата отображаются числа, вокруг которых стоит текст в кавычках:

Основной; Основной; Основной; "@"

Повтор текста

Следующая строка формата числа отображает содержимое ячейки три раза. Например, если в ячейке содержится текст *Budget*, то в ячейке отобразится *Budget Budget Budget*.

;; @@@

Проверка пользовательских форматов числа

При создании пользовательского формата числа не пропустите окно Образец на вкладке Число диалогового окна Формат ячеек. В этом окне отображается значение активной ячейки с использованием строки формата в окне Тип.

Хорошей идеей является проверить пользовательские форматы числа с помощью следующих данных: положительного значения, отрицательного значения, нуля и текста. Часто создание пользовательского формата числа происходит в несколько приемов. Каждый раз при редактировании строки формата она добавляется в список. Когда, наконец, получена правильная строка формата, откройте диалоговое окно Формат ячеек еще раз и удалите все предыдущие попытки.

Отображение знака минуса справа

Приведенная ниже строка формата отображает отрицательные значения со знаком минус справа от числа. К положительным значениям справа добавляется пробел, поэтому и положительное, и отрицательное значения выровнены аккуратно по правому краю.

0,00_- ; 0,00_

На рис. В.4 показана строка формата.

Условное форматирование числа

Условное форматирование применяется тогда, когда необходимо вид данных изменять в зависимости от содержимого ячейки. Средство Условное форматирование в Excel предлагает наиболее эффективный способ выполнить условное форматирование; кроме него также можно воспользоваться пользовательскими форматами чисел.



В условном форматировании можно указать до трех условий — два из них явные, а третье — неявно. Условия заключаются в квадратные скобки и состоят из простых числовых сравнений.

В следующей строке формат изменяется в зависимости от значения в ячейке. Эта строка формата обычно разделяет числа на три группы: меньшие либо равные 4, большие либо равные 8 и остальные. На рис. В.5 показан пример такой строки формата.

[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0

1	A	B	C	D
2	Value	Number Format	Display	
3	1	[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0	Low	1
4	2	[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0	Low	2
5	3	[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0	Low	3
6	4	[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0	Low	4
7	5	[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0	Medium	5
8	6	[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0	Medium	6
9	7	[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0	Medium	7
10	8	[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0	High	8
11	9	[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0	High	9
12	10	[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0	High	10
13	11	[<=4]"Low"* 0; [>=8]"High"* 0;"Medium"* 0	High	11

Рис. В.5. В ячейках в столбце С используется условный формат числа

Следующая строка формата отображает значения меньше 1 с символом евро справа. В противном случае значения отображаются со знаком доллара слева.

[<1],00 ;\$0,00_

Следующий формат числа используется для телефонных номеров. Значения больше 9999999 (т.е. номера с кодами городов) отображаются в виде (xxx) xxx-xxxx. Другие значения (номера без кодов городов) отображаются xxx-xxxx.

[>9999999](000) 000-0000;000-0000

Для почтовых кодов можно воспользоваться следующей строкой формата. Она отображает код в виде пятизначного числа. Но если число больше 99999, то используется формат "плюс четыре" (xxxxx-xxxx).

[>99999] 00000-00000;00000

Выделение цветом

Строки пользовательских форматов могут отображать содержимое ячейки различными цветами. Скажем, приведенный ниже формат выделяет положительные значения красным цветом, отрицательные — зеленым, нули — черным, а текст — голубым.

[Красный] Основной; [Зеленый] Основной; [Черный] Основной; [Синий] Основной

В следующем примере показана строка формата, в которой задаются цвета. Положительные значения отображаются как обычно; отрицательные значения и текст отображаются текстом Ошибка! красного цвета.

Основной; [Красный] "Ошибка!"; 0; [Красный] "Ошибка!"

При использовании следующей строки формата значения, которые меньше 2, отображаются красным цветом. Значения больше 4 отображаются зеленым цветом. Все остальное (текст и значения между 2 и 4) отображаются черным цветом.

[Красный] [<2] Основной; [Зеленый] [>4] Основной; [Черный] Основной

Как показано в предыдущих примерах, Excel распознает такие имена цветов, как [Красный] и [Синий]. Также можно использовать другие цвета. Например, следующая строка формата отображает содержимое ячейки с использованием цветов в шестнадцатеричном представлении:

[Цвет16] Основной



Нельзя изменить ячейки с цветами из строки формата на обычные команды форматирования ячеек.

Форматирование дат и времени

Когда в ячейку вводится дата, Excel форматирует ее системным форматом краткой даты. Этот формат можно использовать, задав региональные настройки в папке Панель управления Windows.

Excel предлагает множество встроенных форматов дат и времени. В приведенной ниже таблице перечислены форматы даты и времени, которые полезны при форматировании данных. В первом столбце показан порядковый номер даты/времени.

Значение	Формат числа	Отображение
36676	MMMM Д, ГГГГ (ДДДД)	Май 30, 2000 (вторник)
36676	"Сегодня" ДДДД!	Сегодня вторник!
36676	ДДДД, ММ/ДД/ГГ	вторник, 05/30/00
36676	"Месяц:" МММ	Месяц: май
36676	Основной (М/Д/ГГГГ)	36676 (5/30/2000)
0,345	ч "часов"	8 часов
0,345	Время ч:мм	Время 8:16



Более подробно о системе серийных номеров для дат и времени читайте в главе 6.

