

EXCEL. ТРЮКИ

*100 профессиональных
примеров*



O'REILLY®

 ПИТЕР®

Рэина Холи, Дэвид Холи

EXCEL HACKS

James Hovey, David Hovey



EXCEL HACKS

Ralna Hawley, David Hawley

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Paris • Sebastopol • Taipei • Tokyo

EXCEL ТРЮКИ

Рэина Холи, Дэвид Холи



Москва • Санкт-Петербург • Нижний Новгород • Воронеж
Новосибирск • Ростов-на-Дону • Екатеринбург • Самара
Киев • Харьков • Минск

2005

ББК 32.973.233
УДК 681.3.06
Х-71

Холи Р., Холи Д.

Х-71 Excel. Трюки. — СПб.: Питер, 2005. — 287 с.: ил.

ISBN 5-469-00384-1

За последние десять лет Excel фундаментально изменил способы работы с числовыми данными, но большинство его возможностей все еще остается в тени. Книга «Excel. Трюки» помогает читателю сразу же перейти к практическим инструментам и методам анализа, обработки и представления данных. В издании рассматриваются методы наиболее эффективной обработки данных, которые помогают воспользоваться всей мощью Excel и быстро добиться необходимых результатов.

Книга предназначена для студентов, аналитиков и всех пользователей компьютеров, желающих узнать о возможностях Excel, связанных с обработкой данных.

ББК 32.973.233
УДК 681.3.06

Права на издание получены по соглашению с O'Reilly.

Все права защищены. Никакая часть данной книги не может быть воспроизведена в какой бы то ни было форме без письменного разрешения владельцев авторских прав.

Информация, содержащаяся в данной книге, получена из источников, рассматриваемых издательством как надежные. Тем не менее, имея в виду возможные человеческие или технические ошибки, издательство не может гарантировать абсолютную точность и полноту приводимых сведений и не несет ответственности за возможные ошибки, связанные с использованием книги.

© 2004 O'Reilly Media, Inc.
ISBN 059600625X (англ.) © Перевод на русский язык ЗАО Издательский дом «Питер», 2005
ISBN 5-469-00384-1 © Издание на русском языке, оформление ЗАО Издательский дом «Питер», 2005

Краткое содержание

Кто создавал эту книгу.....	12
Предисловие.....	14
ГЛАВА 1. Упрощение рабочих книг и листов.....	22
ГЛАВА 2. Трюки со встроенными возможностями Excel.....	60
ГЛАВА 3. Трюки с именованиeм.....	109
ГЛАВА 4. Трюки со сводными таблицами.....	126
ГЛАВА 5. Трюки с диаграммами.....	139
ГЛАВА 6. Трюки с формулами и функциями.....	169
ГЛАВА 7. Трюки с макросами.....	209
ГЛАВА 8. Соединение Excel с остальным миром.....	235
Словарь терминов.....	277
Алфавитный указатель.....	281

Содержание

Кто создавал эту книгу	И
Об авторах.....	12
Помощники.....	12
Благодарности.....	13
Предисловие	14
Зачем нужны трюки Excel?.....	14
Правило 80/20.....	14
Конструктивные советы.....	15
Советы по форматированию.....	16
Советы по применению формул.....	16
Где взять и как применить эти трюки.....	18
Как работать с этой книгой.....	18
Как организована книга.....	18
Пользователи Windows и Macintosh.....	20
Сокращения, принятые в этой книге.....	20
Использование примеров кода.....	21
От издательства.....	21
ГЛАВА 1. Упрощение рабочих книг и листов	22
Трюк № 1. Создание персональных представлений рабочих книг.....	22
Трюк № 2. Одновременный ввод данных в несколько рабочих листов.....	25
Трюк № 3. Запрещение пользователям выполнять определенные действия.....	28
Трюк № 4. Запрещение ненужных сообщений.....	31
Трюк № 5. Как скрыть лист, чтобы его невозможно было отобразить.....	35
Трюк № 6. Настройка диалогового окна шаблонов и рабочей книги по умолчанию.....	36
Трюк № 7. Создание указателя листов в книге.....	38
Трюк № 8. Ограничение диапазона прокрутки листа.....	41
Трюк № 9. Блокировка и защита ячеек, содержащих формулы.....	44
Трюк № 10. Поиск повторяющихся данных при помощи условного форматирования.....	47
Трюк № И. Пользовательские панели инструментов для определенной книги.....	49
Трюк № 12. Как перехитрить обработчик относительных ссылок Excel.....	51

Трюк № 13. Удаление несуществующих ссылок на книги.....	52
Трюк № 14. Предотвращение распухания рабочей книги.....	55
Трюк № 15. Получение данных из поврежденной книги.....	57
ГЛАВА 2. Трюки со встроенными возможностями Excel.....	60
Трюк № 16. Проверка данных на основе списка на другом листе.....	60
Трюк № 17. Управление условным форматированием при помощи флажков.....	62
Трюк № 18. Поиск формул при помощи условного форматирования.....	65
Трюк № 19. Подсчет и суммирование ячеек, отвечающих критерию условного форматирования.....	67
Трюк № 20. Выделение строк или столбцов через один.....	69
Трюк № 21. Трехмерные эффекты в таблицах и ячейках.....	71
Трюк № 22. Включение и выключение условного форматирования и проверки данных при помощи флажка.....	72
Трюк № 23. Поддержка нескольких списков в поле со списком.....	74
Трюк № 24. Создание списков проверки, меняющихся на основе выбранного значения в другом списке.....	76
Трюк № 25. Как при проверке данных заставить Excel использовать список на другом рабочем листе.....	78
Трюк № 26. Удаление ненужных символов при помощи замены.....	80
Трюк № 27. Преобразование текстовых чисел в настоящие числа.....	80
Трюк № 28. Настройка примечаний ячеек.....	83
Трюк № 29. Сортировка более чем по трем столбцам.....	85
Трюк № 30. Случайная сортировка.....	86
Трюк № 31. Манипулирование данными при помощи расширенного фильтра.....	87
Трюк № 32. Создание пользовательских числовых форматов.....	90
Трюк № 33. Увеличение количества уровней отмены действий в Excel для Windows.....	95
Трюк № 34. Создание пользовательских списков.....	96
Трюк № 35. Промежуточные итоги жирным шрифтом.....	96
Трюк № 36. Преобразование формул и функций Excel в значения.....	99
Трюк № 37. Автоматическое добавление данных в список проверки.....	101
Трюк № 38. Трюки с возможностями даты и времени в Excel.....	104
ГЛАВА 3. Трюки с именованием.....	109
Трюк № 39. Адресация данных по имени.....	109
Трюк № 40. Использование одного имени для диапазонов на разных рабочих листах.....	110
Трюк № 41. Создание пользовательских функций при помощи имен.....	112

Трюк № 42. Создание увеличивающихся и уменьшающихся диапазонов.....	115
Трюк № 43. Вложение динамических диапазонов для максимальной гибкости.....	121
Трюк № 44. Поиск именованных диапазонов на рабочем листе.....	124
ГЛАВА 4. Трюки со сводными таблицами.....	126
Трюк № 45. Сводные таблицы: сами по себе — трюк.....	126
Трюк № 46. Совместное использование сводных таблиц, но не их данных.....	130
Трюк № 47. Автоматизация создания сводных таблиц.....	132
Трюк № 48. Перемещение общих итогов сводных таблиц.....	135
Трюк № 49. Эффективная вставка данных из другой рабочей книги.....	137
ГЛАВА 5. Трюки с диаграммами.....	139
Трюк № 50. Выделение одного сектора из круговой диаграммы.....	139
Трюк № 51. Создание двух наборов секторов в одной круговой диаграмме.....	141
Трюк № 52. Создание диаграмм , изменяющихся вместе с данными.....	143
Трюк № 53. Взаимодействие с диаграммами при помощи пользовательских элементов управления.....	146
Трюк № 54. Три быстрых способа обновления диаграмм.....	149
Трюк № 55. Трюк с созданием простой диаграммы термометра.....	153
Трюк № 56. Создание гистограмм с переменной шириной и высотой.....	156
Трюк № 57. Создание диаграммы спидометра.....	159
Трюк № 58. Привязка текстовых элементов диаграмм к ячейкам.....	166
Трюк № 59. Трюки с данными диаграммы, чтобы пустые ячейки не учитывались при построении графика.....	167
ГЛАВА 6. ТРЮКИ С формулами И ФУНКЦИЯМИ.....	169
Трюк № 60. Добавление описаний к формулам.....	169
Трюк № 61. Перемещение относительных формул без изменения ссылок.....	170
Трюк № 62. Сравнение двух диапазонов Excel.....	171
Трюк № 63. Заполнение всех пустых ячеек в списке.....	173
Трюк № 64. Как сделать, чтобы формула ссылалась на следующие строки при копировании по столбцам.....	175
Трюк № 65. Преобразование дат в отформатированные даты Excel.....	177
Трюк № 66. Суммирование и подсчет ячеек.....	178
Трюк № 67. Как уменьшить влияние переменных функций на скорость пересчета.....	180

Трюк № 68. Подсчет только по одному вхождению каждой записи в списке.....	181
Трюк № 69. Суммирование каждой второй, третьей или п-й строки или ячейки.....	182
Трюк № 70. Поиск п-го вхождения значения.....	184
Трюк № 71. Превращение функции Excel ПРОМЕЖУТОЧНЫЕ.ИТОГИ (SUBTOTAL) в динамическую.....	187
Трюк № 72. Добавление расширений данных.....	189
Трюк № 73. Преобразование чисел со знаком минус справа.....	190
Трюк № 74. Отображение отрицательных значений времени.....	192
Трюк № 75. Использование функции ВПР (VLOOKUP) в нескольких таблицах.....	194
Трюк № 76. Отображение общего количества времени в днях, часах и минутах.....	196
Трюк № 77. Определение количества указанных дней в любом месяце.....	197
Трюк № 78. Построение мегаформул.....	199
Трюк № 79. Трюки с мегаформулами, которые ссылаются на другие рабочие книги.....	201
Трюк № 80. Трюк с одной функцией баз данных Excel, чтобы она работала вместо нескольких функций.....	203

ГЛАВА 7. Трюки с макросами..... 209

Трюк № 81. Ускорение выполнения кода и исключение мерцания экрана.....	209
Трюк № 82. Запуск макроса в определенное время.....	210
Трюк № 83. Использование кодовых имен для ссылки на листы в рабочих книгах Excel.....	212
Трюк № 84. Простое подключение кнопок к макросам.....	213
Трюк № 85. Создание всплывающего экрана для рабочей книги.....	214
Трюк № 86. Отображение сообщения «Please Wait».....	216
Трюк № 87. Как помечать и сбрасывать пометку ячеек при выделении.....	217
Трюк № 88. Подсчет или суммирование ячеек с определенным цветом заливки.....	219
Трюк № 89. Добавление элемента управления Календарь в любую рабочую книгу Excel.....	220
Трюк № 90. Включение и снятие защиты паролем для всех рабочих листов Excel одновременно.....	222
Трюк № 91. Получение имени и пути рабочей книги.....	225
Трюк № 92. Как обойти ограничение Excel на три критерия условного форматирования.....	226
Трюк № 93. Выполнение процедур на защищенных рабочих листах.....	228
Трюк № 94. Распространение макросов.....	229

ГЛАВА 8. Соединение Excel с остальным миром.....	235
Трюк № 95. Загрузка документа XML в Excel.....	235
Трюк № 96. Сохранение в виде SpreadsheetML и выделение данных.....	245
Трюк № 97. Создание электронных таблиц при помощи SpreadsheetML.....	254
Трюк № 98. Импорт данных напрямую в Excel.....	258
Трюк № 99. Доступ к веб-службам SOAP из Excel.....	266
Трюк № 100. Создание электронных таблиц Excel в других средах разработки.....	271
Словарь терминов.....	277
Алфавитный указатель.....	281

Посвящается нашим дочерям, Алейше и Кейт

Эта книга посвящена нашим дочерям, Алейше и Кейт. Мы хотим выразить нашу любовь и благодарность за то, что вы сделали для нас. Мы надеемся, что эта книга будет для вас источником вдохновения и радости. Мы также хотим поблагодарить наших помощников, которые помогли нам в этом процессе. Мы надеемся, что вы найдете в этой книге много полезной информации и интересных фактов. Мы также хотим поблагодарить наших читателей, которые поддержали нас в этом процессе. Мы надеемся, что вы найдете в этой книге много полезной информации и интересных фактов. Мы также хотим поблагодарить наших читателей, которые поддержали нас в этом процессе.

Кто создавал эту книгу

Об авторах

Дэвид и Рэйна Хоули (David and Raina Hawley) разрабатывают бизнес-приложения, проводят обучение и консультации по всем вопросам, связанным с Excel и VBA для Excel, в OzGrid Business Applications в Западной Австралии.

Последние 12 лет Дэвид Хоули посвящает себя непрерывному созданию бизнес-приложений на основе Excel и VBA для Excel. Он выпускает ежемесячный информационный бюллетень по использованию Excel и VBA для Excel. Он запустил и поддерживает популярный форум, посвященный Excel, на сайте OzGrid.

Рэйна Хоули преподает на производстве и в колледжах и является дипломированным экспертом по производственным условиям. Рэйна занимается офисом OzGrid и административной стороной бизнеса, а также работает над решениями для Excel вместе со своим мужем. Авторы предлагают сотни надстроек для Excel и программ для анализа данных во всех областях промышленности, которые можно найти на сайте <http://www.ozgrid.com>. Авторы живут в Бенбери, в Западной Австралии, со своими двумя детьми.

Помощники

Следующие люди предоставили свои трюки и программы, а также вдохновили нас на написание этой книги:

- Энди Поуп (Andy Pope) начал работать в Digitab в 1986 году в качестве младшего программиста и в основном писал на языке Fortran. Digitab — это расположенная в Лондоне компания, занимающаяся обработкой данных. Сейчас Энди занимается в Digitab всеми вопросами, связанными с информационными технологиями: он разрабатывает и внедряет сетевую инфраструктуру компаний, приобретает аппаратное и программное обеспечение, устанавливает и поддерживает мейнфреймы, миникомпьютеры, серверы и настольные персональные компьютеры, а также предоставляет пользователям поддержку в вопросах программного и аппаратного обеспечения. В данный момент он занимается разработкой заказных решений для проектов отчетов на основе Office и VBA. Вы можете посетить сайт Энди по адресу: <http://www.andypope.info/>.
- Пол Бауш (Paul Vausch) достиг совершенства в разработке веб-приложений, он один из создателей популярного программного обеспечения для ведения сетевого дневника Blogger (посетите сайт <http://www.blogger.com/>). Он участвовал в написании *We Blog: Publishing Online with Weblogs* (John Wiley & Sons) и практически ежедневно публикует свои мысли и фотографии в личном веблоге <http://www.onfocus.com/>.

- Саймон Сен-Лоран (Simon St.Laurent) работает редактором в O'Reilly and Associates, Inc. До этого он успел побывать веб-разработчиком, сетевым администратором, автором книг на компьютерные темы и занимался XML. Он живет в Драйдене, Нью-Йорк. Среди его книг такие как *XML: A Primer* и *XML Elements of Style*. Он действующий редактор *xmlhack* (<http://www.xmlhack.com>) и иногда предлагает свои идеи для XML.com (<http://www.xml.com>). Подробнее о книгах и проектах Саймона — на сайте <http://simonstl.com>.

Благодарности

Прежде всего, мы хотим поблагодарить наших родителей, Волтера и Берил Фенлон (Walter and Beryl Fenlon) и Майка и Марлин Хоули (Mike and Marlene Hawley), — без вашей любви и поддержки мы никогда не смогли бы сделать то, что сделали.

Выражаем свою благодарность Джону Риду (John Read), который оказывал нам помощь и поддержку в течение всей работы над книгой, и команде издательства O'Reilly, особенно Саймону, за весь их тяжелый труд, который они вложили в эту книгу. Необходимо также поблагодарить Энди Поупа за предоставленные им трюки, а также отметить всех посетителей нашего сайта и форума, которые помогли нам выяснить, с какими же проблемами чаще всего встречаются пользователи.

И, наконец, нужно поблагодарить Алейшу и Кейт, как всегда, за вдохновение. Мы никогда не забудем их понимания и дополнительного старания вести себя хорошо в то время, когда мы работали над книгой!!!

Предисловие

Миллионы пользователей Microsoft Excel каждый день занимаются созданием и распространением электронных таблиц. Действительно, электронная таблица из мощного и удобного средства превратилась в гибкую основу для множества различных видов бизнеса, помогая и подсказывая решения руководителям компаний во всем мире.

Хотя Excel — это чрезвычайно важный инструмент, многие пользователи Excel знают лишь о части его возможностей. Они используют те функции, которые им необходимы, часто применяя более сложные фрагменты из существующих шаблонов, но они не задумываются обо всей той мощи, которую может предложить Excel. Естественно, никому не нужны абсолютно все функции и возможности, существующие в Excel, поэтому такой подход вполне разумен. В то же время, однако, это означает, что множество пользователей никогда не продвинется по кривой обучения, чтобы узнать о методах, которые могли бы значительно упростить их работу.

Зачем нужны трюки Excel?

Хотя огромное количество задач можно выполнить, применяя относительно простой набор возможностей Excel, эта программа предлагает намного более мощные методы, которые могли бы далеко продвинуть качество и скорость работы без многолетних затрат на ее изучение. Однако большинство пользователей фокусируются на существе того, что создают — на данных и формулах, и иногда диаграммах, поэтому переход на более высокий уровень работы с Excel выглядит трудной задачей.

Есть много способов воспользоваться преимуществами Excel и значительно расширить возможности создания сложных электронных таблиц, и чтобы познакомиться с ними, не нужно тратить годы на обучение. Эти средства, *трюки*, — быстрые и хитрые решения проблем или более искусные приемы — были созданы пользователями Excel, которые хотели найти простые решения сложных задач. Трюки, о которых рассказывает эта книга, разработаны, чтобы показать возможности Excel и заставить их работать как надо.

Изучив эти трюки, вы получите преимущество в двух важных аспектах. Во-первых, вы сможете применять эти трюки при построении и улучшении ваших электронных таблиц. Во-вторых, изучив трюки и, возможно, немного кода Visual Basic for Applications (VBA), вы сможете настроить их в точном соответствии с вашими требованиями.

Правило 80/20

Возможно, самое важное правило при разработке электронной таблицы — учитывать долгосрочные перспективы и не предполагать, что в нее никогда не понадо-

биться добавлять данные или формулы, так как шансов, что вам это понадобится, все же больше. Помня об этом, вы должны потратить приблизительно 80 % времени на планирование электронной таблицы и приблизительно 20 % времени на ее реализацию. Хотя на первый взгляд это может показаться чрезвычайно неэффективным, уверяем вас, что долгосрочные выгоды перевесят краткосрочные мучения, а планирование станет намного проще после того, как вы некоторое время потренируетесь. Помните, что электронные таблицы предназначены для того, чтобы пользователи могли легко получать правильную информацию, а не только для ее представления, причем представление это будет эффективным и правильным только один раз в жизни.

Конструктивные советы

Без сомнения, главная ошибка, которую большинство пользователей Excel делают при создании электронных таблиц, состоит в том, что они не планируют данные так, как этого ожидают Excel и его средства. Вот несколько самых распространенных ошибок, встречающихся при проектировании таблицы (я не сортировал их по важности или частоте появления):

- Ненужное распределение данных по нескольким различным рабочим книгам.
- Ненужное распределение данных по нескольким рабочим листам.
- Ненужное распределение данных по различным таблицам.
- Пустые столбцы и строки в таблицах с данными.
- Пустые ячейки там, где должны быть повторяющиеся данные.

Первые три варианта в этом списке ведут к одному выводу: необходимо стараться хранить связанные данные в одной непрерывной таблице. Снова и снова мы видим электронные таблицы, не выполняющие это простое правило и потому ограниченные в своей возможности пользоваться всеми преимуществами некоторых наиболее мощных возможностей Excel, включая Сводные таблицы (PivotTables), промежуточные суммы и формулы в рабочих листах. Подобные возможности можно максимально эффективно использовать, только когда данные собраны в очень простую таблицу.

Не случайно электронные таблицы Excel могут состоять из 65 536 строк, но только из 256 столбцов. Помня это, вы должны сконструировать таблицу так, чтобы заголовки столбцов помещались в первой строке таблицы, а соответствующие данные последовательно хранились непосредственно под связанными с ними заголовками. Обнаружив, что повторяете одни и те же данные снова и снова в двух и более строках в одном из этих столбцов, будьте готовы воспротивиться соблазну оставить в столбце пустые ячейки, обозначающие повторяющиеся данные.

Когда возможно, сортируйте данные. Excel поддерживает широкий набор формул поиска и ссылочных формул, и некоторые из них требуют, чтобы данные были отсортированы в логическом порядке. Кроме того, сортировка существенно ускоряет процесс вычисления многих функций.