Отображение текста с числами

Возможность отображения текста с числами является одним из наиболее важных преимуществ использования пользовательского формата числа. Чтобы добавить текст, просто создайте строку формата числа и поместите текст, заключенный в кавычки. Например, следующая строка формата отображает числовое значение с текстом (*US Dollars*), добавленным в конец:

```
# ##0,00 "(US Dollars)"
```

Ниже приведен пример отображения текста перед числом:

```
"Среднее: "0,00
```

Если используется последний формат числа, то знак минус для отрицательных значений окажется перед текстом. Чтобы отобразить этот знак корректно, следует задать формат так:

```
"Среднее: "0,00;"Среднее: "-0,00
```

Следующая строка формата отображает значение со словами *Dollars and Cents*. Например, число 123,45 отображается в виде *123 Dollars and ,45 Cents*.

```
0 "Dollars and" ,00 "Cents"
```

Отображение нуля с тире

Приведенная ниже строка формата числа отображает нулевые значения в виде числового значения с тире:

```
# ##0,0;- ###0,0;-----
```

Конечно, можно создать много разных вариантов. Например, можно заменить шесть дефисов одним из таких символов:

```
<0>
```

```
-0-
```

```
~~
```

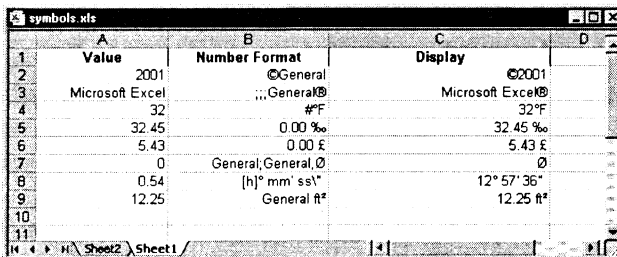
```
<NULL>
```

```
" [NULL] "
```

Помните, что при использовании квадратных скобок их необходимо заключать в кавычки.

Использование специальных символов

В строке формата числа могут содержаться специальные символы, такие как знак авторского права, знак градусов и т.д. На рис. В.6 показаны некоторые специальные символы, используемые в строке формата.



	A	B	C	D
1	Value	Number Format	Display	
2	2001	©General		©2001
3	Microsoft Excel	:::General@		Microsoft Excel@
4	32	#F		32°F
5	32.45	0.00 %		32.45 %
6	5.43	0.00 £		5.43 £
7	0	General;General;@		∅
8	0.54	[h]° mm' ss"		12° 57' 36"
9	12.25	General ft²		12.25 ft²
10				
11				

Рис. В.6. Использование специальных символов в строке формата числа

Чтобы вставить символ, нужно знать клавиатурные сокращения с <Alt+>, необходимые для вставки символов. Например, знак авторского права вставляется при нажатии клавиш <Alt+0169> (для ввода чисел используйте цифровую клавиатуру).



Если вы работаете с Excel 2002, то эти коды можно определить, воспользовавшись командой (Вставка⇒Символ) (рис. В.7). В более ранних версиях Excel воспользуйтесь картой символов Windows.

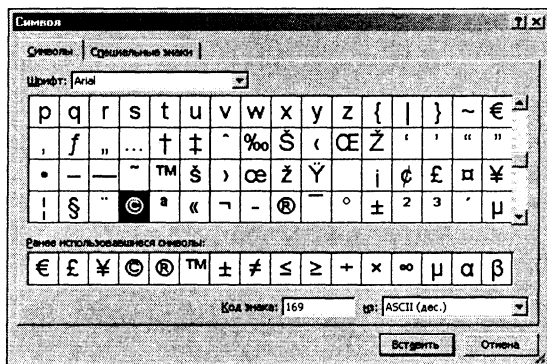


Рис. В.7. Воспользуйтесь диалоговым окном Символ (только в Excel 2002), чтобы определить код специальных символов

Другим использованием специальных символов является отображение символов в верхнем индексе как показателя степени. Например, <Alt+0178> подставляет символ возведения во вторую степень, сочетание <Alt+0179> генерирует символ возведения в третью степень.

Специальные символы можно использовать для отображения символов валют. Например, можно отобразить символ японской йены (<Alt+0165>) или британского фунта (<Alt+0162>).

Устранение определенных типов элементов

Форматирование числа можно использовать, чтобы скрыть определенные типы элементов. Например, следующая строка формата отображает текст, а не значения:

;;

Приведенная ниже строка формата отображает значения, а не текст или нули:

0,0;-0,0;;

Следующая строка формата отображает все, кроме нулей:

0,0;-0,0;;@

Эту строку формата можно использовать, чтобы полностью скрыть содержимое ячейки:

;;;

Обратите внимание, что когда ячейка активизирована, то ее содержимое представляется в строке формул. Наилучший способ скрыть содержимое ячейки — выбрать параметр Скрыть формулы на вкладке Защита диалогового окна Формат ячейки и защитить лист.

Отображение строки формата числа в ячейке

В Excel нет функции рабочего листа, которая отображает формат числа определенной ячейки. Однако такую функцию можно создать при помощи VBA. Вставьте следующую процедуру функции в модуль VBA:

```
Function NumberFormat(cell) As String
' Возвращает числовой формат ячейки
Application.Volatile True
NumberFormat = cell.Range("A1").NumberFormat
End Function
```

Затем создайте такую формулу:

```
=NumberFormat(C4)
```

Эта формула возвращает формат ячейки C4.

Такая функция может использоваться в формулах, которые вычисляют условную сумму. Например, можно создать формулу, которая суммирует только те ячейки, в которых используется определенный формат числа. Более подробно о вычислении условных сумм рассказано в главе 7.