Советы по форматированию

Оставляя в стороне структуру, необходимо заметить, что форматирование также может привести к появлению проблем. Хотя таблицы должны быть максимально простыми для чтения и обзора, это не должно достигаться за счет эффективности. Мы свято верим в принцип простоты. Слишком много пользователей тратят огромное количество времени на форматирование таблиц. Хотя они не всегда осознают это, зачастую это делается за счет снижения эффективности. Также часто излишнее форматирование увеличивает размер рабочей книги, и, хотя ваша рабочая книга именно вам может казаться произведением искусства, для другого пользователя она может стать кошмаром. Лучше всего использовать в электронных таблицах некоторые универсальные цвета, а именно черный, белый и серый.

Всегда полезно оставлять, по меньшей мере, три пустых строки над таблицей (*минимум* три, лучше больше). Их можно будет использовать для определения условий в таких средствах Excel, как Расширенный фильтр (Advanced Filter) и функции работы с базами данных.

Обычно пользователи небрежно работают с данными в ячейках. По умолчанию числа в Excel выравниваются по правому краю, а текст — по левому, и для этого есть основательные причины. Если вы начнете изменять такое форматирование, то не сможете с первого взгляда определять, является ли содержимое ячейки текстом или числом. Очень часто люди ссылаются на какие-то ячейки, считая их содержимое числами, но на самом деле в них хранится текст. Если вы измените выравнивание по умолчанию, останется только в задумчивости чесать голову, глядя на содержимое ячеек. Возможно, исключением из этого правила являются заголовки.

Форматируйте ячейки как текст, только когда это действительно необходимо. Все введенные в ячейки данные, отформатированные как текст, становятся текстовыми, даже если вы хотели оставить их числами или датами. Что еще хуже, любая ячейка с формулой, которая ссылается на ячейку, отформатированную как текст, также станет текстовой. В общем случае вы вряд ли захотите отформатировать как текст ячейки с формулами.

Объединенные ячейки также могут быть причиной проблем. База знаний Microsoft содержит множество часто встречающихся проблем с объединенными ячейками. В качестве хорошей альтернативы используйте параметр По центру выделения (Centeracrossselection) в группе По горизонтали (Horizontal) на вкладке Выравнивание (Alignment) диалогового окна Формат ячеек (Format Cells).

Советы по применению формул

Еще одна ошибка, которую пользователи часто делают в формулах Excel, — это ссылки на целые столбцы. Это приводит к тому, что Excel начинает просматривать тысячи, если не миллионы ячеек, которые в противном случае были бы попросту пропущены.

Предположим, например, у вас есть таблица данных, начинающаяся с ячейки A1 и заканчивающаяся H1000. Вы хотите применить одну или несколько формул поиска Excel для выбора необходимой информации. Так как ваша таблица может со временем увеличиться (по мере добавления новых данных), в формуле вы ссылаетесь на всю таблицу, включая все строки. Другими словами, эта ссылка может выглядеть приблизительно как A:H или, возможно, A1:H65536. Такой вариант используется для автоматического учета в формулах новых добавленных в таблицу данных.

Это чрезвычайно плохая привычка, и практически всегда следует избавляться от нее. Можно исключить необходимость постоянного обновления ссылок в формулах для включения новых добавляемых в таблицу данных с помощью динамических именованных диапазонов, которые рассматриваются в разделе «Создание увеличивающихся и уменьшающихся диапазонов» [Трюк № 42].

Еще одна распространенная проблема, связанная с плохо спроектированными электронными таблицами, — это болезненно низкая скорость пересчета. Многие полагают, что эту проблему может решить режим вычисления вручную (Manual) — Сервис ▶ Параметры ▶ Вычисления (Tools ▶ Options ▶ Calculations).

Однако в общем случае это очень плохой совет, грозящий потенциальными бедами. Электронная таблица существует за счет формул и вычислений и доставляемых ими результатов. Если вы работаете с электронной таблицей в ручном режиме вычисления, рано или поздно в таблице появится необновленная информация. Формулы будут отражать старые, а не обновленные значения, так как в ручном режиме вычисления необходимо запускать пересчет, нажимая клавишу F9. Но об этом так легко забыть! Попробуем провести аналогию. Если бы тормоза на вашей машине были неисправны и, создавая излишнее трение, замедляли бы машину, то отключили бы вы педаль тормоза, положившись исключительно на ручной тормоз, вместо того чтобы устранить неисправность? Большинству из нас такая мысль даже не пришла бы в голову, однако множество пользователей, ни минуты не сомневаясь, переключают свои электронные таблицы в ручной режим вычисления. Если ситуация требует, чтобы таблица работала в ручном режиме вычисления, — у вас проблема с дизайном. Отнеситесь к этому серьезно, не решайте ее поверхностным или временным способом.

Еще одна часто встречающаяся причина проблем — формулы массивов. В этих формулах лучше всего использовать отдельные ячейки. Если вы ссылаетесь на большие диапазоны, делайте это как можно реже. Когда большое количество массивов ссылается на большие диапазоны, страдает производительность рабочей книги, причем это может дойти до такой стадии, что книга будет просто непригодна для работы и таблицу придется переключать в ручной режим вычисления.

Множество альтернатив формулам массивов можно найти среди функций баз данных Excel. Об этом рассказывается в разделе «Суммирование и подсчет ячеек» [Трюк № 66]. Несколько очень хороших примеров использования этих формул на больших таблицах данных и возврата результатов на основе нескольких критериев можно найти в справке Excel. Часто пользователи упускают еще одну возможность — сводную таблицу, которую мы рассмотрим в главе 4. Хотя, впервые встретив сводные таблицы, вы можете быть обескуражены, мы рекомендуем ближе познакомиться с этой замечательной возможностью Excel. Овладев искусством работы со сводными таблицами, вы будете удивляться, как раньше обходились без них.

Если вы все же ничего не запомнили о дизайне электронной таблицы, запомните только, что Excel работает лучше всего, когда все связанные данные собраны в одну непрерывную таблицу. Это существенно упростит применение трюков Excel.

Где взять и как применить эти трюки

Чтобы сэкономить время и силы и не набирать от руки скрипты и электронные таблицы, все трюки (кроме трюков длиной в несколько строк или использующих только GUI) можно загрузить с сайта авторов <http://www.ozgrid.com/BookExamples/excel-hack-examples.htm>.

Вы будете копировать код из примеров и модифицировать его для более точного соответствия вашим требованиям. Электронные таблицы Excel чрезвычайно разнообразны, добейтесь от них того, что нужно именно вам.

И, наконец, помните, что, хотя все сценарии, инструменты, процедуры и источники, описанные и содержащиеся в этой книге, были проверены, автор и эксперты, предоставившие их, не могут гарантировать, что в вашем сетевом окружении все будет работать, как предусмотрено. Мы не несем никакой ответственности и за любые потери и повреждения, проистекающие от их использования. Другими словами, информация в этой книге предоставляется *такой, как она есть*, и мы настоятельно рекомендуем проверить трюк перед тем, как применять его в производственной среде.

Как работать с этой книгой

Хотя книга разделена на главы, перечисленные в следующем разделе, вы можете работать с ней разными способами. Один подход заключается в том, чтобы считать книгу набором инструментов и начинать со знакомства с ними в каждой главе. Затем, если возникает необходимость или проблема, вы просто сможете применить подходящее делу средство. А можно и пролистать книгу или прочитать ее от корки до корки, изучая процедуры и скрипты и все больше узнавая Excel. Некоторые из трюков этим и хороши, содержат объяснения сложных тем или хорошо документированные скрипты. Помимо этого, можно просто выбрать одну главу и посмотреть, что в ней полезного для текущей ситуации, а что может пригодиться в будущем.

Как организована книга

В любом варианте работы с книгой сначала вы захотите познакомиться с содержанием, поэтому далее перечислены главы этой книги и краткое описание каждой из них.

Глава 1. Упрощение рабочих книг и листов

Рабочие книги и листы — основной интерфейс данных в Excel, но иногда этот набор гигантских открытых таблиц не может выполнить именно то, что вам нужно. Эти трюки позволят управлять взаимодействием пользователей и рабочих листов, помогут найти и выделить информацию и научат справляться с хламом и искажениями.

Глава 2. Трюки со встроенными возможностями Excel

В Excel есть много встроенных средств анализа и управления данными. Однако они зачастую бывают чем-нибудь ограничены. Трюки этой главы помогут расширить и автоматизировать такие функции, выходя за пределы ограниченных задач, для выполнения которых они и были первоначально предназначены.

Глава 3. Трюки с именованьем

Хотя ссылки на ячейки вида A2 и IV284:IN1237 определенно бывают полезны, с увеличением электронных таблиц становится проще ссылаться на информацию по имени. Эти трюки демонстрируют не только, как именовать ячейки и диапазоны, но и как создавать имена, приспособленные к данным таблицы.

Глава 4. Трюки со сводными таблицами

Для многих пользователей Excel Сводные таблицы (PivotTables) уже выглядят как сложные магические фокусы. Трюки этой главы покажут, как без проблем добиться от сводных таблиц большего.

Глава 5. Трюки с диаграммами

Встроенные функции Excel для создания диаграмм весьма полезны, но не всегда могут обеспечить лучший способ просмотра данных таблицы. Трюки научат настраивать и комбинировать встроенные функции Excel при создании пользовательских диаграмм.

Глава 6. Трюки с формулами и функциями

Формулы и функции лежат в основе большинства электронных таблиц, по иногда способ их обработки в Excel не вполне удовлетворителен. Эти трюки посвящены различным темам — от перемещения формул до решения проблем с типами данных и улучшения времени пересчета.

Глава 7. Трюки с макросами

Макросы (и VBA) — это аварийный выход из Excel; они позволяют конструировать электронные таблицы, выходящие за пределы собственных возможностей Excel, и разрабатывать таблицы, больше похожие на программы. Трюки этой главы научат максимально эффективно работать с макросами, от управления ими до применения для расширения других возможностей.

Глава 8. Соединение Excel с остальным миром

Хотя большинство электронных таблиц самодостаточно, эта глава демонстрирует, какие преимущества можно получить от веб-сайтов и служб, от Google до Amazon, и от применения XML для ввода и вывода информации таблиц.

Пользователи Windows и Macintosh

Трюки этой книги были написаны для версий Excel для Windows и проверены на Macintosh в Excel.X. Большая часть различий версий для этих двух платформ – чисто косметическая, и многие из них сводятся к разным клавишным комбинациям и некоторым меню. Если комбинации клавиш отличаются, то в книге сначала указан модификатор для Windows, например, **Alt/Command+Q**, что означает **Alt+Q** для Windows и **Command+Q** для Macintosh. Есть также несколько случаев, особенно в VBE (Visual Basic Editor, редактор Visual Basic), когда по-разному выглядят интерфейс и пункты меню. Такие случаи объясняются при первом упоминании. В книге есть несколько трюков только для Windows. Они используют реестр Windows и другие возможности, которые поддерживаются только в версии Excel для Windows. Эти трюки специально отмечены в тексте.

Пользователи Macintosh, владеющие **однокнопочной** мышью, должны также помнить, что для имитации щелчка правой кнопкой мыши им необходимо удерживать клавишу Control. (Пользователи Macintosh с **двухкнопочной** мышью могут просто щелкнуть правой кнопкой мыши.)

Большинство трюков должно работать во всех версиях Excel, начиная с Excel 97, и более поздних. Если это не так, в тексте будет отдельное упоминание. Снимки экрана были сделаны для различных версий Excel и не являются индикатором того, какие трюки работают в каких версиях программы.

Сокращения, принятые в этой книге

В книге используются следующие типографские соглашения.

Специальный шрифт

Показывает идентификаторы ячеек, именованные диапазоны, меню, команды, кнопки и управляющие клавиши клавиатуры (например, Alt или Ctrl). Он же используется для отображения команд, настроек, переключателей, переменных, атрибутов, клавиш, функций, типов, классов, пространств имен, методов, модулей, свойств, параметров, значений, объектов, событий, обработчиков событий, тегов XML и HTML, макросов, содержимого файлов и результатов команд, адресов Интернета (URL) и электронной почты, имен и расширений файлов, путей к файлам, каталогов и переменных в тексте.

Курсив

Показывает новые термины или обращает внимание на важные сведения. Используется в примерах, таблицах и командах, показывая текст, который пользователь должен заменить подходящими значениями.

Моноширинный шрифт

Позволяет выделить листинги.

Стоит уделить особое внимание вынесенным из основного текста замечаниям (врезкам), обозначенным следующим образом:

ПРИМЕЧАНИЕ

Так обозначаются советы, предложения или общие замечания.

ВНИМАНИЕ

Так обозначается предупреждение или предостережение.

Использование примеров кода

Задача этой книги — помочь вашей работе. Вы можете использовать код из этой книги в ваших программах и документации. Для этого не нужно обращаться к нам за разрешением, если только вы не воспроизводите существенную часть кода. Например, применение при написании программы нескольких фрагментов кода из этой книги не требует разрешения. Продажа или распространение компакт-диска с примерами из книг O'Reilly *требует* получения разрешения. Ответ на вопрос со ссылкой на эту книгу и цитирование примера кода не требует разрешения. Внедрение существенного объема кода из этой книги в документацию по вашему продукту *требует* разрешения.

Мы рекомендуем, но не требуем указывать авторство. Обычно упоминается название, автор, издательство и ISBN. Например, *Excel Hacks: 100 Industrial-Strength Tips and Tools*, by David and Raina Hawley. Copyright 2004 O'Reilly & Associates, Inc., 0-596-00625-X.

Если вам кажется, что использование примеров кода выходит за рамки перечисленных разрешений, не стесняйтесь обращаться к нам по адресу: permissions@oreilly.com.

От издательства

Ваши замечания, предложения, вопросы отправляйте по адресу электронной почты: comp@piter.com (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

Все исходные тексты, приведенные в книге, можно найти по адресу: <http://www.piter.com/download>.

На сайте издательства <http://www.piter.com> вы найдете подробную информацию о наших книгах.

Упрощение рабочих книг и листов

Трюки с 1 по 15

Пользователи Excel знают, что рабочие книги — это мощное модельное представление. Но многие пользователи в равной мере знают, что рабочие книги могут вызвать огромное количество заговоздок и помех. Трюки этой главы помогут избежать некоторых из этих заминок, одновременно давая преимущества более эффективных, но зачастую неизвестных способов управления рабочими книгами.

Перед тем как перейти к делу, стоит вкратце повторить основы, которые упростят создание эффективных трюков. Excel — это мощное приложение для работы с электронными таблицами, и вы можете делать в нем совершенно невероятные вещи. К сожалению, многие люди, разрабатывая свои таблицы в Excel, не смотрят в будущее, из-за чего усложняется дальнейшее использование и обновление столь тщательно спроектированной таблицы. В этом разделе мы дадим несколько советов, следуя которым, вы обеспечите максимальную эффективность создаваемых таблиц.

Т Р Ю К
№ 1

Создание персональных представлений рабочих книг

Excel позволяет одновременно раскрывать несколько рабочих книг и настраивать их вид, помещая книги в различные окна. Вы можете сохранять представления рабочих областей в виде файлов .xltw и открывать именно те представления, которые нужны в данный момент.

Иногда при работе в Excel необходимо открыть на экране несколько рабочих книг. Это упрощает использование и просмотр данных из нескольких книг. Далее мы рассмотрим, как сделать это аккуратно и организованно.

Раскройте все рабочие книги, которые вам понадобятся.

ПРИМЕЧАНИЕ

Чтобы за один раз открыть несколько рабочих книг, выберите команду **Файл** ▶ **Открыть** (**File** ▶ **Open**) и выделите нужные книги, удерживая клавишу **Ctrl**, а затем щелкните на кнопке **Открыть** (**Open**).

В любой из открытых книг (неважно, в какой) выберите команду **Окно** ▶ **Расположить** (**Window** ▶ **Arrange**). Если установлен флажок **Только окна текущей книги** (**Windows**

of active workbook), сбросьте его, а затем установите нужный переключатель и щелкните на кнопке ОК.

Если выбрать переключатель Рядом (Tiled), ваши книги будут расположены рядом, как черепица на крыше. Это демонстрируется на пустых рабочих книгах рис. 1.1.

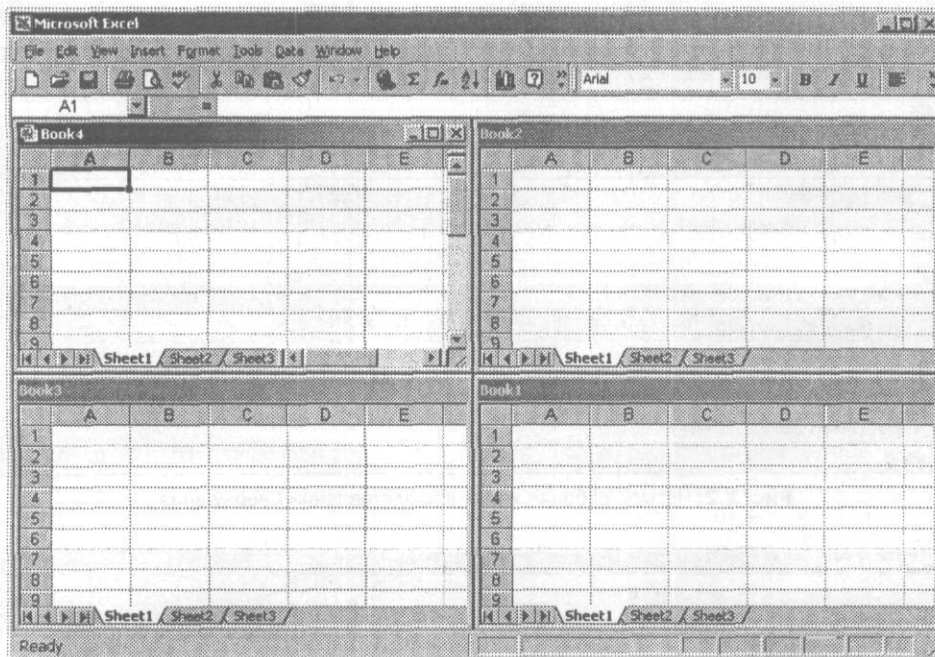


Рис. 1.1. Четыре рабочие книги в представлении Рядом

Выбрав переключатель Сверху вниз (Horizontal), вы расположите книги горизонтально друг над другом (рис. 1.2).

Выберите переключатель Слева направо (Vertical), чтобы расположить книги последовательно, рядом друг с другом (рис. 1.3).

Наконец, переключатель Каскадом (Cascade) (рис. 1.4) позволяет расположить все открытые книги поверх друг друга.

Расположив рабочие книги как вам удобно, вы легко сможете копировать, вставлять, перетаскивать и другими способами изменять в них данные.

Если вы собираетесь повторно использовать созданное представление, его можно сохранить как рабочую область. Для этого выберите команду **Файл** ► **Сохранить рабочую область** (File ► Save Workspace), в поле **Имя файла** (File Name) введите имя файла для рабочей области и щелкните на кнопке ОК. Сохраненный файл будет иметь расширение **.xlw**, а не стандартное расширение **.xls**. Чтобы развернуть одну из книг на всю рабочую область Excel, дважды щелкните на строке заголовка нужной рабочей книги. Можно щелкнуть и кнопку **Развернуть** (Maximize) любого из окон рабочих областей. Закончив работу с книгами, закройте их обычным способом.

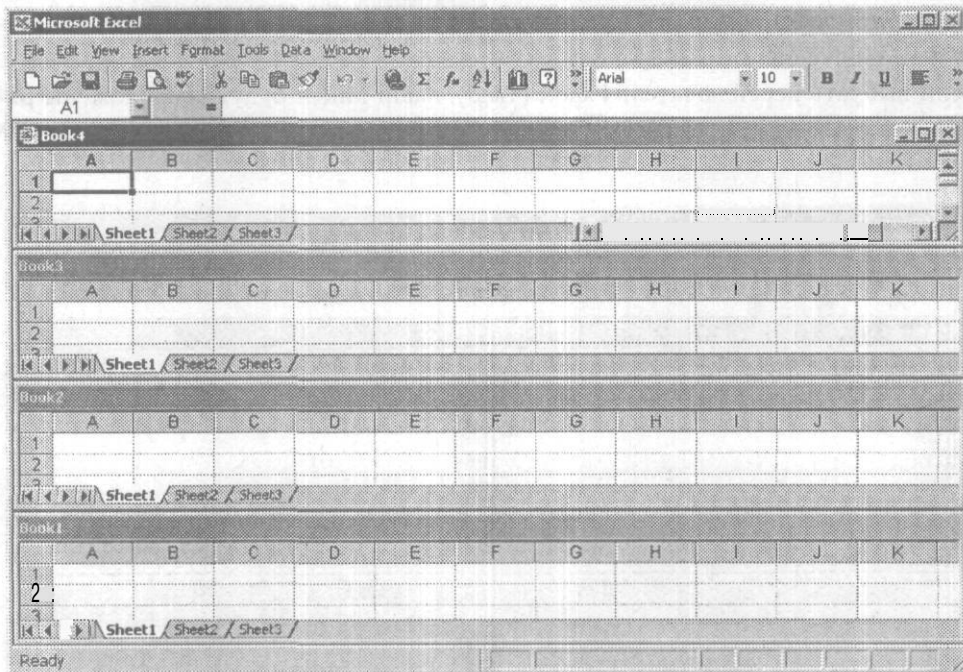


Рис. 1.2. Четыре рабочие книги в представлении Сверху вниз

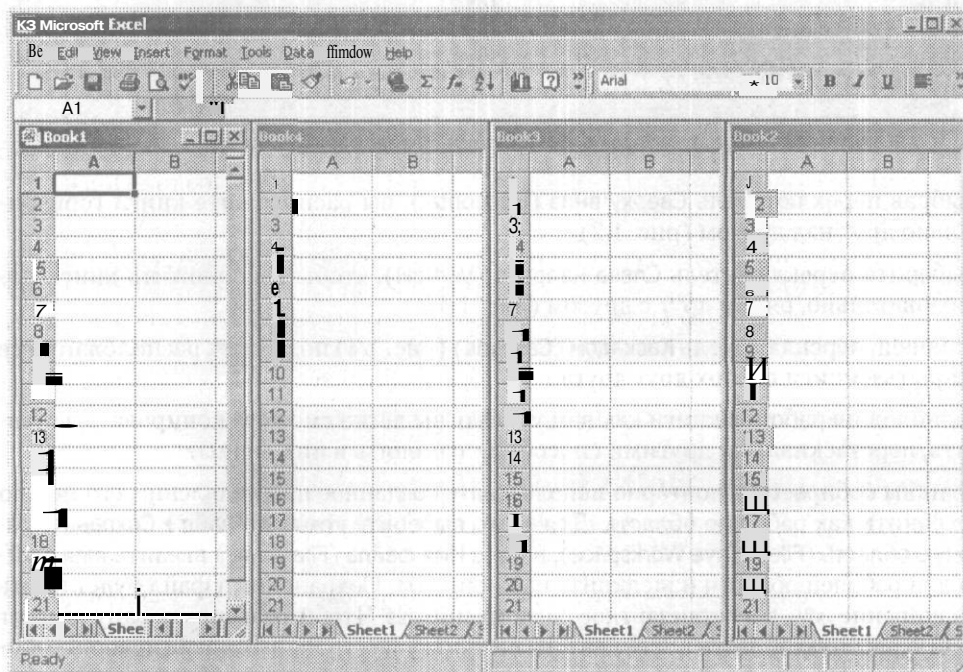


Рис. 1.3. Четыре рабочие книги в представлении Слева направо

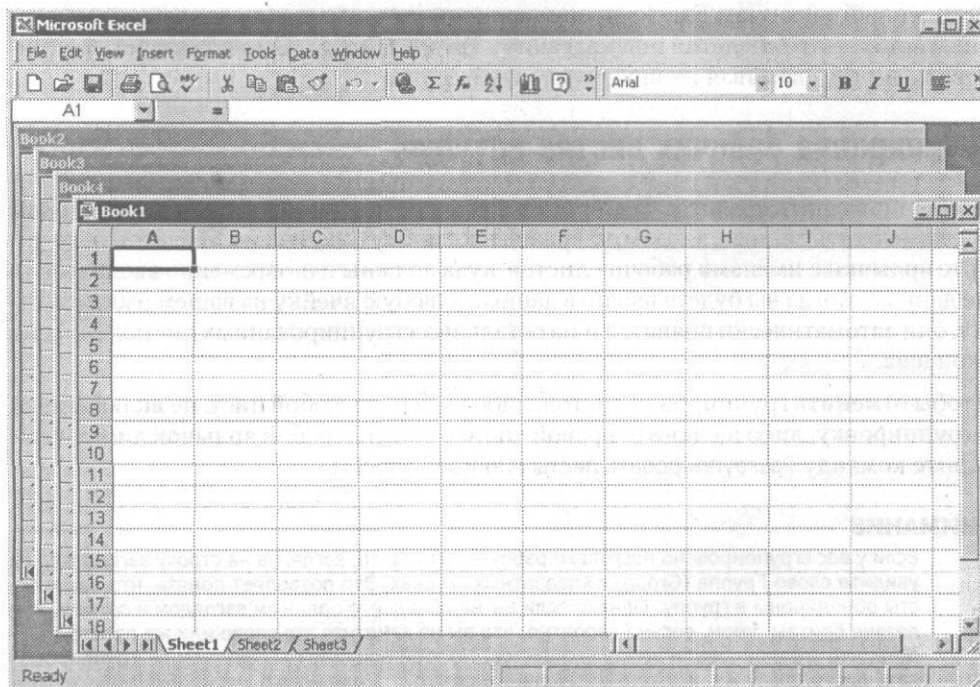


Рис. 1.4. Четыре рабочие книги в представлении Каскадом

Если хотите открыть те же рабочие книги, откройте файл *.xlw, и ранее настроенное представление волшебным образом будет восстановлено для всех книг. Если понадобится только одна из этих рабочих книг, откройте файл обычным способом. Любые изменения, которые вы вносите в рабочие книги в файле *.xlw, будут автоматически сохраняться при закрытии рабочей области. Рабочие книги можно сохранять и по отдельности.

Потратив немного времени на настройку нескольких пользовательских представлений для повторяющихся задач, требующих нескольких открытых рабочих книг, вы обнаружите, что справляться с этими задачами станет проще. Для разных повторяющихся задач можно использовать разные представления, в зависимости от типа задачи или вашего самочувствия в конкретный день.

Т Р Ю К
№ 2

Одновременный ввод данных в несколько рабочих листов

Часто бывает так, что одни и те же данные одновременно содержатся в нескольких рабочих листах. Можно применить инструмент группировки Excel, чтобы данные в одной рабочей книге можно было одновременно вводить в несколько рабочих листов. Кроме того, мы предлагаем более быстрый и гибкий подход, в котором используется пара строчек кода VBA (Visual Basic for Applications).

Встроенный механизм Excel для одновременного размещения данных в нескольких листах — это средство под названием Группа (Group). Оно позволяет группировать листы, связывая их внутри рабочей книги.

Группировка рабочих листов вручную

Чтобы применить средство Группа (Group) вручную, щелкните лист, на котором будете вводить данные, а затем, удерживая клавишу Ctrl (Shift на Macintosh), щелкните ярлычки с именами рабочих листов, куда должны одновременно вводиться те же данные. Когда вы будете вводить данные в любую ячейку на вашем рабочем листе, они автоматически появятся и на остальных сгруппированных листах. Миссия окончена.

Чтобы отменить группировку листов, либо выберите любой лист, не включенный в группировку, либо щелкните правой кнопкой мыши любой ярлычок листа и выберите команду Разгруппировать листы (Ungroup Sheets).

ВНИМАНИЕ

Если у вас сгруппировано несколько рабочих листов, то, взглянув на строку заголовка, вы увидите слово Группа (Group) в квадратных скобках. Это позволяет понять, что ваши листы объединены в группу. Однако если вы не обладаете орлиным взглядом и острым, как лезвие бритвы, умом, весьма вероятно, что вы не заметите это слово или же просто забудете, что сгруппировали несколько листов. По этой причине мы советуем разгруппировывать листы сразу же, как только закончите делать то, для чего была необходима группировка.

Хотя это очень простой метод, он означает, что вам необходимо помнить о необходимости группировать и отменять группировку листов, иначе вы можете случайно заменить или стереть нужные данные с другого рабочего листа. Также это означает, что данные будут одновременно вводиться на другом листе независимо от того, на какой ячейке вы находитесь в данный момент. Например, вам может потребоваться одновременный ввод данных, только если вы находитесь в определенном диапазоне ячеек.

Автоматическая группировка рабочих листов

Эти трудности можно преодолеть при помощи очень простого кода VBA. Чтобы этот код заработал, он должен находиться в частном модуле (private module) для объекта Sheet (Лист). Чтобы быстро перейти в частный модуль, правой кнопкой мыши щелкните ярлычок листа и в контекстном меню выберите команду Исходный текст (View Code). Затем для помещения кода в частный модуль для объекта Лист (Sheet) вы сможете использовать одно из событий листа Excel, то есть событий, которые происходят в вашем листе, например, изменение ячейки, выбор диапазона, активация, деактивация и так далее.

Первое, что необходимо сделать для обработки группировки, — это выбрать имя для диапазона ячеек, который вы хотите сгруппировать, чтобы данные автоматически отображались на других листах. В частный модуль введите код из листинга 1.1.

Листинг 1.1

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Not Intersect(Range("MyRange"), Target) Is Nothing Then
        'Sheet5 намеренно помещен на первое место в списке, так как
        'это активный лист, на котором мы работаем
        Sheets(Array("Sheet5", "Sheet3", "Sheet1")).Select
    Else
        Me.Select
    End If
End Sub
```

В этом коде мы использовали имя диапазона MyRange. (Если вы не знакомы с именованными диапазонами, обратитесь к разделу «Адресация данных по имени» [Трюк № 39].) Измените MyRange на имя диапазона, которое вы используете в своем листе. Кроме того, измените в коде три имени листов на имена листов, которые хотите сгруппировать (рис. 1.5). Закончив, закройте окно модуля или нажмите сочетание клавиш Alt/Apple+Q, чтобы вернуться в Excel.

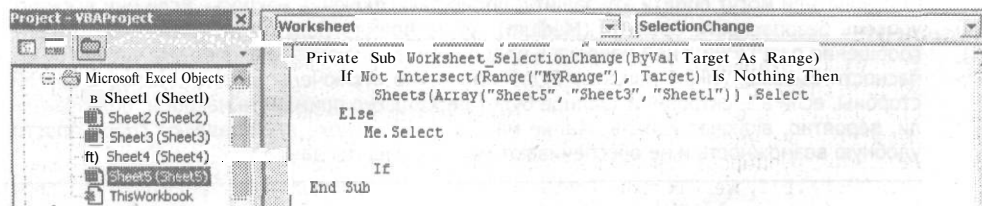


Рис. 1.5. Код автоматической группировки рабочих листов

Важно помнить, что первым в этом массиве должно стоять имя листа, содержащего этот код, то есть листа, на котором вы будете вводить данные. После того как код создан, каждый раз, когда вы будете выбирать ячейку на рабочем листе, код будет проверять, принадлежит ли выбранная ячейка диапазону с именем MyRange. Если да, то код будет автоматически группировать нужные рабочие листы. Если нет, он будет разгруппировывать листы, активируя тот лист, на котором вы в данный момент находитесь. Прелесть этого трюка в том, что вручную группировать листы не нужно, и нет риска забыть разгруппировать их. Этот подход может сэкономить много времени и усилий.

Если вы хотите, чтобы те же данные появлялись на других листах, но не в ячейках с теми же адресами, используйте код из листинга 1.2.

Листинг 1.2

```
Private Sub worksheet_Change(ByVal Target As Range)
    If Not Intersect(Range("MyRange"), Target) Is Nothing Then
        With Range("MyRange")
            .Copy Destination:=Sheets("Sheet3").Range("A1")
            .Copy Destination:=Sheets("Sheet1").Range("D10")
        End With
    End If
End Sub
```

Этот код также необходимо записать в частный модуль объекта Лист (Sheet). Чтобы раскрыть его, выполните действия, описанные ранее в этом трюке.

**Т Р Ю К
№ 3**

Запрещение пользователям выполнять определенные действия

Хотя в Excel реализована общая защита рабочих книг и листов, этот грубый инструмент не может предоставить пользователям ограниченные привилегии — если только вы не примените какие-то трюки.

Управлять взаимодействием пользователей с вашими электронными таблицами можно путем наблюдения и ответов на события. *События* — это действия, которые происходят по мере того, как вы работаете с книгами и листами. Наиболее часто происходящие события — это открытие рабочей книги, ее сохранение и закрытие после завершения работы. Вы можете заставить Excel автоматически выполнять некоторый код Visual Basic в момент, когда происходит одно из этих событий.

ВНИМАНИЕ

Пользователи могут обойти эту защиту, полностью отключив макросы. Если они выберут уровень безопасности Средняя (Medium), то во время открытия рабочей книги увидят сообщение о наличии в ней макроса и предложение отключить макросы. На уровне безопасности Высокая (High) макросы будут полностью отключены автоматически. С другой стороны, если в электронной таблице будет необходимо применять макросы, пользователи, вероятно, включат использование макросов. Эти трюки представляют собой просто удобную возможность и не обеспечивают мощной защиты данных.

Отключение в рабочей книге команды Сохранить как

Можно сделать так, чтобы все рабочие книги сохранялись в режиме «только для чтения». Для этого в настройках Файл ▶ Сохранить (File ▶ Save) нужно установить флажок Только для чтения (Read-only recommended). Так вы запретите пользователю сохранять изменения, которые он может внести в файл. Пользователь сможет сохранить измененный файл только под другим именем или в другой папке.

Иногда, однако, необходимо запрещать пользователям сохранять копии вашей рабочей книги в другом каталоге или папке, независимо от того, будет ли использоваться то же имя или другое. Другими словами, нужно разрешить пользователям сохранять файл поверх существующего, но запретить сохранять копию в другом месте. Это особенно удобно, когда несколько человек сохраняют изменения рабочей книги, и вам не придется собирать множество различных копий одной книги с одинаковыми именами в разных папках.

Событие Before Save, которое вы будете использовать в следующем коде, было впервые создано в Excel 97. Как предполагает его имя, это событие происходит перед сохранением рабочей книги, позволяя вам поймать действия пользователя еще перед этим фактом, выдать предупреждение и запретить Excel сохранять файл.

ВНИМАНИЕ

Перед тем как пытаться выполнить следующий трюк, обязательно сначала сохраните рабочую книгу. Если вы напишете этот код до того, как сохраните книгу, то не сможете более сохранять изменения.

Чтобы написать код, откройте рабочую книгу, правой кнопкой мыши щелкните значок Excel слева от меню Файл (File) в строке меню рабочего листа и выберите пункт Исходный текст (View Code) (рис. 1.6).

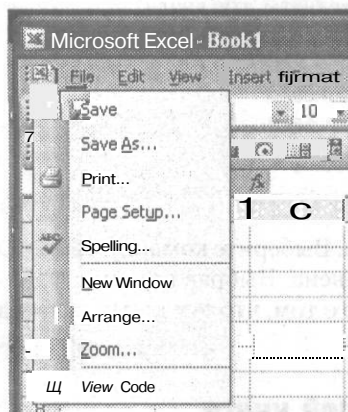


Рис. 1.6. Быстрый доступ через меню к частному модулю для объекта рабочей книги

ПРИМЕЧАНИЕ

На Mac воспользоваться таким быстрым доступом нельзя. Необходимо открыть редактор Visual Basic (Visual Basic Editor, VBE), нажав сочетание клавиш **Option+F11** или выбрав команду **Tools > Macro > Visual Basic Editor**. Открыв редактор, щелкните на нем, удерживая клавишу **Ctrl**, или щелкните правой кнопкой мыши пункт **This Workbook** в окне **Projects**.

Введите в VBE следующий код (рис. 1.7) и нажмите сочетание клавиш **Alt/Apple+Q**, чтобы вернуться в Excel.

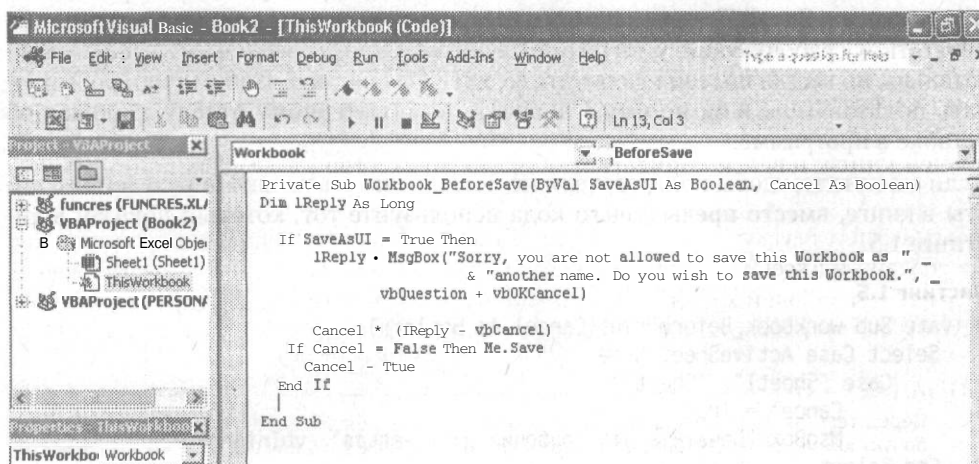


Рис. 1.7. Код в частном модуле ThisWorkbook

Листинг 1.3

```
Private Sub workbook_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)
Dim lReply As Long
If SaveAsUI = True Then
lReply = MsgBox("Нельзя сохранить эту книгу" & _
"под другим именем. Хотите сохранить" & _
"эту книгу?", vbQuestion + vbOKCancel)
Cancel = (lReply = vbCancel)
If Cancel = False Then Me.Save
Cancel = True
End If
End Sub
```

Проверьте, что получилось. Выберите команду **Файл** ▶ **Сохранить** (**File** ▶ **Save**); ваша рабочая книга будет сохранена. Выбрав команду **Файл** ▶ **Сохранить как** (**File** ▶ **Save As**), вы увидите сообщение о том, что эту книгу запрещено сохранять под другим именем.

Запрет печати рабочей книги

Иногда возникает необходимость запретить пользователям печатать рабочую книгу, устранив, таким образом, вероятность попадания ее в мусорную корзину или куда-нибудь на всеобщее обозрение. При помощи события **Excel Before Print** можно отследить и предупредить действия пользователя. Как и ранее, введите в VBE код из листинга 1.4.

Листинг 1.4

```
Private Sub workbook_BeforePrint(Cancel As Boolean)
Cancel = True
MsgBox "Выводить эту рабочую книгу на печать нельзя", vbInformation
End Sub
```

Закончив ввод кода, нажмите сочетание клавиш **Alt/Apple+Q**, чтобы сохранить его и вернуться в Excel. Теперь каждый раз, когда пользователи будут пытаться напечатать эту рабочую книгу, ничего не случится. Строка сообщения **MsgBox** не обязательна, но всегда полезно включать ее, хотя бы для того, чтобы проинформировать пользователя и он не начал докучать компьютерному отделу, сообщая об ошибке в программе.

Если вы хотите, чтобы пользователи не могли печатать только определенные листы в книге, вместо предыдущего кода используйте тот, который показан в листинге 1.5.

Листинг 1.5

```
Private Sub workbook_BeforePrint(Cancel As Boolean)
Select Case ActiveSheet.Name
Case "Sheet1", "Sheet2"
Cancel = True
MsgBox "Печатать этот рабочий лист нельзя", vbInformation
End Select
End Sub
```

Обратите внимание, что печать будет остановлена только для листов Sheet1 и Sheet2. Конечно, имена листов вашей рабочей книги могут быть любыми. Чтобы добавить их в список кода, запишите их в кавычках, разделяя запятыми. Если вы хотите запретить печать только одного листа, укажите только одно название в кавычках, не ставя запятую.

Запрещение добавлять рабочие листы

Excel позволяет защищать структуру рабочей книги, чтобы пользователи не могли удалять листы, изменять их порядок, переименовывать и т. д. Иногда, однако, возникает необходимость запретить пользователям добавлять рабочие листы, одновременно разрешив другие изменения структуры книги.

Сделать это поможет код из листинга 1.6.

Листинг 1.6