Если в ячейке содержится больше 1 024 символов, то строка формата ; ; ; скрывает ее содержимое.



Более подробно о создании пользовательских функций рабочих листов с помощью VBA рассказано в части VI.

Заполнение ячейки повторяющимися символами

Символ звездочки (*) соответствует повторяющимся символам в строке формата числа. Повторяющийся символ полностью заполняет ячейку и изменяется при изменении ширины столбца. Следующая строка формата, например, отображает содержимое ячейки, заполненной справа от тире:

Основной*-; Основной*-; Основной*-; Основной*-

На рис. В.8 показано несколько примеров строк формата числа, в которых используется звездочка для повторения символа.

	A	B	C
1	Value	NumberFormat	Display
2	45.3	General*;*General*;*General*;*General*	45.3-----
3	-45.43	General*;*General*;*General*;*General*	45.43-----
4	Excel	General*;*General*;*General*;*General*	Excel-----
5	45.3	*.General*;*General*;*General*;*General*	-----45.3
6	-45.43	*.General*;*General*;*General*;*General*	-----45.43
7	Excel	*.General*;*General*;*General*;*General*	-----Excel
8	1434.55	\$\$*##0.00*	\$1,434.55-----
9	1545.98	\$\$*##0.00**	\$1,545.98*****
10	509.54	**	-----
11	Excel	*+	Excel-----
12	545.98	*+	+++++++-----
13	12.83	*\$General	\$\$\$\$\$\$\$\$\$\$\$\$12.83

Рис. В.8. Примеры форматов числа, в которых используется повторяющийся символ

Отображение точек впереди

Ниже приведен пользовательский формат числа, который является вариантом финансового формата. Формат числа отображает знак доллара слева, а значения справа. Пробел между ними заполнен точками.

`-$*.# ##0,00_);_($*.(# ##0,00);_($* "-"??_);_(@_)`

Дополнительные ресурсы Excel

Если вы дочитали книгу до этого приложения, то мою миссию можно считать завершенной. Однако в книге невозможно описать все темы, заслуживающие внимания. Поэтому я составил список дополнительных ресурсов, которые могут вам пригодиться в обучении. Эти ресурсы разделены на три категории: техническая поддержка Microsoft, форумы в Internet и Web-узлы.

Кстати, не забудьте о справочной системе Excel в Internet. Она улучшается с каждой новой версией приложения.

Техническая поддержка Microsoft

Техническая поддержка — обычный термин для справочной информации, предоставляемой разработчиком программного обеспечения. В данном случае имеется в виду информация от первоисточника — Microsoft. Техническая поддержка Microsoft доступна в нескольких различных формах.

Параметры поддержки

Чтобы ознакомиться с параметрами поддержки, выберите команду Справка⇒О программе и в открывшемся диалоговом окне щелкните на кнопке Поддержка. При этом откроется файл справочной системы с перечнем всех параметров, предлагаемых Microsoft, в том числе варианты платной и бесплатной поддержки.

Исходя из своего опыта, я предлагаю вам пользоваться телефонной линией поддержки производителя только в крайнем случае. Шансы быть услышанным есть, но за общение со столь именитым собеседником вам придет огромный телефонный счет (допустим, с этим вы смиритесь), ведь вы проведете много времени у телефона, а проблема может остаться нерешенной.

Все дело в том, что сотрудники компании, дающие советы по телефону, могут ответить только на самые общие вопросы. Ответы на подобные вопросы обычно без труда можно найти и в другом источнике.

База знаний Microsoft

Наилучшая попытка решить проблему — воспользоваться базой знаний Microsoft. Это основной источник информации Microsoft — обширная поисковая база данных, состоящая из десятков тысяч подробных статей с технической информацией, списков ошибок и исправлений и т.п. К базе знаний Microsoft предоставлен бесплатный доступ через Internet по адресу:

<http://search.support.microsoft.com/kb/>

Домашняя страница Microsoft Excel

Официальная домашняя страница Excel находится по адресу:

<http://www.microsoft.com/office/excel>

Инструменты Microsoft Office в Web

Более подробно об Office XP (в том числе и об Excel 2002) читайте на узле <http://office.microsoft.com>

Там же находятся пакеты обновления продукта, надстройки, примеры и много другой полезной информации.



Как известно, Internet является динамической структурой, которая быстро изменяется. Web-узлы часто переорганизуются, поэтому тот или иной URL, приведенный в этом приложении, может быть временно недоступным.

Форумы в Internet

Сеть Usenet (пользовательская сеть) — это служба Internet, предоставляющая доступ к некоторым специальным группам новостей по интересам, в которых можно пообщаться с людьми на определенные темы. Форумы работают по принципу досок с информационными объявлениями. В них можно разместить сообщение или вопрос, и (обычно) несколько человек отвечают на него.

Существуют тысячи форумов практически по всевозможным темам. Обычно на вопросы, размещаемые на форуме, ответ приходит в течение 24 часов — предполагается, что вопрос задан так, что другие участники форума заинтересованы на него ответить.



Помимо доступа в Internet, чтобы войти в форум, нужно специальное программное обеспечение. Хорошим вариантом такого продукта является Microsoft Outlook Express (бесплатный). Этот продукт входит в поставку Internet Explorer.

Форум по электронным таблицам

Основной форум Usenet для пользователей электронных таблиц находится по адресу: `comp.apps.spreadsheets`

Этот форум предназначен только для пользователей любых электронных таблиц, но около 90% сообщений имеют дело с Excel.