```
Private Sub Workbook_NewSheet(ByVal Sh As Object)
    Application.DisplayAlerts = False
    MsgBox "Добавлять рабочие листы в эту книгу нельзя", vbInformation
    Sh.Delete
    Application.DisplayAlerts = True
End Sub
```

Сначала этот код выводит окно с сообщением, а затем, как только пользователь щелкает кнопку ОК, сразу же удаляет только что добавленный лист. Строка `Application.DisplayAlerts = False` запрещает вывод стандартного предупреждения Excel, которое спрашивает, действительно ли пользователь хочет удалить лист. Теперь пользователи не смогут добавлять листы в рабочую книгу.

Еще один способ запретить пользователям добавлять листы — выбрать команду Сервис ► Защита ► Защитить книгу (Tools ► Protection ► Protect Workbook). Удостоверьтесь, что флажок Структуру (Structure) установлен, и щелкните на кнопке ОК. Однако, как мы уже упоминали в начале этого раздела, стандартная защита рабочих листов в Excel — это весьма грубый инструмент, который одновременно блокирует и множество других возможностей Excel.



Запрещение ненужных сообщений

Болтливость Excel может, в конце концов, надоесть — он постоянно просит подтвердить, что вам действительно нужно то действие, которое вы только что попросили выполнить. Давайте прекратим разговоры и призовем Excel к делу.

Мы говорим о таких сообщениях, как, например, вопрос, хотите ли вы включить макросы (когда у вас ни одного макроса нет) или же уверены ли вы, что хотите удалить лист. Далее рассказывается, как раз и навсегда избавиться от большинства часто появляющихся сообщений.

Включение макросов, когда у вас их нет

Excel демонстрирует прекрасную память, когда дело доходит до запоминания факта, что вы записали в книге макрос. К сожалению, он не забывает об этом, даже если вы уже удалили все макросы командой Сервис ► Макрос ► Макросы (Tools ► Macro ► Macros) или сочетанием клавиш Alt/Option+F8. Откройте рабочую книгу заново и вы снова увидите приглашение включить макросы, даже если их уже не осталось.

ПРИМЕЧАНИЕ

Приглашение включить макросы появляется, только если выбран средний уровень безопасности. При низком уровне безопасности макросы будут включаться без вывода предупреждения. На высоком уровне безопасности для максимальной защиты макросы автоматически выключаются.

Когда вы записываете макрос, Excel вставляет модуль Visual Basic, содержащий ваши команды и пользовательские функции. При открытии рабочей книги Excel проверяет наличие таких модулей, пустых или заполненных макросами. При удалении макроса из рабочей книги удаляется только код внутри модуля, но не сам модуль — как если бы, выпив остатки молока, вы поставили пустую коробку обратно в холодильник. Чтобы избежать ненужных приглашений включить макросы, необходимо удалить модуль. Вот как это делается.

Откройте VBE командой Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor) или сочетанием клавиш Alt/Option+F11 и выберите команду View ► Project Explorer. (На Macintosh окно Projects открыто всегда, поэтому Project Explorer открывать не нужно.) Вы увидите окно, как на рис. 1.8.

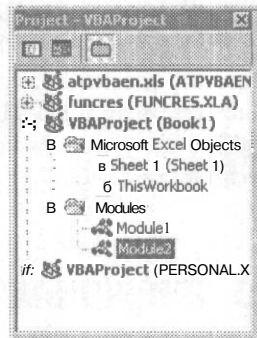


Рис. 1.8. Модули Project Explorer с раскрытой папкой Modules

Найдите в Project Explorer свою рабочую книгу и щелкните значок плюс (+) слева от нее, чтобы развернуть компоненты книги, в частности, модули. Щелкните значок плюс слева от папки Modules, чтобы увидеть все существующие модули.

Правой кнопкой мыши по очереди щелкайте все модули и в контекстном меню выбирайте команду Remove Module. Откажитесь от экспорта модулей. Не удаляйте модули, не подумав. Перед удалением по очереди дважды щелкайте все модули.

ли, чтобы убедиться, что они действительно не нужны. Как обычно, чтобы вернуться обратно в Excel, нажмите сочетание клавиш Alt/Apple+Q.

Приглашение сохранить несуществующие изменения

Вы могли заметить, что иногда, когда вы просто открываете рабочую книгу и просматриваете ее, Excel выдает приглашение сохранить изменения в вашей личной книге макроканд, хотя вы не вносили никаких изменений. Знаете вы это или нет, в вашей личной книге макроканд, вероятно, есть пересчитываемая функция.

Личная книга макросов (personal macro workbook) — это скрытая рабочая книга, созданная, когда вы впервые записывали макрос, и открывающаяся каждый раз вместе с Excel. *Пересчитываемая функция* (volatile function) (или формула) — это функция, которая автоматически пересчитывается, когда вы практически ничего не делаете в Excel, например, просто открываете или закрываете рабочую книгу или все приложение. Две самые распространенные пересчитываемые функции — это Today() и Now().

Так, хотя вручную в рабочей книге вы ничего не изменили, эти пересчитываемые функции, выполняющиеся в фоновом режиме, могли все же внести какие-то изменения. Это считается допустимым и вызывает появление приглашения Excel сохранить подобные невидимые изменения.

Если вы хотите, чтобы Excel прекратил требовать сохранять изменения, которые вы не делали, это можно сделать двумя способами. Самый очевидный — не хранить пересчитываемые функции в личной книге макроканд и удалить уже записанные туда. Или, если они все же нужны, можно применить следующий простой фрагмент кода из листинга 1.7, чтобы обойти проверку, заставив Excel думать, что ваша личная книга макроканд была сохранена в момент открытия.

Листинг 1.7

```
Private Sub workbook_Open( )  
    Me.Saved = True  
End Sub
```

Этот код должен находиться в частном модуле вашей личной книги макроканд. Чтобы открыть его, из любой рабочей книги выберите команду Окно ► Отобразить (Window ► Unhide), затем в диалоговом окне Вывод на экран скрытого окна книги (Unhide Workbook) выберите книгу Personal.xls и щелкните на кнопке ОК. Откройте редактор VBE и введите предыдущий код. Закончив, нажмите сочетание клавиш Alt/Apple+Q, чтобы вернуться в Excel.

Конечно, если у вас есть переменная функция, которая должна пересчитываться, и вы хотите сохранить изменения, необходимо сказать об этом Excel явно, при помощи кода из листинга 1.8.

Листинг 1.8

```
Private Sub workbook_Open( )  
    Me.Save  
End Sub
```

Этот макрос будет автоматически сохранять вашу личную книгу макроккоманд при каждом ее открытии.

Запрещение предупреждений и приглашений Excel для записанных макросов

Один из многих недостатков записанных макросов заключается в том, что, хотя они довольно хорошо имитируют практически любую команду, они обычно забывают ваши ответы на приглашения. Удалите рабочий лист и увидите просьбу подтвердить команду; запустите макрос, делающий то же самое, и также увидите приглашение. Давайте выключим эти приглашения.

Выберите команду Сервис ► Макрос ► Макросы (Tools ► Macro ► Macros) (Alt/Option+F8), чтобы вывести список ваших макросов. Удостоверьтесь, что в списке Находится в (Macros In) выбран пункт Все открытые книги (All Open Workbooks). Выберите нужный макрос и щелкните кнопку Изменить (Edit). Поместите курсор перед первой строкой кода, то есть перед первой строкой, в начале которой не стоит апостроф, и введите следующую строку:

```
Application.DisplayAlerts = False
```

В самом конце кода добавьте следующую строку:

```
Application.DisplayAlerts = True
```

Теперь ваш макрос должен выглядеть так, как это показано в листинге 1.9.

Листинг 1.9

```
Sub MyMacro( )
    ' макрос MyMacro
    ' удаляет активный рабочий лист

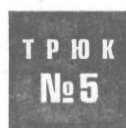
    Application.DisplayAlerts = False
    ActiveSheet.Delete
    Application.DisplayAlerts = True
End Sub
```

Обратите внимание, что в конце макроса вы снова включили вывод сообщений, чтобы во время работы в Excel выводились стандартные сообщения и приглашения Excel. Если этого не делать, то вы не увидите вообще никаких сообщений, даже тех, которые бывают весьма полезны.

ВНИМАНИЕ

Если выполнение вашего макроса по какой-то причине не завершится, например, из-за ошибки выполнения, Excel может никогда не добраться до строки кода, включающей предупреждения. В этом случае, вероятно, будет полезно закрыть и заново запустить Excel, чтобы вернуться к нормальному состоянию дел.

Теперь вы знаете, как работать в Excel, не натываясь на предупреждения и приглашения. Однако помните, что для создания этих сообщений существовали определенные причины. Удостоверьтесь, что вы полностью понимаете назначение сообщения, перед тем как отключить его.



Как скрыть лист, чтобы его невозможно было отобразить

Иногда требуется место для информации, которую пользователи не смогут прочитать или изменить. Встройте тайное место в рабочую книгу, место, где вы сможете хранить данные, формулы и прочие мелочи, используемые, но невидимые на ваших листах.

При настройке новой книги полезно резервировать один лист для хранения информации, которую пользователи не увидят, например, для вычисления формул, проверки данных, списков, полезных переменных и специальных значений, важных данных и т. п. Хотя этот лист можно скрыть, выбрав команду **Формат** ▶ **Лист** ▶ **Скрыть** (**Format** ▶ **Sheet** ▶ **Hide**), лучше гарантировать, что пользователи не смогут отобразить лист, выбрав команду **Формат** ▶ **Лист** ▶ **Отобразить** (**Format** ▶ **Sheet** ▶ **Unhide**).

Конечно, можно просто защитить лист. Однако он все так же будет полностью виден — важные данные, страшные формулы и все остальное. Кроме того, вы не сможете защитить от панели инструментов **Формы** (**Forms**) ячейку, связанную с любым доступным элементом управления.

Вместо этого мы поиграем со свойством рабочего листа **Visible**, присвоив ему значение **xlVeryHidden**. В **VBE** (**Сервис** ▶ **Макрос** ▶ **Редактор Visual Basic (Tools** ▶ **Macro** ▶ **Visual Basic Editor)** или **Alt/Option+F11**) откройте окно **Project Explorer** командой **View** ▶ **Project Explorer**. Найдите там имя своей книги и раскройте ее иерархию, щелкнув знак «плюс» слева от имени книги. Откройте папку **Microsoft Excel Objects**, чтобы увидеть все листы книги.

В **Project Explorer** выберите лист, который хотите скрыть, и откройте его свойства, выбрав команду **View** ▶ **Properties Window** (или нажав клавишу **F4**). Удостоверьтесь, что выбрана вкладка **Alphabetic** и взгляните на свойство **Visible** в самом низу. Щелкните справа поле со значением свойства **Visible** и выберите последнее значение, **2 - xlSheetVeryHidden** (рис. 1.9). Нажмите сочетание клавиш **Alt/Apple+Q**, чтобы сохранить изменения и вернуться в Excel. Этот лист не будет виден в окне Excel и не появится, если воспользоваться командой **Формат** ▶ **Лист** ▶ **Отобразить** (**Format** ▶ **Sheet** ▶ **Unhide**).

ВНИМАНИЕ

После того как вы выбрали **2 - xlSheetVeryHidden** в окне свойств **Properties**, может показаться, что этот выбор не имел никакого воздействия. Это обман зрения, который иногда происходит в Excel. Если этот лист не перечислен в списке **Формат** ▶ **Лист** ▶ **Отобразить** (**Format** ▶ **Sheet** ▶ **Unhide**), значит, желаемый эффект достигнут.

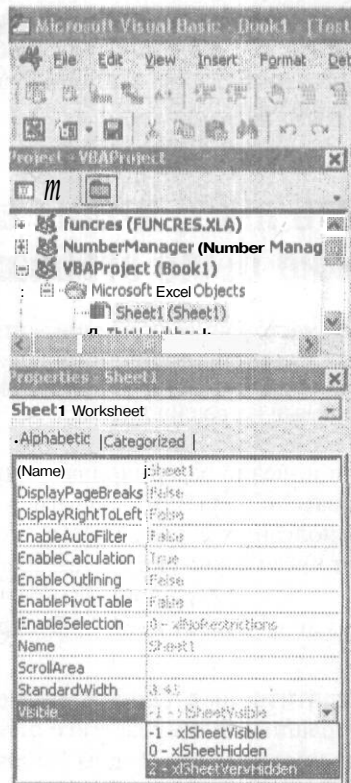


Рис. 1.9. Окно свойств листа; для свойства Visible выбрано значение 2 — xlSheetVeryHidden

Чтобы добиться обратного результата, выполните те же шаги, но в этот раз выберите свойство -1 - xlSheetVisible.

ТРЮК
№ 6

Настройка диалогового окна шаблонов и рабочей книги по умолчанию

Если вы часто выполняете одни и те же задачи или используете один и тот же макет таблицы, можете сконструировать собственную вкладку Шаблон (Template) в стандартном диалоговом окне Шаблон (New) Excel, создав для себя удобную отправную точку.

Представьте, что у вас есть электронная таблица, содержащая дни года, и формулы, вычисляющие различные данные для этих дней. Вы прекрасно отформатировали ее, применив цвета и логотип вашей компании и вставив формулы, и эту таблицу вы будете использовать ежедневно. Чтобы не изобретать каждый день колесо (или копировать и удалять ненужные данные), вы можете сэкономить много сил и времени, создав шаблон.

Шаблоны Excel для листов и книг обеспечивают вам хороший старт для новых проектов, позволяя пропустить шаги по первоначальной настройке, форматированию, построению формул и т. д. Чтобы сохранить шаблонный лист, нужно всего лишь открыть новую книгу, удалить все, за исключением одного листа, а затем создать базовый шаблон, который вы и будете использовать. Закончив, выберите команду **Файл** ▶ **Сохранить как** (File ▶ Save As) и в списке Тип файла (Type) диалогового окна Сохранение документа (Save As) выберите **Шаблон** (Template). Если вы хотите создать шаблон рабочей книги, которая должна содержать несколько листов, снова создайте новую книгу, внесите все необходимые изменения и сохраните ее как шаблон командой **Файл** ▶ **Сохранить как** (File ▶ Save As).

Сохранив шаблон, вы сможете создавать его копии. Для этого нужно выполнить команду **Файл** ▶ **Создать** (File ▶ New) и выбрать шаблон рабочей книги или правой кнопкой мыши щелкнуть ярлычок листа и в контекстном меню выбрать команду **Добавить** (Insert), а затем вставить из шаблона новый лист. Разве не здорово было бы иметь эти шаблоны прямо в стандартном диалоговом окне Excel **Добавить шаблон** (Insert Template) или задавать желаемую книгу книгой по умолчанию? Это можно сделать, создав собственную вкладку **Шаблон** (Template).

ВНИМАНИЕ

Для этого трюка на вашем компьютере должна быть установлена одна копия Excel. Если у вас установлено несколько копий или версий Excel, он может не работать.

Создание собственной вкладки Шаблон

Если у вас огромное количество шаблонов — книг, листов или и того и другого, — которые вы регулярно используете, их можно сгруппировать прямо в диалоговом окне **Добавить** (Insert).

В любой книге выберите команду **Файл** ▶ **Сохранить как** (File > Save As), затем в списке Тип файла (Files of Type) выберите **Шаблон (*.xlt)** (Template (*.xlt)). По умолчанию Excel запишет шаблон в стандартную папку, где хранятся созданные вами шаблоны. Если у вас еще нет такой папки, создайте ее, нажав кнопку **Создать папку** (New Folder).

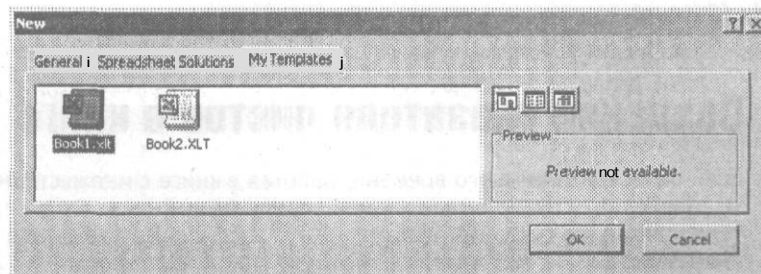


Рис. 1.10. Диалоговое окно Templates

Выберите команду **Файл** ▶ **Создать** (File ▶ New). В Excel 2000 и более поздних версиях в раскрывшемся диалоговом окне **Создание документа** (New Workbook)

перейдите на вкладку Общие (General Templates). Теперь в открывшемся диалоговом окне вы должны увидеть только что созданную вкладку (на рис. 1.10 это вкладка My Templates). Если вы сохраняли шаблоны книг и листов в эту папку, они также должны появиться в диалоговом окне.

Работа с пользовательской книгой по умолчанию

При запуске Excel раскрывается пустая книга по умолчанию, которая называется Книга1 (Book1) и содержит три пустых листа. Это прекрасно, если каждый раз вы хотите все начинать с нуля. Однако если вы такой же человек, как мы, то, вероятно, предпочитаете одну книгу остальным. Так, для нас запуск Excel включает отмену книги по умолчанию и поиск нужной нам обычной книги. **Определенно**, было бы удобнее, чтобы нужная книга открывалась каждый раз сама и была готова к работе.

Для этого необходимо сохранить книгу по умолчанию (шаблон) в папке XLSTART (обычно она находится в C:\Documents and Settings\Владелец\Application Data\Microsoft\Excel\XLSTART в Windows и в Applications/Microsoft Office X/Office/Startup/Excel на Mac OS X). После этого Excel по умолчанию будет открывать ту книгу или книги, которые найдет в этой папке.

ПРИМЕЧАНИЕ

В папке XLSTART создается и хранится ваша личная книга макросов, когда вы записываете макрос. Личная книга макросов — это скрытая книга. Открывать собственные скрытые книги можно и в фоновом режиме. Для этого нужно открыть нужную книгу, выбрать команду Окно ► Скрыть (Window ► Hide), закрыть Excel и щелкнуть на кнопке Да (Yes), чтобы сохранить изменения в скрытой книге. После этого необходимо скопировать эту книгу в папку XLSTART. Все книги, которые вы скрываете и помещаете в папку XLSTART, будут открываться как скрытые книги при каждом запуске Excel.

Не следует помещать в эту папку слишком много книг, особенно больших, так как все они будут открываться при запуске Excel. Слишком большое количество открытых книг может существенно снизить производительность Excel.

Конечно, если вы решите перейти обратно к пустой книге по умолчанию, нужно будет всего лишь удалить соответствующую книгу или шаблон книги из папки Автозагрузка (Startup).

ТРЮК
№7

Создание указателя листов в книге

Если вы потратили много времени, работая в книге с множеством листов, то знаете, как трудно бывает найти определенный лист. Страница указателя, связанная со всеми листами, — это необходимая вещь для упрощения навигации.

Страница указателя позволит быстро и просто перемещаться по книге одним щелчком мыши, и вы без всяческой неразберихи будете попадать куда нужно. Указатель можно создать двумя способами.

Возможно, вам покажется привлекательной идея создать указатель вручную. Создайте новый лист, назовите его Index или подобным именем, введите имена всех остальных листов и свяжите их гиперссылками с соответствующими листами, выбрав команду Вставка ► Гиперссылка (Insert ► Hyperlink) или нажав сочетание клавиш **Ctrl/Apple+K**. Хотя этого способа, вероятно, достаточно для книг с небольшим количеством листов, которые будут меняться нечасто, можно все же запутаться, поддерживая указатель вручную.

Следующий код автоматически создаст указатель с гиперссылками, щелкая которые, можно будет переходить ко всем листам в книге. Указатель создается заново каждый раз при активации содержащего его листа.

Этот код нужно поместить в частный модуль для объекта Sheet. Вставьте в вашу книгу новый лист и назовите его удобным именем — например, Index. Правой кнопкой мыши щелкните ярлычок листа с указателем и в контекстном меню выберите команду Исходный текст (View Code). Введите код Visual Basic (Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor) или **Alt/Option+F11**), приведенный в листинге 1.10.

Листинг 1.10

```
Private Sub Worksheet_Activate( )
Dim wSheet As Worksheet
Dim l As Long
l = 1
With Me
.Columns(1).ClearContents
.Cells(1, 1) = "INDEX"
.Cells(1, 1).Name = "Index"
End With

For Each wSheet In Worksheets
If wSheet.Name <> Me.Name Then
l = l + 1
With wSheet
.Range("A1").Name = "Start" & wSheet.Index
.Hyperlinks.Add Anchor:=.Range("A1"), Address:="", SubAddress:= _
"Index", TextToDisplay:="Назад куказателю"
End With
Me.Hyperlinks.Add Anchor:=Me.Cells(1, 1), Address:="",
SubAddress:="Start" & wSheet.Index, TextToDisplay:=wSheet.Name
End If
Next wSheet
End Sub
```

Нажмите сочетание клавиш **Alt/Apple+Q**, чтобы вернуться обратно к рабочей книге, и сохраните изменения. Обратите внимание в коде, что ячейке A1 на каждом листе *присваивается имя Start* (таким образом вы *присваиваете имя* ячейке или диапазону ячеек в Excel), а также уникальное целое число, представляющее номер листа в указателе. Это гарантирует, что у ячейки A1 каждого листа будет уникальное имя. Если у ячейки A1 на вашем листе уже есть имя, необходимо каждое

упоминание ячейки A1 в коде изменить на что-то более подходящее — например, любую неиспользуемую ячейку листа.

ВНИМАНИЕ

Помните, что если выбрать команду Файл ► Свойства > Документ (File ► Properties ► Summary) и ввести в качестве базы гиперссылки URL, возможно, указатель, созданный предыдущим кодом, не будет работать. База гиперссылки — это путь или URL, который удобно использовать для всех присутствующих в текущем документе гиперссылок с тем же базовым адресом.

Другой, более удобный способ конструирования указателя — добавить ссылку к списку листов как элемент стандартного контекстного меню кнопок прокрутки ярлычков (рис. 1.11)¹.

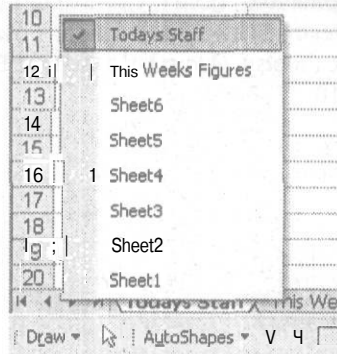


Рис. 1.11. Контекстное меню кнопок прокрутки ярлычков

Чтобы контекстное меню кнопки прокрутки ярлычков открывалось по щелчку правой кнопки мыши на любой ячейке, введите код в VBE, приведенный в листинге 1.11.

Листинг 1.11