Форум Microsoft

Microsoft поддерживает обширный набор форумов, в том числе те, которые посвящены Excel. Если Internet-провайдер не предоставляет доступ к форумам Microsoft, то в них можно попасть прямо с сервера новостей Microsoft. Программное обеспечение для чтения новостей нужно настроить, чтобы зайти на сервер новостей Microsoft по этому адресу:

`msnews.microsoft.com`

В табл. Г.1 перечислены основные форумы по Excel, которые находятся на сервере новостей Microsoft.

Таблица Г.1. Форумы по Excel на сервере Microsoft

Форум	Тема
<code>microsoft.public.excel.programming</code>	Программирование в Excel на VBA или с помощью XLM-макросов
<code>microsoft.public.excel.123quattro</code>	Преобразование листов Lotus 1-2-3 и Quattro Pro в файлы Excel
<code>microsoft.public.excel.worksheet.functions</code>	Функции рабочих листов
<code>microsoft.public.excel.charting</code>	Построение диаграмм в Excel
<code>microsoft.public.excel.printing</code>	Печать в Excel
<code>microsoft.public.excel.queryDAO</code>	Использование Microsoft Query и Data Access Objects (DAO) в Excel
<code>microsoft.public.excel.datamap</code>	Использование средства Data Map в Excel
<code>microsoft.public.excel.crashesGPFs</code>	Справка по ошибкам общей защиты или системным неисправностям
<code>microsoft.public.excel.misc</code>	Общие темы, которые не соответствуют ни одной из категорий
<code>microsoft.public.excel.links</code>	Использование связей в Excel
<code>microsoft.public.excel.macintosh</code>	Проблемы в Excel в ОС Macintosh
<code>microsoft.public.excel.interopoledde</code>	OLE, DDE и проблемы взаимодействия приложений
<code>microsoft.public.excel.setup</code>	Настройка и установка Excel
<code>microsoft.public.excel.templates</code>	Шаблоны популярных электронных таблиц и другие файлы XLT
<code>microsoft.public.excel.sdk</code>	Проблемы использования Software Development Kit для Excel

Поиск по форуму

Многие не знают, как можно выполнить поиск по последним сообщениям на форуме. Часто форум — это самый удобный способ получения ответа на вопрос, поскольку в процессе поиска ответ можно найти почти сразу (если знать, где искать). Наилучшим источником поиска по группам новостей является узел Google.com, который находится по такому адресу: <http://groups.google.com>



Когда-то поиск по форуму осуществлялся на узле Deja.com. Этот узел закрылся, а архивы форумов были выкуплены Google.

Как работает поиск? Предположим, что у вас возникают сложности при определении уникальных значений в диапазоне ячеек. Поиск можно осуществить при помощи ключевых слов:

Excel, диапазон и уникальный. Поиск Google найдет довольно много сообщений на форуме по этой теме. Чтобы просмотреть все найденные сообщения, потребуется не один час, но есть прекрасная возможность найти ответ на поставленный вопрос.

Советы по размещению сообщений на форуме

1. Убедитесь, что на ваш вопрос еще нет ответа. Проверьте FAQ (если они есть) и выполните поиск на Google.com (см. "Поиск по форуму" в этом приложении).
2. Сделайте тему вопроса максимально наглядной. Например на запрос "Помогите мне!" и "Вопрос по Excel" менее вероятно получить ответ, чем на более конкретный вопрос, скажем "Нужна помощь по пользовательским функциям рабочего листа".
3. Укажите продукт и используемую версию. Во многих случаях ответ на вопрос зависит от версии Excel.
4. Сделайте формулировку вопроса как можно более конкретной.
5. Вопрос должен быть коротким и по существу, но достаточно информативным, чтобы на него могли адекватно ответить.
6. Попробуйте сами ответить на ваш вопрос.
7. Разместите вопрос на соответствующем форуме и не дублируйте его на других форумах, кроме случаев, когда вопрос относится к нескольким форумам.
8. Не вводите текст только в верхнем или в нижнем регистре и проверьте грамматику и правописание.
9. Не вкладывайте в сообщение файл.
10. Избегайте размещения сообщений в формате HTML.

Web-узлы

При наличии доступа к World Wide Web (WWW) можно найти много полезных Web-узлов, посвященных работе в Excel. Я перечислю только некоторые из них.

Страница электронных таблиц

Это мой собственный Web-узел, где находятся файлы, которые можно загрузить, советы разработчиков, инструкции по использованию скрытых возможностей Excel, шутки в электронных таблицах, обширные списки ссылок на другие узлы Excel и информация о моей книге. Его адрес:

<http://www.j-walk.com/ss>



На этом узле содержится список ошибок, найденных мной в каждой из моей книг, в том числе в книге, которую я читаю сейчас. (Да, от ошибок никто не застрахован).

Web-ресурс по Excel

Этот узел, созданный Чипом Пирсоном (Chip Pearson), содержит большое количество полезных примеров VBA-кодов и хитроумных формул и находится по адресу:

<http://www.cpearson.com/excel.htm>

Страница по Excel Стефена Баллена

Web-узел Стефена Баллена (Stephen Bullen) содержит несколько интересных примеров использования Excel, в том числе раздел "Говорят, что это нельзя сделать". Адрес такой:
<http://www.bmsltd.co.uk/excel>

Вопросы и ответы

На многих форумах есть раздел FAQ — список часто задаваемых вопросов. Цель такого списка вопросов — предотвратить повторение одних и тех же вопросов. Раздел FAQ форума `сорт. apps . spreadsheets` находится по адресу:

<http://www.faqs.org/faqs/spreadsheets/faq>

Предметный указатель

A

A1, 66
Apple, 34

H

HTML, 37

L

Lotus 1-2-3, 573
Lotus 1-2-3, 34; 112; 240

M

MultiPlan, 34

P

Power Utility Pak, 161; 483

Q

Quattro Pro, 34

R

RIC1, 66

V

VBA, 49; 142; 200; 359; 487
VBA-функция, 497
VisiCalc, 34

X

XLM, 39

A

Абсолютная ссылка, 65
Автозаполнение, 70
Автоисправление формул, 469

Автоматический ввод функций, 116

Автоматическое
обновление рядов данных, 390
применение имен, 90
присвоение имени, 80

Автофильтр, 50; 223

Активная книга, 38

Амортизация, 278

займа с переменной ставкой, 315

Анализ денежных потоков, 268

Анимированная диаграмма, 409

Аппроксимация кривой, 403

Аргумент функции, 111; 501

Б

База данных, 50; 221

Беспроцентный заем, 285

Ближайшее воскресенье, 161

БС, 268

Будущее значение, 268

Быстрое суммирование, 179

В

Введение

даты, 146

кода, 494

Ввод

данных в ячейки, 43

несовместимых дат, 152

формул, 55

формулы массива, 333

Версии Excel, 35

Високосный год, 151; 166

Влияющие ячейки, 478

Вложенная диаграмма, 39

Вложенные

скобки, 63

функции, 113

Внешняя база данных, 50

Возвращение

последнего значения, 353

самого длинного текста, 349

Возраст человека, 159

Восстановление файла, 68

Временной
 пояс, 173
 снимок ячейки, 370
Время дня, 149
Вставка
 в виде рисунка, 380
 имен, 57
 столбца, 92
 строки, 92
Вставки функции, 501
Встроенные функции VBA, 522
Выбор данных из списка, 393
Выделение, 43
 диапазона формулы массива, 333
 уникальных записей списка, 229
Вызов функции из процедуры, 507
Выражение, 113; 520
 присвоения, 520
Вычисление
 дня года, 159
 квартального периода, 166
 количества дней, 156
 количества лет, 159
 периметра, 255
 площади, 255
 праздничных дат, 163
 среднего без учета нулевых значений, 346
 среднего уровня, 212
 точки пересечения с осью Y, 401
 угла наклона, 401
 формул, 64
 бесценивания, 308
Вычисляемое поле, 428
Вычисляемые условия, 236
Вычисляемый элемент, 428

Г

Генерирование
 массива целых чисел, 338
 уникальных случайных чисел, 371
Гистограмма сравнения, 386
График
 выполнения работ, 383
 разброса данных, 387
 тенденции, 399
Графический слой, 46
Группа, 417
Группирование элементов сводной таблицы,
 426
Групповой символ, 232

Д

Дата и время, 145
Двумерный массив, 327; 331
Действительность номера кредитной карточки,
 461
Действия в массиве, 336
День
 года, 159
 недели, 160
Детальный план амортизации, 314
Диаграмма, 39; 47; 48; 377
 Гантта, 384
Диалоговое окно, 41
Диапазон условий, 228
Динамическая сортировка, 357
Динамически именованная формула, 104
Динамические финансовые планы, 311
Добавление
 времени, 173
 модуля VBA, 490
 описания функции, 502
 символов, 132
Достоверность аппроксимации, 403

Е

Единая формула, 328
Ежегодная эффективная ставка, 282

З

Зависимые ячейки, 479
Задачи
 амортизации, 278
 дисконтирования, 275
 пользовательских функций, 498
 прямоугольных треугольников, 253
Засм
 с единообразной ставкой, 284
 с ежегодными выплатами, 285
Замена текста, 135
Запись, 221
Заполнение промежутков, 227
Запуск мегаформулы, 460
Защита данных, 52
Значение, 55
 ошибки, 182

И

- Извлечение имени, 140
 - имени файла, 139
 - непустых ячеек, 357
 - первого слова, 139
 - положительных значений, 356
 - последнего слова, 139
 - символов, 134
- Изменение регистра текста, 134
- Имена в нескольких листах, 83
- Именованная формула, 97
- Импортирование данных, 420
- Имя, 77
 - диаграммы, 95
 - константы, 96
 - столбца, 81
 - строки, 81
 - функции, 500
- Инженерные функции, 120
- Интерактивное отображение данных, 392
- Интуитивно понятные формулы, 87
- Информационные функции, 119
- Исключение нечисловых символов из строки, 353
- Использование
 - дат до 1900 года, 152
 - имен в формулах, 86
 - формул проверки, 451
- Исправление
 - ошибок циклической ссылки, 480
 - циклических ссылок, 366
- История Excel, 33
- Источник данных, 420

К

- Календарь, 358
- Категории функций, 118; 502
- Категория функции,
- Книга, 37
- Код символа, 126
- Коды пользовательского формата числа, 596
- Количество символов в ячейке, 123
- Комбинации клавиш, 42; 592
- Комментарий, 515
- Конкатенация, 129
- Константа, 96; 518
- Конструкция
 - For Each-Next, 531
 - If-Then, 524
 - Select Case, 525

- Контекстное меню, 40
- Копирование
 - VBA-кода, 496
 - сводной таблицы, 427
 - текста из формулы, 455
 - формулы, 67
 - отфильтрованных данных, 226
- Косвенные циклические ссылки, 367