```
Private Sub Workbook_SheetBeforeRightClick(ByVal Sh As Object,
    ByVal Target As Range, Cancel As Boolean)
Dim cCont As CommandBarButton
On Error Resume Next
Application.CommandBars("Cell").Controls("Sheet Index").Delete
On Error GoTo 0
Set cCont = Application.CommandBars("Cell").Controls.Add _
    (Type:=msoControlButton, Temporary:=True)
With cCont
    .Caption = "Sheet Index"
    .OnAction = "IndexCode"
End With
End Sub
```

¹ Собственно, а почему бы не ограничиться этим контекстным меню? — *Примеч. ред.*

Затем необходимо вставить стандартный модуль, который будет содержать макрос `IndexCode`, вызываемый предыдущим кодом, когда пользователь щелкает ячейку правой кнопкой мыши. Обязательно используйте стандартный модуль, так как помещение кода в тот же модуль, где находится `Workbook_SheetBeforeRightClick`, будет означать, что Excel не сможет найти макрос с именем `IndexCode`.

Выберите команду `Insert ▶ Module` и введите код из листинга 1.12.

Листинг 1.12

```
Sub IndexCode()  
Application.CommandBars("workbook Tabs").ShowPopup  
End Sub
```

Нажмите сочетание клавиш `Alt/Apple+Q`, чтобы вернуться обратно в окно Excel.

Теперь, щелкнув правой кнопкой мыши любую ячейку на листе, вы увидите новый элемент меню с именем `Sheet Index`, который позволит раскрыть список всех листов в рабочей книге.



Ограничение диапазона прокрутки листа

Если вы постоянно перемещаетесь по своей электронной таблице или же хотите, чтобы читатели могли просматривать определенные данные, удобно ограничить видимую область электронной таблицы только необходимыми данными.

Все листы Excel, созданные в Excel 97 и более поздних версиях, имеют 256 столбцов (от A до IV) и 65 536 строк. Чаще всего на листе используется лишь небольшая доля доступных ячеек. Немного усилий — и вы сможете ограничить область прокрутки листа только ячейками, содержащими нужные данные. Затем можно поместить данные, которые пользователь видеть не должен, в ячейки за пределами области прокрутки. Это также упростит переход по ячейкам на листе, так как часто возникает ситуация, когда пользователь неожиданно оказывается на 50 000 строке и начинает кричать, что на листе совершенно невозможно найти необходимые данные.

Самый простой способ установить границы — просто спрятать все неиспользуемые столбцы и строки. На своем листе найдите последнюю строку, содержащую данные, и целиком выделите строку под ней, щелкнув кнопку строки. Удерживайте клавиши `Ctrl` и `Shift` и нажимайте стрелку вниз, чтобы выделить все строки ниже. Чтобы спрятать их, выберите команду `Формат > Строка ▶ Скрыть (Format ▶ Row ▶ Hide)`. То же самое сделайте и с неиспользуемыми столбцами: найдите последний столбец с данными, полностью выделите столбец справа от него, удерживайте клавиши `Ctrl` и `Shift`, нажимая стрелку вправо, а затем выберите команду `Формат ▶ Столбец ▶ Скрыть (Format > Column ▶ Hide)`. Если все произошло как надо, используемые ячейки будут окружены серой областью, за пределы которой перейти невозможно.

Второй способ определить границы — указать допустимый диапазон в окне `Properties` листа. Правой кнопкой мыши щелкните ярлычок листа в левом нижнем углу

экрана и в контекстном меню выберите пункт View Code (Исходный текст). В Windows выберите команду View ► Project Explorer (Ctrl+R), а в Mac OS X нажмите сочетание клавиш Apple+R, чтобы открыть окно Project Explorer. Если окно Properties не открыто, нажмите клавишу F4. Выберите нужный лист, и перейдите к свойству ScrollArea в окне Properties.

Теперь в Project Explorer выберите лист, область прокрутки которого хотите ограничить, а затем в окне Properties (рис. 1.12) перейдите к свойству ScrollArea. В соответствующем поле значений справа введите желаемые границы листа, например, \$A\$1:\$G\$50.

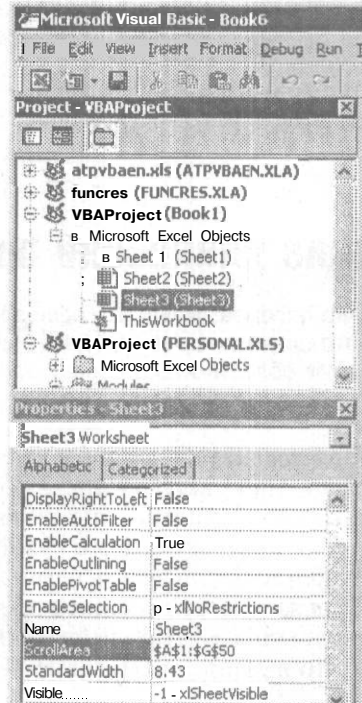


Рис. 1.12. Окно Properties в Project Explorer

Вы не сможете прокручивать лист за пределы указанной области. К сожалению, после закрытия Excel не сохранит этот параметр. Это означает, что для того, чтобы автоматически выбирать желаемый диапазон прокрутки, необходимо создать простой макрос, поместив немного кода в событие worksheet_Activate.

Правой кнопкой мыши щелкните ярлычок с именем того листа, где нужно ограничить диапазон прокрутки, и в контекстном меню выберите команду Исходный текст (View Code), затем введите код из листинга 1.13.

Листинг 1.13

```
Private Sub Worksheet_Activate ()
Me.ScrollArea = "A1:G50"
End Sub
```

Как обычно, нажмите сочетание клавиш **Alt/Apple+Q**, чтобы вернуться в Excel и сохранить книгу.

Хотя вы не сможете увидеть результат в виде серой области в первом способе, вы все же не сможете прокрутить лист или выбрать что-либо за пределами заданной области.

ВНИМАНИЕ

Любой макрос, выбирающий диапазон за пределами этой области прокрутки (включая выбор целых строк и столбцов), сделать это более не сможет. Это относится, в частности, к записанным макросам, так как в них зачастую применяется выделение.

Если макрос использует диапазон за пределами области прокрутки, можно легко изменить его, чтобы он не был ограничен в своей работе каким-то диапазоном ячеек. Выберите команду **Сервис ▶ Макрос ▶ Макросы (Tools ▶ Macro ▶ Macros)** или сочетание клавиш **Alt+F8**, выберите имя макроса, выделите его и щелкните кнопку **Изменить (Edit)**. Вставьте следующую строку перед первой строкой существующего кода:

```
ActiveSheet.ScrollArea = ""
```

На самую последнюю строку кода поместите следующее:

```
ActiveSheet.ScrollArea = "$A$1:$G$50"
```

Теперь код должен выглядеть приблизительно так, как показано в листинге 1.14.

Листинг 1.14

```
Sub MyMacro()  
' макрос MyMacro  
' макрос записан 19/9/2003 в OzGrid.com
```

```
ActiveSheet.ScrollArea = ""  
Range("Z100").Select  
Selection.Font.Bold = True  
ActiveSheet.ScrollArea = "$A$1:$G$50"  
Sheets("Daily Budget").Select  
ActiveSheet.ScrollArea = ""  
Range("T500").Select  
Selection.Font.Bold = False  
ActiveSheet.ScrollArea = "$A$1:$H$25"
```

```
End Sub
```

Наш записанный макрос выбирает ячейку Z100 и форматирует ее, делая шрифт жирным. Затем он выбирает лист с именем Daily Budget, выбирает на нем ячейку T500 и отменяет жирный шрифт. Мы добавили строку `ActiveSheet.ScrollArea = ""`, поэтому на листе можно выбрать любую ячейку, а затем восстановить желаемый ограниченный диапазон прокрутки. Выбирая другой лист (Daily Budget), мы снова позволяем коду выбирать любую ячейку на этом листе, а затем восстанавливаем ограниченный диапазон.

Третий метод, наиболее гибкий, автоматически ограничивает область прокрутки используемым диапазоном ячеек на листе, для которого вы создаете следующий код. Чтобы применить этот способ, правой кнопкой мыши щелкните ярлычок с именем нужного листа, в контекстном меню выберите пункт Исходный текст (View Code) и введите код, приведенный в листинге 1.15.

Листинг 1.15

```
Private Sub Worksheet_Activate()  
    Me.ScrollArea = Range(Me.UsedRange, Me.UsedRange(2,2)).Address  
End Sub
```

Теперь нажмите сочетание клавиш Alt/Apple+Q или щелкните крестик в правом верхнем углу окна, чтобы вернуться в Excel и сохранить книгу.

Этот макрос будет выполняться автоматически каждый раз, когда вы будете активировать лист, где он хранится. Однако некоторые проблемы могут возникнуть, если понадобится ввести данные за пределами текущего используемого диапазона. Чтобы избежать этого, примените стандартный макрос, который восстановит область прокрутки, равную полному листу. Выберите команду Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor), затем выберите команду Insert ► Module и введите код из листинга 1.16.

Листинг 1.16

```
Sub ResetScrollArea()  
    ActiveSheet.ScrollArea = ""  
End Sub
```

Теперь нажмите сочетание клавиш Alt/Apple+Q или щелкните крестик в правом верхнем углу окна, чтобы вернуться в Excel и сохранить книгу.

Если хотите, можно упростить способ выполнения макроса, назначив ему сочетание клавиш. Выберите команду Сервис ► Макрос ► Макросы (Tools ► Macro ► Macros) или нажмите сочетание клавиш (Alt/Option+F8). Выберите ResetScrollArea (это имя вашего макроса), щелкните кнопку Параметры (Options) и выберите сочетание клавиш.

Каждый раз, когда понадобится добавить данные за пределами установленных границ листа, выполняйте макрос ResetScrollArea для переопределения границ. После выполнения макроса внесите изменения, которые было невозможно сделать с ограниченным диапазоном прокрутки. Закончив, активируйте любой другой лист, а затем активируйте только что измененный лист. Это действие запустит код, ограничивающий диапазон прокрутки желаемой областью.



Блокировка и защита ячеек, содержащих формулы

Иногда вы хотите разрешить пользователям изменять ячейки, содержащие данные, не разрешая менять формулы. Можно заблокировать ячейки, содержащие формулы, не защищая целый лист или книгу.

При создании электронной таблицы большинство из нас используют определенного рода формулы. Иногда, однако, мы не хотим, чтобы другие пользователи портили, удаляли или переписывали любые формулы, присутствующие в таблице. Самый простой и распространенный способ запрещения модификации формул — защитить лист. Однако защита листа не только мешает пользователям изменять формулы, она также подразумевает, что пользователи не смогут ввести или изменить никакие данные. Обычно вы не хотите заходить так далеко,

По умолчанию все ячейки на листе заблокированы; однако это не имеет никакого эффекта, пока не применяется защита листа. Далее мы расскажем об очень простом способе защиты листа, когда блокируются и защищаются только ячейки с формулами.

Выберите все ячейки на листе, нажав сочетание клавиш **Ctrl/Apple+A** или щелкнув кнопку выделения листа (серый квадрат в месте пересечения столбца А и строки 1). Затем выберите команду **Формат ▸ Ячейки ▸ Защита** (**Format ▸ Cells ▸ Protection**) и сбросьте флажок **Защищаемая ячейка (Locked)**. Щелкните кнопку **ОК**.

Теперь выделите любую ячейку, выберите команду **Правка ▸ Перейти (Edit ▸ Go To)** (**Ctrl+G** или **F5**) и щелкните кнопку **Выделить (Special)**. Вы увидите диалоговое окно выделения группы ячеек (рис. 1.13).

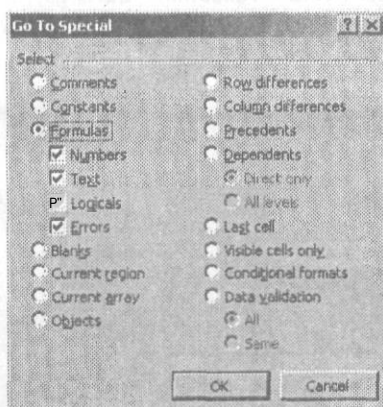


Рис. 1.13. Диалоговое окно Выделение группы ячеек

В диалоговом окне **Go To Special** (Выделение группы ячеек) выберите переключатель **Формулы (Formulas)** и, если необходимо, выберите определенные типы формул, установив соответствующие флажки. Щелкните на кнопке **ОК**. Будут выделены только ячейки с формулами. Выберите команду **Формат ▸ Ячейки ▸ Защита** (**Format ▸ Cells ▸ Protection**) и установите флажок **Защищаемая ячейка (Locked)**. Щелкните на кнопке **ОК**. Теперь выберите команду **Сервис ▸ Защита ▸ Защитить лист** (**Tools ▸ Protection ▸ Protect Worksheet**), чтобы защитить лист и, если необходимо, назначить пароль.

Предыдущий метод определенно экономит много времени и исключает возможные ошибки при поиске формул для защиты. К сожалению, при этом пользователи не смогут воспользоваться несколькими средствами, например, сортировкой,

форматированием, выравниванием текста и многими другими, о которых вы можете и не знать, даже в незаблокированной ячейке. Эту проблему можно решить двумя способами.

Первый подход совершенно не использует защиту листа, зато применяет проверку правильности данных.

ВНИМАНИЕ

Проверка правильности данных далека от идеала, когда дело доходит до предотвращения помещения в ячейки непроверенных данных. Пользователи легко могут вставлять любые скопированные данные в проверенную ячейку и, делая это, удалять пометку о проверке из этой ячейки. Если копируемая ячейка содержит пометку о проверке, то в этом случае новая пометка перекроет исходную.

Чтобы увидеть, о чем мы говорим, выделите любую ячейку, выберите команду Правка ► Перейти (Edit ► Go To) (Ctrl+G или F5) и щелкните кнопку Выделить (Special). В диалоговом окне Выделение группы ячеек (Go To Special) установите переключатель Формулы (Formulas) и, если необходимо, укажите только отдельные типы формул. Щелкните на кнопке ОК.

Теперь, когда выбраны только ячейки с формулами, перейдите на вкладку Данные ► Проверка ► Параметры (Data ► Validation ► Settings), в поле Тип данных (Allow) выберите Другой (Custom), а в поле Формула (Formula) введите "=" (рис. 1.14). Щелкните на кнопке ОК.

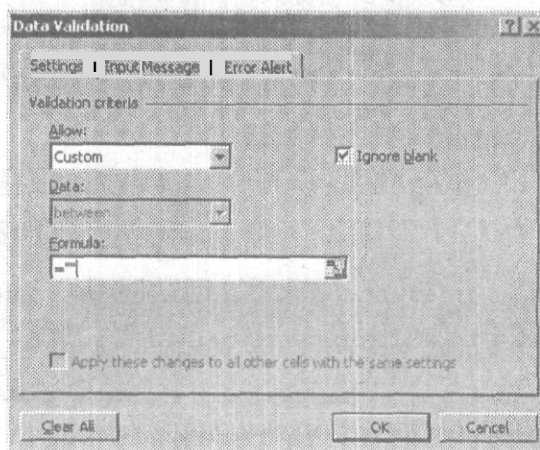


Рис. 1.14. Формулы проверки

Этот метод позволяет предотвратить случайную запись данных в любые ячейки формул, хотя, как подчеркивалось в предыдущем предупреждении, это не абсолютно безопасный способ, и его можно использовать только против случайного ввода данных и т. д. Однако большое преимущество этого метода заключается в том, что на листе можно применять все возможности Excel.

Последний метод позволит использовать все возможности Excel, но только когда вы находитесь в незаблокированной ячейке. Для начала удостоверьтесь, что за-

блокированы только те ячейки, которые вы хотите защитить, а все остальные не заблокированы. Правой кнопкой мыши щелкните ярлычок **листа**, в контекстном меню выберите пункт Исходный текст (View Code) и введите код из листинга 1.17.

Листинг 1.17

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Target.Locked = True Then
        Me.Protect Password:="Secret"
    Else
        Me.Unprotect Password:="Secret"
    End If
End Sub
```

Если вы не хотите использовать пароль, уберите строку Password:="Secret". Если вы указали пароль, измените слово Secret на ваш пароль. Нажмите сочетание клавиш Alt/Apple+Q или щелкните крестик в правом верхнем углу окна, чтобы вернуться в Excel и сохранить книгу. Теперь каждый раз, когда вы будете выбирать заблокированную ячейку, ваш лист будет автоматически защищать себя. Как только вы выберете незаблокированную ячейку, лист автоматически снимет с себя защиту.

ВНИМАНИЕ

Этот трюк работает не идеально, хотя обычно вполне сносно. Ключевое слово, используемое в коде, Target, относится только к ячейке, активной на время выбора. Поэтому важно помнить, что, если пользователь выберет диапазон ячеек (где активные ячейки будут не заблокированы), он сможет удалить весь выделенный диапазон, так как целевая ячейка не заблокирована и, следовательно, лист снимает с себя защиту.



Поиск повторяющихся данных при помощи условного форматирования

Условное форматирование в Excel обычно применяется для определения значения в определенных диапазонах, но мы можем использовать эту возможность в нашем трюке для поиска повторяющихся данных в списке или таблице.

Пользователям часто бывает необходимо найти повторяющиеся данные в списке или таблице. Выполнение этой задачи вручную — чрезвычайно долгая и не защищенная от ошибок процедура. Чтобы упростить ее, можно выполнить трюк с одной из стандартных возможностей Excel условным форматированием.

Возьмите, например, таблицу данных в диапазоне \$A\$1:\$H\$100. Щелкните левую верхнюю ячейку, A1, и, удерживая кнопку мыши, перетащите указатель вниз на H100. Важно, чтобы при выделении активной ячейкой была A1, поэтому перетаскивание указателя от H100 до A1 не будет идентично описанному действию. Выберите команду **Формат** ▶ **Условное форматирование** (Format ▶ Conditional Formatting) и в диалоговом окне **Условное форматирование** (Conditional Formatting) в левом

верхнем раскрывающемся списке выберите пункт Формула (Formula Is). В поле справа введите следующий код:

```
=COUNTIF($A$1:$H$100,A1)>1
```

В русской версии Excel

```
=СЧЁТЕСЛИ($A$1:$H$100,A1)>1
```

Перейдите на вкладку Format (Формат) (в Mac OS X нужно щелкнуть кнопку Format), затем на вкладку Вид (Patterns) и выберите цвет, которым будут выделяться повторяющиеся данные. Щелкните на кнопке ОК, чтобы вернуться в диалоговое окно Условное форматирование (Conditional Formatting) и еще раз щелкните на кнопке ОК, чтобы применить форматирование.

Теперь ячейки, содержащие повторяющиеся данные, должны быть выделены выбранным цветом, что упрощает поиск, удаление, перемещение или другое изменение подобных данных.

Необходимо помнить, что при выделении ячеек активной должна быть ячейка A1, так как адрес ячейки — это относительная, а не абсолютная ссылка, так же, как и ваша таблица данных, \$A\$1:\$H\$100. Используя условное форматирование подобным образом, вы даете Excel знать, какую ячейку необходимо использовать как условие для функции СЧЁТЕСЛИ (COUNTIF). То есть формула условного форматирования в ячейке A1 выглядит как

```
=COUNTIF($A$1:$H$100,A1)>1
```

В русской версии Excel

```
=СЧЁТЕСЛИ($A$1:$H$100,A1)>1
```

тогда как в ячейке A2 она такая

```
=COUNTIF($A$1:$H$100,A2)>1
```

в русской версии Excel

```
=СЧЁТЕСЛИ($A$1:$H$100,A2)>1
```

а в ячейке A3

```
=COUNTIF($A$1:$H$100,A3)>1
```

в русской версии Excel

```
=СЧЁТЕСЛИ($A$1:$H$100,A3)>1
```

и так далее.

Если необходимо увидеть данные, повторяющиеся несколько раз, можно использовать условное форматирование с тремя различными условиями и разными цветовыми кодами для упрощения визуальной идентификации. Для этого выберите ячейку A1 (это ячейка в левом верхнем углу таблицы) и протащите указатель к ячейке H100. И снова важно, чтобы при выделении активной была ячейка A1.

Теперь выберите команду Формат ► Условное форматирование (Format ► Conditional Formatting) и в левом текстовом поле условий вместо пункта Значение (Cell Value Is) выберите пункт Формула (Formula Is). В поле справа от поля Формула (Formula Is) введите следующий код:

```
=COUNTIF($A$1:$H$100,A1)>3
```

В русской версии Excel

```
•СЧЁТЕСЛИ($A$1:$H$100,A1)>3
```

Щелкните кнопку **Формат (Format)**, перейдите на вкладку **Вид (Patterns)** и выберите цвет, который будет идентифицировать данные, встречающиеся более трех раз. Щелкните на кнопке **ОК**, затем на кнопке **А также (Add)**. В группе **Условие 2 (Condition 2)** выберите пункт **Формула (Formula Is)** и введите следующую формулу:

```
=COUNTIF($A$1:$H$100,A1)=3
```

В русской версии Excel

```
-СЧЁТЕСЛИ($A$1:$H$100,A1)=3
```

ПРИМЕЧАНИЕ

Чтобы не вводить формулу заново, выделите ее в поле **Условие 1 (Condition 1)**, нажмите сочетание клавиш **Ctrl/Apple+C**, чтобы скопировать ее, щелкните поле **Формула (Formula)** в группе **Условие 2 (Condition 2)**, нажмите сочетание клавиш **Ctrl/Apple+V**, чтобы вставить ее, и измените условие «>3» на «=3».