Л

- Линейная
 - интерполяция, 217
 - фильтрация, 406
- Линейное предсказание, 402
- Линейные тенденции, 400
- Лист, 38
- Логарифмическая тенденция, 404
- Логические
 - операторы VBA, 520
 - ошибки, 468; 504
 - функции, 119

М

- Макетирование сводной таблицы, 422
- Макрос, 49
 - XLM, 39
- Массив, 102; 114; 327; 521
 - констант, 329
- Мастер
 - создания формул, 196
 - функций, 111; 115
- Математические функции, 118
- Мегаформула, 453
- Меню, 40
- Многомерный массив, 521
- Множественные условия, 437
- Модуль VBA, 37

Н

- Настройка, 49; 510
- Назначение имени константе, 96
- Накопленная сумма, 299
- Настройка
 - Excel, 48
 - интерфейса, 43
- Нахождение дат, 440
- Неверные ссылки, 468
- Недостающие скобки, 468
- Нелинейная тенденция, 403

Нелогичное поведение логических функций, 345
Непостоянные функции, 120
Несколько
 процентных ставок, 304
 условий, 184
Неявное пересечение, 87
Номинальная ставка, 282

О

Область
 данных, 417
 действия имен, 84
Общая сумма, 194; 417
Общее количество ячеек, 180
Объединение, 129
Объектная модель, 37
Объектно-ориентированное
 программирование, 37
Объекты графического слоя, 46
Объявление, 493
 массива, 521
 переменных, 517
 функции, 499
Ограничения финансовых функций, 289
Одинаковые значения, 187
Одномерный
 вертикальный массив, 331
 горизонтальный массив, 330
 массив, 327
Окно
 быстрого доступа, 489
 кода, 489
 проекта, 489
 свойств, 489
 сообщений VBA, 505
Округление, 259
 времени, 174
 денежных величин, 260
 до четного и нечетного целого, 262
Оператор, 55; 60
 Debug.Print, 506
 On Error, 529
 И, 184; 198
 ИЛИ, 185; 199
 пересечения, 87
Описание
 функции, 502
 ближайшего значения, 353
 високосного года, 166
 возраста человека, 159
 данных, 421
 дня недели, 160
 допустимых значений, 350
 категории функции, 502
 литеры столбца, 138
 максимального значения, 441
 неупорядоченных значений, 443
 нечисловых данных, 439
 положительных изменений, 445
 порядкового номера, 138
 расположения значения, 215
 суммы цифр числа, 350
 текста в ячейке, 125
 текущей даты, 440
 тенденции, 443
 условий, 435
Отделение ряда диаграммы, 380
Отладка формул, 467
Отмена имени, 90
Относительная ссылка, 99
Отображение
 последних точек данных, 391
 текущего времени, 168
 текущей датой, 154
Отслеживание
 ошибочных значений, 480
 связей ячейки, 478
Отсроченное начало регулярных выплат, 289
Ошибка
 #ДЕЛ/0!, 470
 #ЗНАЧ!, 472
 #ИМЯ?, 471
 #Н/Д, 471
 #ПУСТО!, 471
 #ССЫЛКА!, 472
 #ЧИСЛО!, 472
 в формуле, 71
 високосного года, 151
 записи формулы массива, 468
 незавершенных вычислений, 468
 формулы, 470
 выполнения, 504
 плавающей запятой, 475

П

Пакет анализа, 49; 121; 191; 247
Панель инструментов, 42; 49
Параметры сводной таблицы, 423
Первая строка списка, 223
Переименование проекта, 490

- Перекрестные таблицы, 355
- Переменная, 515
 - объекта, 536
- Переназначение имени, 91
- Пересечение, 213
 - диапазона, 87
- Перетаскивание, 42
- Периодическая эффективная ставка, 282
- Пирамида заселенности, 386
- План
 - амортизации, 312
 - накопления, 319
- Планы будущего поступления, 320
- Платеж, 268
- Повторяющиеся символы, 131
- Подбора параметра, 73
- Подсчет
 - количества ошибок, 344
 - количества символов, 339
 - количества слов, 141
 - непустых ячеек, 181
 - отличающихся значений, 348
 - пустых ячеек, 180
 - символов, 133; 137
 - строк, 137
 - текстовых ячеек, 340
- Поиск
 - в множестве таблиц, 210
 - данных, 201
 - даты, 148
 - значения, 347
 - максимально приближенного значения, 216
 - ошибок, 470
 - ошибок электронных таблиц, 467
 - решения, 52
 - с учетом регистра, 210
 - символа, 136
 - точного значения, 208
- Поле, 221
 - столбца, 417
 - страницы, 418
 - строки, 418
- Пользовательская функция, 283; 359; 497
- Пользовательские
 - форматы чисел, 591
 - функции, 142
 - интерфейс, 40
 - формат числа, 594
- Порядковая дата, 146
- Порядок вычислений, 575
- Последний день месяца, 165
- Правила создания имен, 79
- Предварительное форматирование ячейки, 596
- Предельный размер формул, 58
- Преобразование времени, 172
- Преобразование
 - дат, 147
 - единиц измерений, 247
 - процентных ставок, 281
 - ссылки в массив, 380
 - формул в значения, 68
- Привычка объявления переменных, 518
- Примеры пользовательских функций VBA, 539
- Принципы
 - программирования на VBA, 513
 - управления данными, 145
- Принятие решений, 110
- Приоритет операторов, 62; 473
- Присвоение имени, 78
 - имени массивам, 332
- Пробел, 57
- Проверка данных, 52; 447
- Проектирование списка, 223
- Промежуточная сумма, 418
- Промежуточные вычисления, 313
- Просмотр
 - именованных диапазонов, 91
 - формул, 477
- Простой план амортизации, 312
- Простые
 - задачи накопления, 270
 - условия, 435
- Процедура, 493
- Процентная ставка, 268
- Прямоугольный треугольник, 253
- ПС, 268
- Пустые ячейки, 180; 207