Щелкните на кнопке **Формат (Format)**, а затем перейдите на вкладку **Вид (Patterns)** и выберите цвет, который будет идентифицировать данные, появляющиеся три раза. Щелкните на кнопке **ОК**, затем на кнопке **А также (Add)**. В группе **Условие 3 (Condition 3)** выберите пункт **Формула (Formula Is)** и введите следующую формулу:

```
=COUNTIF($A$1:$H$100,A1)=2
```

В русской версии Excel

```
-СЧЁТЕСЛИ($A$1:$H$100,A1)-2
```

Наконец, щелкните кнопку **Формат (Format)** и перейдите на вкладку **Вид (Patterns)**. Выберите цвет для данных, появляющихся два раза. Щелкните на кнопке **ОК**. Теперь цвета ячеек будут различаться в зависимости от того, сколько раз данные появились в таблице.

И снова необходимо помнить, что при выделении ячеек активной должна быть ячейка **A1**, так как адрес ячейки — это относительная, а не абсолютная ссылка, так же, как и ваша таблица данных, **\$A\$1:\$H\$100**. Используя условное форматирование подобным образом, вы даете Excel знать, какую ячейку необходимо использовать как условие для функции **СЧЁТЕСЛИ (COUNTIF)**.



Пользовательские панели инструментов для определенной книги

Хотя большинство панелей инструментов, которые вы создаете, подходят для практически любой задачи, иногда функциональность пользовательской панели инструментов применима только к одной определенной книге. Этот трюк позволит привязывать панели инструментов к соответствующим книгам.

Если вы когда-либо создавали пользовательскую панель инструментов, то, без сомнения, замечали, что она загружается и становится видимой независимо от того, какую книгу вы открыли. А что если пользовательская панель инструментов содержит записанный макрос, предназначенный только для одной специфической рабочей книги? Вероятно, лучше всего было бы привязать специальные пользовательские панели к соответствующим книгам, чтобы исключить беспорядок и неразбериху. Это можно сделать, вставив немного очень простого кода в частный модуль рабочей книги.

Чтобы попасть в частный модуль, правой кнопкой мыши щелкните значок Excel в левом верхнем углу экрана рядом с меню Файл (File), и в контекстном меню выберите пункт Исходный текст (View Code).

ПРИМЕЧАНИЕ -

Этот путь недоступен на Mac. Чтобы открыть редактор VBE (Visual Basic Editor), нужно нажать сочетание клавиш **Option+F11** или выбрать команду **Tools > Macro > Visual Basic Editor**. Открыв редактор, в окне Projects щелкните, удерживая клавишу Ctrl, или щелкните правой кнопкой мыши пункт This Workbook.

Затем введите код из листинга 1.18.

Листинг 1.18

```
Private Sub Workbook_Activate()  
    On Error Resume Next  
    With Application.CommandBars("MyCustomToolbar")  
        .Enabled = True  
        .Visible = True  
    End With  
    On Error GoTo 0  
End Sub  
  
Private Sub Workbook_Deactivate()  
    On Error Resume Next  
    Application.CommandBars("MyCustomToolbar").Enabled = False  
    On Error GoTo 0  
End Sub
```

Измените текст **MyCustomToolbar** на имя вашей пользовательской панели инструментов. Чтобы вернуться обратно в окно Excel, закройте окно модуля или нажмите сочетание клавиш **Alt/Apple+Q**. Если вы откроете или активируете другую книгу, ваша панель инструментов исчезнет и станет недоступной. Снова активируйте соответствующую книгу, и — сюрприз! — панель вернется.

Вы можете даже углубить эту задачу, ограничив пользовательскую панель инструментов только определенным листом в рабочей книге. Правой кнопкой мыши щелкните ярлычок с названием листа, для которого должна быть доступна панель инструментов, и в контекстном меню выберите команду Исходный текст (View Code). Введите код, приведенный в листинге 1.19.

Листинг 1.19

```
Private Sub Worksheet_Deactivate()  
    On Error Resume Next  
        Application.CommandBars("MyCustomToolbar").Enabled = False  
    On Error GoTo 0  
End Sub  
  
Private Sub Worksheet_Activate()  
    On Error Resume Next  
        With Application.CommandBars("MyCustomToolbar")  
            .Enabled = True  
            .Visible = True  
        End With  
    On Error GoTo 0  
End Sub
```

Теперь нажмите сочетание клавиш **Alt/Apple+Q** или закройте окно, чтобы вернуться в Excel.

Первая процедура (**Worksheet_Deactivate**) будет автоматически запускаться каждый раз, когда вы будете покидать выбранный лист, активируя другой. При этом значение свойства **Enabled** вашей пользовательской панели инструментов меняется на значение **False**, запрещая ее отображение. Вторая процедура будет запускаться каждый раз, когда вы будете активировать нужный лист, устанавливая значение свойства **Enabled** равным **True** и отображая пользовательскую панель инструментов. Строка кода **Application.CommandBars(«MyCustomToolbar»).Visible = True** заново выводит вашу панель инструментов, чтобы она стала видна пользователю. Перейдите на другой лист, и панель пропадет; снова активируйте выбранный лист, и панель, как по магии волшебной палочки, появится.

**ТРЮК
№ 12**

Как перехитрить обработчик относительных ссылок Excel

В Excel ссылка в формуле может быть относительной или абсолютной, но иногда вам может понадобиться переместить ячейки с относительными ссылками, не превращая их в абсолютные. Вот как это можно сделать.

Когда формулу нужно сделать абсолютной, в ссылке на ячейку перед номером столбца или строки записывается знак доллара (\$), например, **\$A\$1**. После того как вы это сделали, неважно, куда вы скопируете эту формулу, она будет ссылаться на те же ячейки. Иногда, однако, вы создаете множество формул не с абсолютными, а с относительными ссылками. Обычно при этом, если ячейка с формулой копируется куда-либо в другое место таблицы, ссылки на строки и столбцы меняются соответствующим образом.

Если вы уже создали формулы только с относительными ссылками или, возможно, формулы с набором из относительных и абсолютных ссылок, эти же формулы

можно воспроизвести в другом диапазоне на том же рабочем листе, на другом листе в той же книге или даже на другом листе в другой рабочей книге.

Для этого, не меняя ссылки на диапазоны внутри формул, выберите диапазон ячеек, который хотите скопировать, и выберите команду Правка ▸ Заменить (Edit ▸ Replace). В поле Найти (Find What) введите знак равенства (=), а в поле Заменить на (Replace With) введите знак коммерческого at (@). (Конечно же, вы можете использовать любые символы, которые заведомо не применяются в ваших формулах.) Щелкните на кнопке Заменить все (Replace All). Знаки равенства во всех формулах на листе будут заменены на знак @.

Теперь можно скопировать этот диапазон, вставить его в нужное место, выделить только что вставленный диапазон и выбрать команду Правка ▸ Заменить (Edit ▸ Replace). После этого замените знак @ на знак равенства. Формулы будут ссылаться на те же ячейки, что и в оригинале.

ТРЮК
№ 13

Удаление несуществующих ссылок на книги

О, несуществующие ссылки! Вы открываете рабочую книгу и видите сообщение с просьбой обновить ссылки — но у вас нет никаких ссылок! Как же можно обновить то, что не существует?

Внешние ссылки — это ссылки, указывающие на другие книги. Непредвиденные внешние ссылки могут появиться по различным причинам, начиная от перемещения или копирования диаграмм, листов диаграмм или рабочих листов в другую рабочую книгу. Но знание того, что внешние ссылки существуют, не всегда помогает найти их. Вот несколько способов расправиться с проблемой несуществующих ссылок.

Сначала необходимо узнать, присутствуют ли в книге реальные внешние ссылки (существующие), о которых вы просто забыли. Если вы не уверены, есть ли такие, начните с поиска в самом очевидном месте — в формулах. Для этого удостоверьтесь, что никакие другие книги не открыты и выполните поиск по ключевой фразе [*] в формулах на всех листах. Закройте все остальные книги, чтобы гарантировать, что все ссылки в формулах будут содержать [*], где звездочка представляет собой символ шаблона.

ПРИМЕЧАНИЕ

В Excel 97 не предусмотрена возможность поиска во всей рабочей книге, но вы можете выполнить поиск во всех листах в книге, сгруппировав их. Для этого правой кнопкой мыши щелкните ярлычок любого листа и в контекстном меню выберите команду Выделить все листы (Select All Sheets). В более поздних версиях Excel команды Найти (Find) и Заменить (Replace) обеспечивают возможность поиска в пределах листа или книги.

Выполнив поиск ссылок в формулах, измените или вообще удалите соответствующие формулы. Решение изменять формулу или удалять зависит от конкретной ситуации, и только вам решать, по какому пути пойти.

Кроме того, можно отправиться в Центр загрузки Microsoft Office (Microsoft Office Download Center), расположенный по адресу: <http://office.microsoft.com/Downloads/default.aspx>, и в категории Надстройки (Add-Ins) выбрать Delete Links Wizard. Эта утилита предназначена для поиска и удаления ссылок, таких, как ссылки на определенные имена, ссылки на скрытые имена, ссылки на диаграммы, ссылки Microsoft Query и ссылки на объекты. Однако, как показывает наш опыт, она не находит несуществующие ссылки.

Когда вы удостоверитесь, что в формулах ссылок нет, нужно будет убедиться, что допустимые ссылки не прячутся где-либо еще. Для этого нужно начать из глубины книги Excel, содержащей несуществующие ссылки. Выберите команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define). Пролистайте список имен, щелкая и выделяя их по очереди и просматривая формулу в поле Формула (Refers to) внизу. Удостоверьтесь, что ни одно из имен не ссылается на другую рабочую книгу.

ПРИМЕЧАНИЕ

Вместо того чтобы щелкать подряд все имена в диалоговом окне Присвоение имени (Define Name), можно вставить в книгу новый лист и выбрать пункт меню Вставка ► Имя ► Вставить (Insert ► Name ► Paste). Затем в диалоговом окне Вставка имени (Paste Name) щелкните кнопку Вставить все (Paste Link). Будет создан список всех имен в вашей книге, а диапазоны, на которые они ссылаются, будут перечислены в соответствующем столбце.

Если любое из имен указывает за пределы вашей книги, вы не найдете источника, по меньшей мере, одной ссылки, что и решит вопрос обновления. Теперь вы должны решить, что необходимо делать: изменить имя диапазона, чтобы оно ссылалось только на эту книгу, или оставить все как есть.

Другой возможный источник ссылок — это ваши диаграммы. У них может проявиться та же проблема, которую мы только что рассматривали. Необходимо проверить, что все диапазоны данных и метки оси X (X-axis) диаграммы не ссылаются на внешние книги. И снова вам решать, являются ли найденные ссылки правильными.

Ссылки скрываются и в объектах, например, текстовых полях, автофигурах и т. д. Объекты могут ссылаться на внешние рабочие книги. Самый простой способ найти объекты — выбрать любую ячейку на листе и затем выбрать команду Правка ► Перейти (Edit ► Go To). В диалоговом окне Переход (Go To) щелкните на кнопке Выделить (Special), установите флажок Объекты (Objects) и щелкните на кнопке ОК. Будут выбраны все объекты на листе. Это необходимо выполнять на копии вашей рабочей книги. Когда все объекты будут выделены, вы сможете удалить, сохранить, закрыть и снова открыть вашу копию, чтобы проверить, решена ли проблема.

И, наконец, последнее неочевидное место, где можно найти реальные ссылки — это скрытые листы, которые вы могли создать и забыть о них. Отобразите такие листы, выбрав команду Формат ► Лист ► Отобразить (Format ► Sheet ► Unhide). Если пункт Отобразить (Unhide) во вложенном меню Лист (Sheet) неактивен, это означает, что у вас нет скрытых листов. (Если вы уверены, что в книге есть листы, не появляющиеся в этом меню, обратитесь за подробностями к разделу «Трюк № 5. Как скрыть лист, чтобы его невозможно было отобразить».)

Теперь, когда вы устранили возможность появления реальных ссылок, настало время удалить несуществующие. Откройте рабочую книгу с несуществующими ссылками и выберите команду Правка ▶ Связи (Edit ▶ Links). Иногда можно просто выделить ненужную ссылку, щелкнуть кнопку Изменить (Change Source) и связать ссылку с самой собой. Но чаще всего появляется сообщение, что одна из формул содержит ошибку, и это сделать невозможно.

Если применить простой способ невозможно, обратите внимание, к чему, по мнению Excel, привязана ссылка (мы будем называть эту книгу хорошей книгой). Создайте настоящую связь между двумя книгами, открыв обе. Перейдите к проблемной книге и в любой ячейке любого листа введите =. Теперь щелкните любую ячейку в хорошей книге и нажмите клавишу Enter, чтобы создать настоящую внешнюю ссылку на другую книгу.

Сохраните обе книги, но не закрывайте их. В книге с несуществующими ссылками выберите команду Правка ▶ Связи (Edit ▶ Links) и щелкните кнопку Изменить (Change Source), чтобы увидеть все ссылки на новую рабочую книгу, с которой вы только что намеренно создали новую связь. Снова сохраните вашу рабочую книгу и удалите ячейку, в которой создали настоящую внешнюю ссылку. И еще раз сохраните файл.

При этом часто исчезает ненужная несуществующая ссылка, так как Excel, наконец, понимает, что вы удалили внешнюю ссылку на книгу. Если проблема, однако, не решена, попробуйте выполнить следующее, но сначала сохраните копию книги.

ВНИМАНИЕ

Следующий процесс включает удаление данных без возможности их восстановления. Таким образом, прежде чем начать, создайте резервную копию вашей книги. Если вы этого не сделаете, возможно появление проблем.

В открытой проблемной книге удалите один лист, сохраните книгу, затем закройте и заново откройте ее. Если вы не увидите приглашение обновить отсутствующие связи, значит, удаленный лист содержал несуществующую ссылку. Теперь проблема должна быть решена. Однако если это не так, повторите первый шаг для каждого листа в книге. Вам понадобится добавить новый лист перед удалением последнего, так как в рабочей книге должен быть хотя бы один лист.

Мы продолжаем, предполагая, что эта техника у вас сработала. Вот что следует сделать далее. Откройте копию вашей рабочей книги (ту, которая содержит все данные) и сделайте еще одну копию. Вы будете работать с проблемным листом (или листами) и выполнять процесс удаления, чтобы обнаружить, где на листе прячется проблема.

Открыв проблемный лист, выделите некоторый диапазон ячеек (область приблизительно 10 x 10 ячеек) и затем выберите команду Правка ▶ Очистить > Все (Edit ▶ Clear ▶ All). (Вы уверены, что сохранили копию книги?) Сохраните, закройте и заново откройте лист. Если приглашение обновить ссылки не появилось, вы нашли проблему, и теперь необходимо переделать этот блок ячеек. Если приглашение открылось, продолжайте очищать ячейки, пока при очередном открытии приглашение не будет выведено. После этого измените проблемные ячейки.

Мы надеемся, что эти приемы сэкономят ваше рабочее время и нервы, которые тратятся на эти несчастные несуществующие ссылки в рабочих книгах. Это не самые простые и приятные решения, но они позволят забыть о проблемах.

**ТРЮК
№ 14**

Предотвращение распухания рабочей книги

Вы когда-нибудь замечали, что рабочая книга увеличивается в размере с пугающей скоростью без какой-либо очевидной причины? Это происходит по нескольким причинам, и мы предлагаем решения, помогающие вашим книгам похудеть.

Бывало ли с вами такое, что из-за переедания случались проблемы пищеварения? Распухание книги в Excel очень похоже на такую ситуацию. Распухание рабочей книги — это термин, обозначающий ситуацию, когда в книге сделано так много и она увеличилась до такого размера, что не может правильно функционировать.

Мы проверили размер типичной книги, содержащей достаточно много данных. Только с данными размер файла книги был равен 1,37 Мбайт. Затем мы добавили сводную таблицу, которая ссылалась на четыре полных столбца исходных данных, и заметили, что размер файла существенно увеличился — до 2,4 Мбайт. Добавьте еще немного форматирования, и размер типичной книги возрастет практически вдвое после выполнения всего лишь нескольких действий.

Одна из наиболее часто встречающихся причин распухания файла, особенно в ранних версиях Excel, — это наложение форматирования на столбцы или строки целиком, а не только на используемый диапазон данных. Другая причина — это ссылки на полные столбцы как на источники данных для диаграмм и сводных таблиц, а не только на ячейки с фактическими данными. Чтобы исправить эти ошибки, понадобится исключить излишнее форматирование и ограничить источники данных только нужными диапазонами ячеек.

ВНИМАНИЕ

Перед тем как выполнять такие изменения, всегда создавайте резервную копию рабочей книги.

Устранение излишнего форматирования

Первый шаг при устранении излишнего форматирования — это вычисление ячейки, где заканчиваются данные на рабочем листе, то есть правого нижнего угла таблицы данных. Не полагайтесь на команду Правка ▶ Перейти ▶ Выделить ▶ Последнюю ячейку (Edit ▶ Go To ▶ Special ▶ Last Cell), так как она приведет вас к последней ячейке, содержащей форматирование, а не фактические данные. Вручную найдите последнюю ячейку с фактическими данными, выделите строку сразу же за ней. Удерживая клавиши Ctrl и Shift, нажимайте на клавиатуре стрелку вниз, чтобы выделить все строки ниже, и выберите команду Правка ▶ Очистить ▶ Все (Edit ▶ Clear ▶ All), чтобы очистить их.

Теперь примените ту же логику к нежелательному форматированию в столбцах. Найдите ячейку в последнем столбце, содержащем данные, и щелкните кнопку столбца справа. Удерживайте сочетание клавиш **Ctrl+Shift** и нажимайте на клавиатуре стрелку вправо, чтобы выделить все остальные столбцы справа, а затем очистите их командой **Правка ▶ Очистить ▶ Все (Edit ▶ Clear ▶ All)**.

ВНИМАНИЕ

Не нужно фактически удалять эти строки или столбцы вместо их очистки, так как при этом в формулах любых ячеек, которые могут ссылаться на них, часто возникает ошибка **#REF!**.

Сохраните книгу и радостно взгляните на изменение в размере файла, выбрав команду **Файл ▶ Свойства ▶ Общие (File ▶ Properties ▶ General)**.

Если у вас есть макросы, то теперь необходимо обратиться к модулям, в которые помещен код макросов. Это достаточно быстрый, безболезненный и очевидный процесс, который включает экспорт всех модулей и пользовательских форм **UserForm** на жесткий диск, удаление существующих модулей и форм, сохранение и импортирование только что экспортированных модулей.

Чтобы сделать это, перейдите в редактор **Visual Basic** и в окне **Project Explorer** правой кнопкой мыши щелкните каждый модуль и выберите команду **Remove Module1** (считайте, что вместо **Module1** стоит имя любого вашего модуля). Когда появится сообщение с вопросом, нужно ли экспортировать модуль перед удалением, ответьте **Yes** и запомните путь, куда будут экспортированы модули.

Выполните это для всех модулей по очереди, а также для любых форм **UserForm**, которые у вас могут быть. Не забывайте частные модули вашей книги и листов, если в них также содержится код. Закончив, сохраните рабочую книгу. Затем выберите команду **Файл ▶ Импорт (File ▶ Import File)** и импортируйте все модули и формы **UserForm** обратно в рабочую книгу. При этом будет создан текстовый файл для каждого модуля и удалено все лишнее, что может содержаться в модулях.

В сети можно найти несколько бесплатных утилит, которые до некоторой степени автоматизируют эту задачу, но мы встречались со случаями, когда такие утилиты только усложняли код, увеличивая размер файлов. Если вы все же применяете подобные утилиты, всегда сначала сохраняйте резервную копию книги, так как разработчики утилит не несут ответственности за потерю данных.

Настройка источников данных

Если после выполнения предыдущих действий вы все же уверены, что размер файла слишком большой, можно проверить ссылки на неиспользуемые ячейки в сводных таблицах (**PivotTable**) и сводных диаграммах (**PivotChart**). Это особенно касается сводных таблиц, так как пользователи часто создают ссылки на все 65 536 строк, чтобы избежать обновления диапазонов вручную после добавления данных. Если это ваш случай, используйте динамические именованные диапазоны [Трюк № 42] для источников данных.

Очистка поврежденных книг

Если вы все так же считаете, что ваша книга слишком большая, **возможно**, эта книга или составляющие ее листы повреждены. К сожалению, определение точки повреждения требует выполнить процесс исключения вручную.