Р

- Работа с текстом, 123
- Рабочая книга, 37
- Рабочий лист, 38
- Равенство строк, 128
- Разброс данных, 387
- Разделение текстовых строк, 141
- Размер таблицы, 39
- Размерность массива, 330
- Разрыв строки, 57
- Расположение данных, 420
- Распределение частот, 188

- Расширение формулы массива, 334
- Расширенный фильтр, 227
- Региональные стандарты, 145
- Регистр текста, 134
- Редактирование формул, 59
 - массива, 334
- Редактор Visual Basic, 487
- Решение
 - рекурсивного уравнения, 372
 - систем линейных уравнений, 258
 - системы уравнений, 373
 - сложных задач, 101
- Ручной ввод
 - формул, 56
 - функций, 114
- Ряд последовательных дат, 155

С

- Сводная таблица, 51; 415
- Свойство
 - Address, 534
 - Cells, 532
 - Columns, 535
 - Count, 534
 - Font, 535
 - Formula, 533
 - Hidden, 535
 - Name, 534
 - Numberformat, 535
 - Offset, 533
 - Range, 532
 - Rows, 535
 - UsedRange, 537
- Синтаксические ошибки, 468; 504
- Система
 - дат, 146
 - линейных уравнений, 258
- Скрытие
 - ошибочных значений, 441
 - формул, 70
- Скрытые имена, 82
- Сложные задачи накопления, 273
- Смарт-тэг, 41; 69
- Смешанная ссылка, 100
- Советы по построению диаграмм, 378
- Содержимое ячейки, 187
- Создание
 - вычисляемого поля, 428
 - графиков, 395
 - динамических финансовых планов, 311
 - календаря, 167
 - мегаформулы, 453
 - надстроек, 510
 - пользовательского формата числа, 594
 - ряда последовательных дат, 155
 - сводной таблицы, 419
 - списка имен, 85
 - ссылок, 67
 - ссылок на ячейки, 380
 - текстовой гистограммы, 131
- Сокращение формулы массива, 334
- Сохранение
 - кода, 493
 - нескольких диаграмм на одном листе, 407
 - проекта, 496
- Сочетание клавиш, 43
- Сочетания клавиш, 592
- Специальные символы, 128
- Список, 221
 - имен, 85
- Средства
 - анализа, 49
 - проверки данных, 476
- Срок, 268
- Ссылка, 61
 - на диапазон, 532
 - на заголовок, 381
 - на имя, 85
 - на объект, 47
 - на подписи к данным, 381
 - на рисунок, 382
 - на ячейки, 55
 - на другие листы, 439
- Статистические функции, 119
- Стилевое форматирование, 45
- Строка, 519
- Структура таблицы, 51
- Суммирование, 177
 - n-го числа наибольших значений, 346
 - в диапазоне, 343
 - времени, 169
 - каждого n-го значения, 352
 - округленных значений, 351
 - отрицательных значений, 197

Т

- Таблица, 38
 - данных, 316
 - подстановки, 240; 317
- Табличная база данных, 50
- Текст, 123
- Текстовая ссылка, 381