ВНИМАНИЕ

И снова мы настоятельно рекомендуем сохранить копию рабочей книги перед переходом к следующей процедуре.

Чтобы гарантировать, что вы ничего не пропустили, отобразите все скрытые листы командой **Формат** ▶ **Лист** ▶ **Отобразить** (Format ▶ Sheet ▶ Unhide). Если эта команда недоступна, вам незачем беспокоиться о скрытых листах — у вас их нет. Теперь, когда все листы видимы, начните с самого левого и по очереди обрабатывайте все листы. Для каждого листа: удалите лист, сохраните книгу и проверьте размер файла, выберите команду **Файл** ▶ **Свойства** ▶ **Общие** (File ▶ Properties ▶ General). Если размер файла существенно уменьшился, учитывая количество данных на удаленном листе, то вы, вероятно, нашли место повреждения.

Чтобы заменить в книге поврежденный лист, создайте новый лист, вручную выделите данные на поврежденном листе, вырежьте (не копируйте!) и вставьте их на новый лист. Удалите поврежденный лист из книги, сохраните ее и повторите процесс.

ПРИМЕЧАНИЕ

Благодаря вырезанию, а не копированию данных Excel автоматически будет переносить на новый лист данные с соответствующими ссылками.

Т Р Ю К № 15 Получение данных из поврежденной КНИГИ

Повреждение книги может означать потерю жизненно важных данных и стоить вам не только денег. Этот трюк изучает несколько способов восстановления данных.

Иногда рабочие книги повреждаются безо всяких видимых причин. При этом могут возникать проблемы любого сорта, особенно если это очень важная книга и по какой-то причине у вас нет ее резервной копии. Урок 1: всегда создавайте резервную копию ваших данных. В действительности, конечно же, это делается не постоянно, поэтому повреждение может произойти прямо перед очередным архивированием данных.

Чтобы еще больше запутать вас, скажем, что, даже если вы знаете, что ваша книга повреждена, иногда все же можно ее открыть и даже выполнить в ней определенные действия.

Если книгу можно открыть

Если проблемную книгу можно открыть, то перед тем как делать что-либо еще, обязательно сохраните ее копию, иначе можно об этом пожалеть. Если у вас есть копия, вы всегда сможете обратиться за профессиональной помощью!

Теперь попробуйте открыть книгу в более поздней версии Excel и просто сохранить ее. Очевидно, это невозможно, если вы уже работаете в самой последней версии программы.

Если это не работает, попробуйте открыть и сохранить книгу как файл в формате HTML или HTM, затем закройте файл и заново откройте его; после этого сохраните файл в нужном формате, то есть .xls.

При сохранении в формате HTML или HTM будут потеряны следующие возможности¹:

- пользовательские представления;
- неиспользуемые числовые форматы;
- неиспользуемые стили;
- настройки объединения данных;
- сценарии;
- формулы естественных языков (они преобразуются в стандартные ссылки на диапазоны);
- пользовательские категории функций;
- элементы зачеркивания, нижние и верхние индексы;
- изменения в истории;
- пользовательские настройки страниц для диаграмм, встроенные в лист;
- настройки списков (ListBox) и полей со списками (ComboBox) на панели инструментов Формы (Forms);
- условное форматирование, хранящееся в листе макроса XLM.

Кроме того, совместно используемые книги уже не будут совместно использоваться. Значение параметра Пересечение с осью Y (значений) в категории номер (Value (Y) axis crosses at category number) на вкладке Шкала (Scale) диалогового окна Формат оси (Format Axis) не сохраняется, если установлен флажок Пересечение с осью Y (значений) в максимальной категории (Value (Y) axis crosses a maximum category). Значение параметра Разноцветные точки (Vary colors by point) диалогового окна Формат ряда данных (Format Data Series) не сохраняется, если диаграмма содержит более одного ряда данных.

Наконец, попробуйте открыть файл и сохранить его в формате SYLK (.slk, символическая ссылка). Помните, что когда книга записывается в этом формате, сохраняется только активный лист. Следовательно, эту процедуру нужно будет выполнить для каждого листа. Заново откройте файл и сохраните ее в желаемом формате, например, .xls.

¹ Результат сильно зависит от версии Excel. — Примеч. ред.

Если файл открыть невозможно

Если ваша книга повреждена до такой степени, что вы даже не можете открыть ее, откройте электронную таблицу в Microsoft Word или в программе просмотра электронных таблиц Spreadsheet viewer, которую можно загрузить с сайта Microsoft, а затем скопируйте данные из открытого файла. Однако большая часть форматирования, формул и т. п. будет утеряна.

Теперь откройте новую книгу и создайте внешнюю связь с поврежденной книгой, например, ='C:\Documents and Settings\Raina\My Documents\[ChookSheet.xls]Sheet1!A1. Скопируйте эту ссылку в нужное количество ячеек (вниз по строкам и вправо по столбцам). Сделайте то же самое для всех листов книги. Если вы не можете запомнить имена листов, восстановите старые имена листов. Для этого укажите правильный путь к файлу, и после нажатия клавиши Enter Excel отобразит старые названия листов.

Кроме того, можно посетить сайт OpenOffice.org и загрузить бесплатную версию программы OpenOffice.org. Кроме отличающихся названий различных утилит и команд, OpenOffice.org очень похожа на Excel. OpenOffice.org основана на той же базовой структуре электронных таблиц, что и Excel, поэтому в ней удобно работать пользователям Excel. В действительности, около 96 % формул, которые используются в Excel, можно создать и применить в электронных таблицах OpenOffice.org.

Чтобы загрузить бесплатную версию OpenOffice.org, перейдите по адресу: <http://download.openoffice.org/index.html> и загрузите ее, выбрав FTP-сайт. Затем установите программу. Можно загрузить OpenOffice.org и для Macintosh.

Чаще всего восстановить данные Excel удастся. Однако код VBA из-за несовместимости OpenOffice.org и Excel восстановить невозможно.

К сожалению, если ни один из перечисленных способов не работает, вам, вероятно, придется заплатить, чтобы восстановить книгу особой программой. Один из источников, где можно приобрести подобное надежное программное обеспечение (для Windows) — это сайт авторов этой книги, расположенный по адресу: <http://www.ozgrid.com/Services/corrupt-file-recovery-index.htm>.

После приобретения и установки запустите программу ExcelFix. Щелкните на кнопке Select File, выберите поврежденный файл, а затем щелкните на кнопке Diagnose, чтобы восстановить его. Теперь вы сможете увидеть восстановленный файл в программе просмотра рабочих книг. Щелкните на кнопке Save Workbook, чтобы сохранить книгу в читаемом файле, который можно будет открыть в Excel.

Можно загрузить и демонстрационную версию программы, которая не позволяет сохранять файл. Однако все версии программы позволяют начать заново и восстановить столько файлов, сколько вам нужно.

Трюки со встроенными возможностями Excel

Трюки с 16 по 38

Хотя Excel поддерживает множество стандартных возможностей управления и анализа данных, ограничения этих возможностей зачастую вводят в заблуждение. Трюки этой главы предложат несколько способов, как обойти эти ограничения и превратить Excel в намного более мощный инструмент.

Т Р Ю К
№ 16

Проверка данных на основе списка на другом листе

При проверке данных легко можно указать правила, которым должны отвечать данные. К сожалению, Excel настаивает, что списки, которые используются при проверке, должны находиться на том же рабочем листе, что и проверяемые данные. К счастью, всегда есть способы уклониться от выполнения этого требования.

В этом трюке мы познакомим вас с двумя способами проверки данных на основе списка на другом листе. Первый из них использует преимущество именованных диапазонов Excel (которые подробнее рассматриваются в главе 3), а во втором применяется вызов функции.

Способ 1. Именованные диапазоны

Возможно, самый простой и быстрый способ преодолеть барьеры, которые воздвигает Excel при проверке данных, — присвоить имя диапазону, где содержится список. Для создания именованного диапазона выделите ячейки, содержащие список, и введите имя в поле имени строки формул. Для выполнения этого примера мы будем предполагать, что диапазону присвоено имя **MyRange**.

Выделите **ячейку**, в которой должен будет появиться раскрывающийся список, а затем выберите команду **Данные ▶ Проверка (Data ▶ Validation)**. В поле **Тип данных (Allow)** выберите пункт **Список (List)**, а в поле **Источник (Source)** введите **=MyRange**. Щелкните на кнопке **ОК**.

Поскольку вы использовали именованный диапазон, ваш список (хотя он и находится на другом листе) теперь **можно** использовать как список проверки.

Способ 2. Функция ДВССЫЛ

Функция ДВССЫЛ (INDIRECT) позволяет ссылаться на ячейку, содержащую текст, представляющий адрес ячейки. Эту ячейку можно использовать как локальную ссылку, даже если она получает данные из другого листа. Можно применять эту возможность для связи с листом, где расположен список.

Предположим, список находится на листе Sheet1 в диапазоне \$A\$1:\$A\$8. Щелкните любую ячейку на другом листе, где должен появиться этот список проверки (список выборки). Затем выберите команду Данные ▸ Проверка (Data ▸ Validation) и в поле Тип данных (Allow) выберите пункт Список (List). В поле Источник (Source) введите следующий код:

```
=INDIRECT("Sheet1!$A$1:$A$8")
```

В русской версии Excel

```
=ДВССЫЛ("Sheet1!$A$1:$A$8")
```

Удостоверьтесь, что флажок Список допустимых значений (In-Cell) установлен, и щелкните на кнопке ОК. Список на листе Sheet1 должен появиться в раскрывающемся списке проверки.

Если имя листа, на котором расположен список, содержит пробелы, необходимо использовать следующий синтаксис функции ДВССЫЛ (INDIRECT):

```
=INDIRECT("'Sheet1'!$A$1:$A$8")
```

В русской версии Excel

```
=ДВССЫЛ("'Sheet1'!$A$1:$A$8")
```

Различие заключается в том, что здесь после первой кавычки стоит один апостроф, а второй апостроф находится перед восклицательным знаком.

ПРИМЕЧАНИЕ

Полезно всегда использовать одиночные апострофы, независимо от того, содержит имя пробелы или нет. С апострофами вы все так же сможете ссылаться на листы с именами без пробелов, и это также упростит внесение изменений позже.

Преимущества и недостатки обоих способов

У именованных диапазонов и функции ДВССЫЛ (INDIRECT) есть преимущества и недостатки.

Преимущество использования именованного диапазона заключается в том, что изменение названия листа не повлияет на список проверки. Это подчеркивает недостаток функции ДВССЫЛ (INDIRECT): любое изменение названия листа не будет автоматически в ней отражаться, поэтому придется вручную изменить функцию, указав новое название листа.

Преимущество функции ДВССЫЛ (INDIRECT): когда из именованного диапазона будет удалена первая ячейка или строка либо последняя ячейка или строка, то именованный диапазон вернет ошибку #REF!. В этом недостаток именованного диапазона — если удалить из него ячейки или строки, изменения не повлияют на список проверки.

Т Р Ю К
№ 17

Управление условным форматированием при помощи флажков

Хотя условное форматирование — одна из наиболее мощных возможностей Excel, достаточно неудобно включать и выключать его, пробираясь по меню и диалоговым окнам. Добавив на лист флажки, включающие и выключающие форматирование, вы упростите считывание данных удобным для вас способом.

Условное форматирование, которое впервые появилось в Excel 97, применяет форматирование к выбранным ячейками, отвечающим условиям, которые основаны на указанных вами значениях или формулах. Хотя условное форматирование обычно применяется на основе значений в ячейках, форматирование на основе формул обеспечивает достаточную гибкость для расширения окна условного форматирования на всю сетку таблицы.

Настройка флажков для условного форматирования

Флажки на панели инструментов Формы (Forms) возвращают в связанную ячейку значение ИСТИНА (TRUE) или ЛОЖЬ (FALSE) (установлен — не установлен). Комбинируя флажок с панели инструментов Forms (Формы) с условным форматированием при помощи параметра Formula Is (Формула) (рис. 2.1), вы сможете включать и выключать условное форматирование флажком.

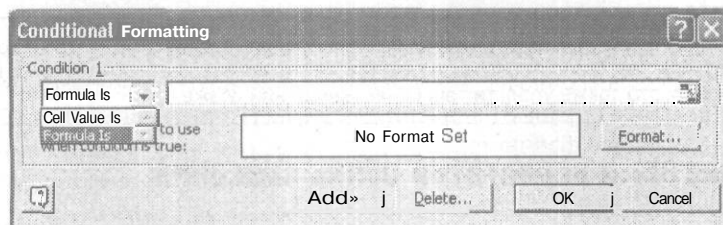


Рис. 2.1. Диалоговое окно Условное форматирование с условием Формула

ПРИМЕЧАНИЕ

При совместном использовании с формулой (например, с параметром Формула (Formula Is)) условное форматирование автоматически форматирует ячейку, когда формула возвращает значение ИСТИНА (TRUE). Поэтому формулы, которые вы будете применять для этого трюка, должны возвращать ИСТИНА (TRUE) или ЛОЖЬ (FALSE).

Чтобы понять, о чем идет речь, попробуйте выполнить следующий простой пример, который скрывает данные при помощи условного форматирования и флажка. Для этого примера мы используем диапазон $\$A\$1:\$A\10 , последовательно заполненный числами от 1 до 10. Чтобы получить флажок с панели инструментов Формы (Forms), выберите команду Вид \blacktriangleright Панели инструментов \blacktriangleright Формы (View \blacktriangleright Toolbars \blacktriangleright Forms), на появившейся панели инструментов Формы (Forms) щелкните элемент управления Флажок (Checkbox), затем щелкните около ячейки C1 на листе, чтобы поместить туда флажок. Правой кнопкой мыши щелкните этот флажок, в контекстном меню выберите команду Формат объекта (Format) и в диалоговом окне Формат элемента управления (Format Control) перейдите на вкладку Элемент управления (Control). В поле Связь с ячейкой (Cell Link) введите C1 и щелкните кнопку ОК (рис. 2.2).

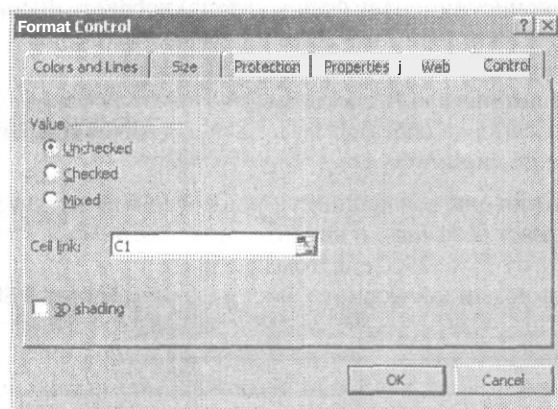


Рис. 2.2. Диалоговое окно

Щелчок на флажке, размещенном рядом с ячейкой C1, будет возвращать в ячейку C1 значение ИСТИНА (TRUE) или ЛОЖЬ (FALSE). Так как видеть эти значения не нужно, выделите ячейку C1 и измените цвет шрифта на белый.

Теперь выделите ячейки $\$A\$1:\$A\10 , начиная с A1. Выберите команду Формат \blacktriangleright Условное форматирование (Format \blacktriangleright Conditional Formatting) и в группе условий выберите пункт Формула (Formula Is) — сначала вы увидите значение Значение (Cell Value Is). В поле ввода справа от значения Формула (Formula) введите $=\$C\1 . В диалоговом окне Условное форматирование (Conditional Formatting) щелкните кнопку Формат (Format), перейдите на вкладку Шрифт (Font) и измените цвет шрифта на белый. Щелкните на кнопке ОК, затем еще раз щелкните на кнопке ОК.

Установите флажок, и цвет шрифта для данных в диапазоне $\$A\$1:\$A\10 автоматически изменится на белый. Сбросьте флажок, будет восстановлен обычный цвет.

Включение и выключение выделения числа

Возможность автоматически выделять числа, отвечающие определенному критерию, намного упрощает поиск необходимых данных в электронной таблице. Чтобы

сделать это, выделите ячейку E1 (или любую другую, которая вам больше нравится) и присвойте ей имя CheckBoxLink в поле имени слева в строке формул (рис. 2.3).

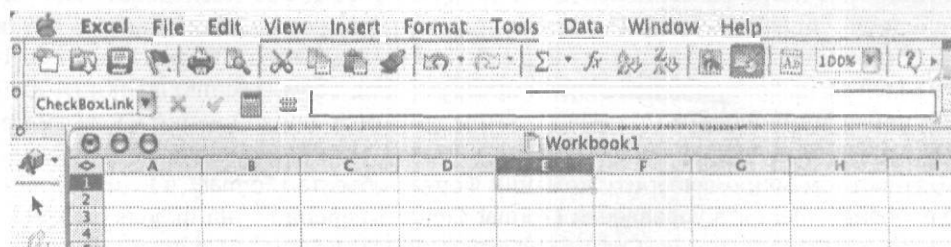


Рис. 2.3. Ячейка E1 с именем CheckBoxLink

При помощи панели инструментов **Формы** (Forms) добавьте флажок на чистый рабочий лист, назовите этот лист Checkboxes и переместите флажок в ячейку A1. Для этого флажка выберите связанную ячейку CheckBoxLink, правой кнопкой мыши щелкнув флажок и выбрав в контекстном меню команду **Формат объекта** ▶ **Элемент управления** (Format Control ▶ Control). В поле **Связь с ячейкой** (Cell Link) введите CheckBoxLink и щелкните на кнопке **ОК**.

Правой кнопкой мыши еще раз щелкните флажок, в контекстном меню выберите команду **Изменить текст** (Edit Text) и введите слова Show Me. В столбце A на другом листе введите числа от 25 до 2500 с шагом 25. Дайте этому диапазону имя Numbers и скройте этот лист командой **Формат** ▶ **Лист** ▶ **Скрыть** (Format > Sheet ▶ Hide).

ПРИМЕЧАНИЕ

Чтобы быстро ввести числа, в ячейке A1 введите 25. Затем правой кнопкой мыши щелкните маркер заполнения (fill handle) (который выглядит как небольшой черный квадрат в правом нижнем углу выделенной ячейки) и, удерживая правую кнопку мыши, перетащите маркер вниз до строки 100. Теперь отпустите правую кнопку мыши, в контекстном меню выберите команду **Прогрессия** (Series), введите шаг 25 и введите предельное значение 2500. Щелкните на кнопке **ОК**.

На листе Checkboxes выделите ячейку B1 и присвойте ей имя FirstNum. Выделите ячейку D1 и присвойте ей имя SecondNum. В ячейке C1 введите слово И (AND). Теперь выделите ячейку B1 (FirstNum) и, удерживая клавишу Ctrl, выделите ячейку D1 (SecondNum). Выберите команду **Данные** ▶ **Проверка** ▶ **Параметры** (Data ▶ Validation ▶ Settings). В поле **Тип данных** (Allow) выберите пункт **Список** (List), а в поле **Источник** (Source) введите =Numbers. Убедитесь, что установлен флажок **Список доступных значений** (In-Cell), и щелкните на кнопке **ОК**. Теперь в обеих ячейках появятся раскрывающиеся списки чисел от 25 до 2500.

В ячейке A1 введите заголовок Amount. Сразу под ней заполните диапазон A2:A20 любыми числами в диапазоне от 25 до 2500. Выделите ячейки A2:A20 (начиная с ячейки A2, чтобы она оставалась активной ячейкой при выделении) и выберите команду **Формат** ▶ **Условное форматирование** (Format ▶ Conditional Formatting).

В открывшемся диалоговом окне (рис. 2.4), выберите пункт **Формула** (Formula Is) (сначала там появляется параметр **Значение** (Cell Value Is)). Затем в поле **Формула** (Formula) введите следующую формулу:

```
=AND($A2>=FirstNum,$A2<=SecondNum,CheckBoxLink)
```

В русской версии Excel

```
=И($A2>=FirstNum,$A2<=SecondNum,CheckBoxLink)
```

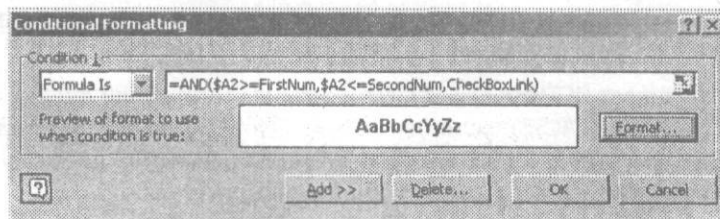


Рис. 2.4. Диалоговое окно Условное форматирование

Щелкните кнопку **Формат (Format)** и выберите любое нужное форматирование. Щелкните на кнопке **ОК**, затем еще раз щелкните на кнопке **ОК**, чтобы закрыть диалоговые окна. Измените цвет шрифта для ячейки **CheckBoxLink (E1)** на белый, чтобы значения **ИСТИНА (TRUE)** и **ЛОЖЬ (FALSE)** не отображались. В ячейке **FirstNum (B1)** выберите любое **число**, а затем в ячейке **SecondNum (D1)** выберите любое другое число, **большее** первого.