Текстовые функции, 119; 125

Текущая дата, 154

Текущее

время, 168

значение, 268

Тенденция, 399

Тестирование и отладка функций, 504

Тип

данных, 515

ссылок, 66

форматирования, 434

аргументов, 111

данных VBA, 516

условий проверки, 449

файлов Lotus 1-2-3, 573

Точка остановки, 509

Точный аргумент, 113

Транспонирование массива, 337

У

Удаление

имени, 90

лишних пробелов, 133

столбца, 93

строки, 93

условного форматирования, 438

Умышленные циклические ссылки, 365

Уникальные записи, 229

Уникальные значения, 358

Упорядочение данных, 354

Управление

отсутствующими данными, 387

сценариями, 51

данными даты и времени, 145

Упрощение формулы, 109

Уровень

рабочего листа, 84

рабочей книги, 85

Ускорение выполнения задач, 110

Условие, 183

расширенного фильтра, 230

основанные на формулах, 436

Условное

суммирование, 195; 344

форматирование, 45; 433

Установка точек остановки, 509

Устранение промежуточных формул, 340

Ф

Фантомная ссылка, 476

Фигура, 46

Фильтрация, 223

списка, 229

Финансовые формулы, 267

Финансовые функции, 118

Фоновая проверка ошибок, 480

Формат

HTML, 37

даты и времени, 145

Форматирование, 44

даты и времени, 150

Формула, 46

массива, 327

РЯД, 377

Lotus 1-2-3, 574

линейной интерполяции, 218

с циклическими ссылками, 370

суммирования, 179; 193

условного форматирования, 439

Функции

баз данных, 238

БСЧЁТ, 178

даты и времени, 118

накопления, 269

округления, 259

просмотра, 119

работы с базами данных, 119

суммирования, 178

управления временем, 167

управления датами, 153

Функция, 46; 109; 493

Intersect, 536

ТИП, 126

Union, 537

автозаполнения, 70

БЗРАСПИС, 306

ВПР, 202

ВСД, 300

ГПР, 204

ДЕНЬНЕД, 441

ЕТЕКСТ, 125

ЗАМЕНИТЬ, 135

КОДСИМВ, 127

ЛЕВСИМВ, 134

МАКС, 441

МОДА, 185

НАЙТИ, 136

НЕЧЁТ, 262

ОКРУГЛ, 260

ОСНПЛАТ, 287
ПОДСТАВИТЬ, 135
ПОИСК, 136
ПЛАТ, 287
ПРАВСИМВ, 134
ПРОМЕЖУТОЧНЫЕ.ИТОГИ, 225; 226
ПРОСМОТР, 205
ПСТР, 134
РАБДЕНЬ, 158
РИМСКОЕ, 166
РУБЛЬ.ДЕС, 261
РУБЛЬ.ДРОБЬ, 261
СИМВОЛ, 127
СУММ, 193
СЧЁТЕСЛИ, 183
СЧЁТЗ, 181
СЧИТАТЬПУСТОТЫ, 181
ЦЕЛОЕ, 262
ЧАСТОТА, 189
ЧЁТН, 262
ЧИСТВНДОХ, 321
ЧИСТНЗ, 322
ЧИСТРАБДНИ, 157
ЧПС, 293; 294
ЯЧЕЙКА, 126

Эффективная стоимость займа, 283

Я

Явное пересечение, 87

Х

Хранение модулей VBA, 37

Ц

Цикл, 526
Do Until, 529
Do While, 528
For-Next, 526
Циклическая
обработка, 526
ссылка, 72; 476

Ч

Числа как текст, 124
Числовое форматирование, 45

Э

Экспоненциальная тенденция, 405
Элемент, 418
массива, 330
управления, 48
формул, 55

Научно-популярное издание

Джон Уокенбах

Подробное руководство по созданию формул в Excel 2002

Литературный редактор *Н.В. Никифорова*

Верстка *А.А. Линник (мл.)*

Художественный редактор *В.Г. Павлютин*

Корректоры *Л.А. Гордиенко и О.В. Мишутина*

Издательский дом “Вильямс”.

101509, Москва, ул. Лесная, д. 43, стр. 1.

Изд. лиц. ЛР № 090230 от 23.06.99

Госкомитета РФ по печати.

Подписано в печать 26.08.2002. Формат 70×100/16.

Гарнитура Times. Печать офсетная.

Усл. печ. л. 50,31. Уч.-изд. л. 35,7.

Тираж 3500 экз. Заказ № 1149.

Отпечатано с диапозитивов в ФГУП “Печатный двор”
Министерства РФ по делам печати,
телерадиовещания и средств массовых коммуникаций.
197110, Санкт-Петербург, Чкаловский пр., 15.



Подробное руководство по созданию формул в Excel 2002

Джон Уокенбах является директором консалтинговой фирмы Jwalk and Associates Inc., которая специализируется на разработке приложений для электронных таблиц. За свою многолетнюю практику Джон успел написать около 30 книг и опубликовать более 300 статей в самых различных изданиях, посвященных электронному представлению табличных данных, таких как *PC World*, *InfoWorld*, *PC Magazine* и *PC Computing*. Он также самостоятельно поддерживает весьма популярный Web-узел *The Spreadsheet Page* (www.j-walk.com/ss).

НА WEB-УЗЛЕ ИЗДАТЕЛЬСТВА ВЫ НАЙДЕТЕ:

- пробную версию созданного автором пакета Power Utility Pak 2000
- демонстрационную версию утилиты Sound-Proof 2000
- все примеры рабочих книг, используемых в данной книге

Системные требования:

компьютер с процессором Pentium,
32 Мбайт оперативной памяти,
OS Windows,
Microsoft Excel 2002

Категория:

для пользователей средней
и высшей квалификации

Предмет рассмотрения:

Excel/электронные таблицы

WWW.DIALEKTIKA.COM
WWW.MANDTBOOKS.COM

Полный справочник по формулам Excel 2002

Наряду с описанием всех новых средств Excel 2002, это полное справочное руководство, в новое издание которого добавлены две главы, посвященные экономическим формулам, детально рассматривает все аспекты и методики применения формул для упорядочения данных в электронных таблицах. Простое и точное описание операторов, функций, уровней вложения и принципов взаимодействия формул, а также сотни примеров, взятых из реальной жизни эксперта по электронным таблицам, позволяют быстро найти максимально эффективное решение самых сложных задач. Управление датами, просмотр диапазонов, создание формул массивов, финансовые расчеты — все это выделяет книгу на фоне «однотипных» изданий, посвященных Excel.

Залог успеха эффективного использования Excel заключается в четком понимании формул

- Познакомьтесь с многофункциональными средствами форматирования ячеек, анализа формул и защиты данных
- Освойте основные элементы формул — абсолютные и относительные ссылки, операторы, сообщения об ошибках, соглашение об именовании и т.д.
- Получите ценные советы по добавлению в формулы функций, полностью соответствующих условиям поставленной задачи
- Откройте для себя удивительный мир финансовых расчетов
- Примените на практике совершенные и многогранные формулы массивов
- Улучшите с помощью формул сводные таблицы и диаграммы
- Научитесь отлаживать формулы с помощью специальных средств проверки данных
- Разработайте собственные функции с помощью VBA

ISBN 5-8459-0314-9



9 785845 903143