Установите флажок, и только что установленное условное форматирование будет автоматически применено к числам в указанном диапазоне. Сбросьте флажок, будет восстановлено форматирование по умолчанию.

Как можно видеть, используя флажок в сочетании с условным форматированием, вы сможете сделать то, что обычно считается возможным только с применением кода **VBA**.



Поиск формул при помощи условного форматирования

В Excel не предусмотрено встроенной функции для поиска формул. Когда формула введена в ячейку, узнать, является ячейка постоянным значением или значением, полученным из формулы, можно, только щелкнув ячейку и взглянув на строку формул. Этот трюк заполнит пробел при помощи пользовательской функции.

Код **VBA** в этой пользовательской функции (также называемой *функцией, определенной пользователем*) позволяет идентифицировать ячейки, содержащие формулы, не щелкая и не изучая все 10 000 ячеек.

Чтобы превратиться в хитрого охотника на формулы, начните с перехода в редактор **Visual Basic — Сервис ▶ Макрос ▶ Редактор Visual Basic (Tools ▶ Macro ▶ Visual Basic Editor)** — и выберите команду **Insert ▶ Module**. Введите следующую функцию:

```
Function IsFormula(Check_Cell As Range)
    IsFormula = Check_Cell.HasFormula
End Function
```

Закройте окно (нажмите сочетание клавиш **Alt/Apple+Q** или кнопку закрытия в строке заголовка окна). Теперь эта функция доступна из любой ячейки на любом листе в этой книге; для этого нужно ввести формулу `=IsFormula(A1)`. Добавить эту функцию можно также, выбрав команду **Вставка** ▶ **Функция** (**Insert** ▶ **Function**), для параметра **Категория** (**Category**) выберите вариант **Определенные пользователем** (**User-Defined**) и в списке функций выберите **IsFormula**.

Формула возвращает **ИСТИНА** (**TRUE**), если в соответствующей ячейке содержится формула, и **ЛОЖЬ** (**FALSE**), если это не так. Этот булевский результат можно использовать совместно с условным форматированием, чтобы автоматически выделять все формулы с применением нужного форматирования.

Одно из основных преимуществ этого метода заключается в том, что возможности идентификации формул на листе динамические. Это означает, что, если вы добавите или удалите формулу, то форматирование изменится соответствующим образом. Вот как это сделать.

Выделите диапазон ячеек на листе, скажем, **A1:J500**, и захватите еще несколько ячеек на случай, если формулы будут добавлены позднее.

ВНИМАНИЕ

Не выделяйте весь лист, так как этот метод может увеличить размер и излишне нагрузить электронную таблицу.

Теперь, когда ячейки выделены, причем ячейка **A1** должна быть активной, выберите команду **Формат** ▶ **Условное форматирование** (**Format** ▶ **Conditional Formatting**). В поле условий выберите **Формула** (**Formula Is**) и в поле справа от него введите `=IsFormula(A1)`. Щелкните кнопку **Формат** (**Format**) и выберите нужное форматирование, которое будет применяться для идентификации ячеек с формулами. Щелкните на кнопке **ОК**, затем еще раз щелкните на кнопке **ОК**.

ПРИМЕЧАНИЕ

Иногда при вводе формул в окне условного форматирования Excel пытается добавить кавычки вокруг формул после того, как вы щелкаете на кнопке **ОК**. Это означает, что Excel распознал то, что вы ввели, как **текст**, а не как формулу. Если это произошло, вернитесь в окно **Условное форматирование** (**Conditional Formatting**), удалите кавычки и щелкните на кнопке **ОК**.

Теперь указанная формула должна быть применена ко всем ячейкам на листе, содержащим формулы. Если вы удалите или исправите содержимое ячейки с формулой, условное форматирование исчезнет. Схожим образом, если вы введете новую формулу в любую ячейку диапазона, она также будет выделена форматированием.

Этот простой трюк с условным форматированием может сделать ваши электронные таблицы намного проще для работы, когда время дойдет до их поддержки или модификации.

Т Р Ю К
№ 19

Подсчет и суммирование ячеек, отвечающих критерию условного форматирования

Увидев результат условного форматирования, вы, вероятно, захотите создать формулы, ссылающиеся только на данные, которые подверглись условному форматированию. В Excel не очень хорошо реализована такая возможность, но мы поможем ему.

Пользователи Excel регулярно задают вопрос, как можно провести вычисления только с ячейками определенного фонового цвета. Этот вопрос возникает так часто, потому что в Excel не предусмотрена стандартная функция для выполнения этой задачи; однако это можно сделать при помощи пользовательской функции, как демонстрирует Трюк № 88 «Подсчет или суммирование ячеек с определенным цветом заливки».

Единственная неприятность при использовании пользовательской функции – она не понимает форматирование, наложенное при помощи условного форматирования. Однако, немного подумав, вы сможете добиться нужного результата, не применяя пользовательскую функцию.

Пусть есть длинный список чисел в диапазоне \$A\$2:\$A\$100. Вы наложили на эти ячейки условное форматирование, чтобы помечены были только числа между 10 и 20. Теперь необходимо сложить значения в ячейках, отвечающих только что установленному критерию, а затем выделить сумму значений при помощи условного форматирования. Неважно, какое именно форматирование применяется к этим ячейкам, однако необходимо знать критерий, согласно которому ячейки выделяются (в нашем случае это условие, что значение находится между 10 и 20).

Чтобы сложить диапазон ячеек, отвечающих определенному критерию, можно использовать функцию СУММЕСЛИ (SUMIF), но при этом можно указать только один критерий. Если у вас несколько факторов проверки, можно использовать формулу массива.

Формулу массива можно записать так:

```
=SUM(IF($A$2:$A$100>10,IF($A$2:$A$100<20,$A$2:$A$100)))
```

В русской версии Excel

```
=СУММ(ЕСЛИ($A$2:$A$100>10,ЕСЛИ($A$2:$A$100<20,$A$2:$A$100)))
```

ВНИМАНИЕ

При вводе формул массива нажимайте не клавишу Enter, а сочетание клавиш Ctrl+Shift+Enter. После этого Excel поместит фигурные скобки вокруг формулы, чтобы она выглядела так:

```
{=SUM(IF($A$2:$A$100>10,IF($A$2:$A$100<20,$A$2:$A$100)))}
```

Если вы поставите скобки самостоятельно, формула не будет работать. Это должен сделать Excel.

Помните также, что использование формул массива может замедлить пересчет в Excel, если в этих формулах присутствует слишком много ссылок на большие диапазоны.

Чтобы больше узнать о формулах массива, посетите страницу <http://www.ozgrid.com/Excel/arrays.htm>.

Альтернативный путь

В качестве альтернативы можно использовать дополнительный столбец (например, столбец В) для ссылки на ячейки в столбце А. Эти ссылки будут возвращать результат в столбец В, только если значение отвечает установленному условию, то есть >10 , <20 . Для этого сделайте следующее:

Выделите ячейку В1 и введите следующую формулу:

```
=IF(AND(A2>10,A2<20),A2,"")
```

В русской версии Excel

```
=ЕСЛИ(И(A2>10,A2<20),A2,"")
```

Запишите эту формулу в каждую ячейку до ячейки В100. После этого, если в столбце А есть значения, в столбце В должны оказаться значения между 10 и 20.

ПРИМЕЧАНИЕ

Чтобы быстро скопировать формулу в соседние ячейки вниз до последней строки, введите формулу в первую ячейку (В2), заново выделите эту ячейку и дважды щелкните маркер заполнения. Это можно сделать и выбрав команду Правка ▶ Заполнить ▶ Вниз (Edit ▶ Fill ▶ Down).

Теперь выберите любую ячейку, где должна появиться сумма, и воспользуйтесь для сложения стандартной функцией СУММ (SUM). (Можно скрыть столбец В, если вы не хотите видеть дополнительный столбец с возвращенными формулой значениями.)

Оба предыдущих способа хорошо справляются с задачей, однако в Excel предусмотрена еще одна функция, которая позволяет указать несколько условий. Эта функция входит в набор функций баз данных Excel и называется БДСУММ (DSUM). Чтобы проверить ее, используйте тот же набор чисел в диапазоне А2:А100. Выделите ячейки С1:Д2 и присвойте этому диапазону имя SumCriteria, введя его в поле имени слева от строки формул. Теперь выделите ячейку С1 и введите =\$А\$1, то есть ссылку на первую ячейку на листе. Скопируйте то же самое в ячейку Д1 и вы получите две копии заголовка столбца А. Эти копии будут использоваться как заголовки для условий БДСУММ (DSUM) (С1:Д2), который вы назвали SumCriteria.

В ячейке С2 введите >10 . В ячейке Д2 введите <20 . В ячейке, где должен быть результат, введите следующий код:

```
=DSUM($A$1:$A$100,$A$1,SumCriteria)
```

В русской версии Excel

```
=БДСУММ($A$1:$A$100,$A$1,SumCriteria)
```

Функция БДСУММ (DSUM) — это предпочтительный и самый эффективный способ работы с ячейками, отвечающими определенному критерию. В отличие от массивов, встроенные функции баз данных разработаны специально для этой

цели, и, даже когда они ссылаются на очень большой диапазон и используются в большом количестве, их отрицательное воздействие на скорость пересчета и эффективность достаточно мало по сравнению с формулами массива.

Т Р Ю К
№20

Выделение строк или столбцов через один

Вы определенно встречались с таблицами Excel, в которых цвета строк чередуются. Например, у нечетных строк белый цвет, у четных — серый. Это легко сделать при помощи условного форматирования.

Чередующиеся цвета и тени придают таблице профессиональный вид и упрощают чтение данных. Это форматирование можно наложить вручную, однако, как можно представить (или вспомнить из собственного опыта), это очень долгая задача, требующая постоянного обновления по мере добавления и удаления данных из таблицы. Кроме того, она требует бесконечного терпения. К счастью, *условное форматирование* может сократить объем необходимого терпения и улучшить ваш профессиональный образ.

Мы будем предполагать, что данные занимают диапазон **A1:H100**. Выделите этот диапазон ячеек, начиная с ячейки A1, гарантируя, таким образом, что A1 будет активной ячейкой выделения. Теперь выберите команду **Формат** ► **Условное форматирование** (**Format** ► **Conditional Formatting**). В раскрывающемся списке, где вы видите параметр **Значение** (**Cell Value Is**), выберите вариант **Формула** (**Formula Is**). В поле **Формула** (**Formula**) введите следующую формулу (рис. 2.5):

```
=MOD(ROW(),2)
```

В русской версии Excel

```
=ОСТАТ(СТРОКА(),2)
```

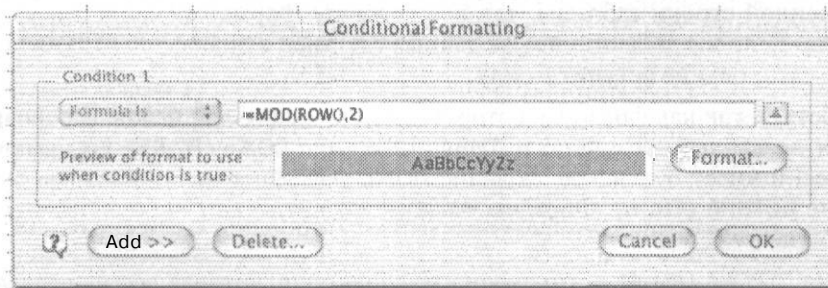


Рис. 2.5. Диалоговое окно условного форматирования с формулой для форматирования каждой второй строки в диапазоне

Щелкните кнопку **Формат** (**Format**) и выберите форматирование, которое будет применяться к каждой второй строке. Щелкните на кнопке **OK**, затем еще раз щелкните на кнопке **OK**. Указанный формат будет применен к каждой второй строке

в диапазоне **A1:H100**. Кроме того, вы должны сэкономить еще немного терпения до конца рабочего дня.

Если форматирование нужно применить к столбцам, а не к строкам, то используйте эту формулу:

```
=MOD(COLUMN(),2)
```

В русском варианте Excel

```
=ОСТАТ(СТОЛБЕЦ(),2)
```

Хотя в этом методе указанное форматирование легко и просто применяется к каждой второй строке или столбцу, оно не является динамическим. Строки, не содержащие данных, все также будут содержать форматирование. Это выглядит немного неаккуратно и затрудняет чтение таблицы. Чтобы сделать форматирование каждой второй строки или столбца динамическим, нужно еще немного поработать с формулами.

Снова выделите диапазон **A1:H100**, следя за тем, чтобы **A1** была активной ячейкой. Выберите команду **Формат** ► **Условное форматирование** (**Format** ► **Conditional Formatting**) и в раскрывающемся списке **Значение** (**Cell Value Is**) выберите **Формула** (**Formula Is**). В поле **Формула** (**Formula**) введите следующую формулу:

```
=AND(MOD(ROW(),2),COUNTA($A1:$H1))
```

В русской версии Excel

```
=И(ОСТАТ(СТРОКА(),2),СЧЁТЗ($A1:$H1))
```

ПРИМЕЧАНИЕ

Обратите внимание, что ссылки на строки не абсолютные (без знака доллара), а на столбцы — абсолютные.

Щелкните в этом диалоговом окне кнопку **Формат** (**Format**) и **выберите** нужное форматирование, затем щелкните две кнопки **ОК**. Форматирование не будет применяться к строкам диапазона **A1:H100**, где нет данных. Если вы удалите данные определенной строки таблицы, для нее условное форматирование также будет отменено. Если вы добавите новые данные где-либо в диапазоне **A1:H100**, условное форматирование вступит в силу.

Это работает, так как формула, которую вы указываете для условного форматирования, возвращает результат **ИСТИНА** (**TRUE**) или **ЛОЖЬ** (**FALSE**). На языке формул Excel **0** имеет булевское значение **ЛОЖЬ** (**FALSE**), а число, большее нуля, — значение **ИСТИНА** (**TRUE**). Формула `=MOD(ROW(),2)` вернет либо **0** (**ЛОЖЬ**), либо число, большее **0**.

Функция **СТРОКА** (**ROW**). — это переменная функция, которая всегда возвращает номер строки для ячейки, в которой находится. Функция **ОСТАТ** (**MOD**) возвращает остаток от деления одного числа на другое. В случае наших формул мы делим номер строки на **2**, поэтому для всех четных строк будет возвращаться результат **0**, а для всех нечетных — число, большее **0**.

Когда вы помещаете функцию **СТРОКА** (**ROW**) и функцию **СЧЁТЗ** (**COUNTA**) в функцию **И** (**AND**), то для того чтобы функция **И** (**AND**) вернула значение **ИСТИНА** (**TRUE**),

и функция **ОСТАТ** (MOD), и функция **СЧЁТЗ** (COUNTA) должны вернуть значение **ИСТИНА** (TRUE). Функция **СЧЁТЗ** (COUNTA) подсчитывает все непустые ячейки.

ТРЮК
№ 21

Трехмерные эффекты в таблицах и ячейках

Когда вы видите искусные трехмерные эффекты, например, в Excel, в действительности это иллюзия, созданная при помощи специального форматирования. Эту иллюзию легко создать самостоятельно, наложив форматирование на ячейки или диапазон ячеек.

Начнем с простого примера — мы придадим ячейке эффект объемности, чтобы она выглядела поднятой, как кнопка. На чистом рабочем листе выберите ячейку D5. (Мы выбираем ячейку D5 потому, что она находится не в углу.) Выберите команду **Формат** ▶ **Ячейки** ▶ **Граница** (Format ▶ Cells ▶ Border). В поле **Линия** (Line) выберите вторую по толщине линию. Убедитесь, что выбран черный цвет или **Авто** (Automatic), если вы не меняли значение по умолчанию для этого параметра. Теперь щелкните правую границу, а затем нижнюю границу. Выберите белый цвет. В поле **Линия** (Line) должна все так же быть выбрана вторая по толщине линия. Щелкните две оставшиеся границы ячейки — верхнюю и левую. Перейдите на вкладку **Вид** (Patterns) диалогового окна **Формат ячеек** (Format Cells) и выберите серый цвет. Щелкните на кнопке **ОК** и снимите выделение ячейки D5. Теперь эта ячейка будет выглядеть поднятой, как кнопка. Это достигается при помощи границ и теней.

Если для разнообразия вы хотите, чтобы ячейка выглядела вогнутой или нажатой, выделите ячейку E5 (так как она рядом с D5 и пригодится для следующего упражнения). Выберите команду **Формат** ▶ **Ячейки** ▶ **Граница** (Format ▶ Cells ▶ Border), выберите среди линий вторую по толщине и удостоверьтесь, что выбран черный цвет.

Примените форматирование к верхней и левой границам ячейки. Выберите белый цвет и наложите белую линию на правую и нижнюю границы. Перейдите на вкладку **Вид** (Patterns) и выберите серый цвет. Щелкните на кнопке **ОК**. Ячейка E5 теперь должна выглядеть вогнутой. Так она выглядит еще лучше рядом с ячейкой D5 с противоположным эффектом.

Наложение трехмерного эффекта на таблицу данных

Теперь мы поэкспериментируем с этим инструментом, чтобы увидеть, какие эффекты можно применить к таблицам, чтобы придать им объемный вид.

Выделите ячейки D5 и E5 и щелкните кнопку инструмента **Формат по образцу** (Format Painter) (это значок кисти) на стандартной панели инструментов. Удерживая левую кнопку мыши, щелкните ячейку F5, а затем перетащите указатель к ячейке J5 и отпустите.

Теперь выделите ячейки D5:J5 и снова щелкните кнопку **Формат по образцу** (Format Painter) на стандартной панели инструментов. Удерживая левую кнопку мыши,

щелкните ячейку D6 и перетащите указатель к ячейке J15, а затем отпустите. Результат должен быть похож на представленный на рис. 2.6.

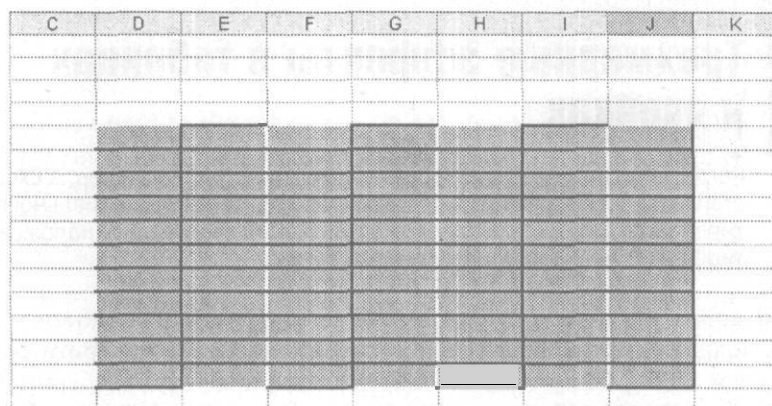


Рис. 2.6. Трехмерный эффект для диапазона ячеек

Мы выбрали достаточно толстую границу, чтобы гарантировать, что эффект будет ясно виден; однако можно сделать его немного мягче, выбрав более тонкую линию. Можно также использовать другой стиль линии, чтобы добиться еще большего эффекта. Самый простой способ найти хорошие комбинации — это метод проб и ошибок, который нужно применить на пустом листе для поиска нужного эффекта. Ограничения будут накладываться только вашим воображением и, возможно, вашим вкусом.

ВНИМАНИЕ

Всегда помните, что трехмерные эффекты могут улучшить читаемость и придать электронным таблицам более профессиональный вид, однако, если применять их чрезмерно, эффект будет противоположным. Помните, что все нужно использовать в меру.

Если вы хотите продвинуться дальше и автоматически и динамически применять трехмерные эффекты, можете комбинировать их с условным форматированием, автоматизируя наложение выбранных стилей.

Т Р Ю К
№ 22

Включение и выключение условного форматирования и проверки данных при помощи флажка

Проверка данных существенно уменьшает вероятность того, что пользователь случайно введет неправильные данные. Иногда, однако, возникает желание по-иному упростить ввод данных, которые бы помечались как неправильные с помощью условного форматирования и были бы заблокированы средствами проверки.

Обычно, чтобы разрешить пользователям вводить данные, которые могли бы быть расценены как неправильные, из ячеек удаляется условное форматирование или проверка данных. Однако есть более простой путь: можно совместить простой флажок с панели инструментов Формы (Forms) с проверкой данных.

Для этого примера мы применим условное форматирование к диапазону ячеек, чтобы любые данные, встречающиеся более одного раза, выделялись для упрощения идентификации. Предположим, что в таблице данные расположены в диапазоне $\$A\$1:\$H\100 . Условное форматирование этого диапазона данных для упрощения идентификации повторяющихся данных требует выполнить несколько шагов.

Выделите ячейку K1 и присвойте ей имя `CheckBoxLink`, введя его в поле имени слева от строки формул и нажав клавишу `Enter`. Если панель инструментов Формы (Forms) не видна, правой кнопкой мыши щелкните любую панель инструментов и выберите команду Формы (Forms), а затем щелкните значок флажка. Теперь щелкните лист где-либо за пределами диапазона $\$A\$1:\$H\100 , чтобы добавить флажок.

Правой кнопкой мыши щелкните флажок и в контекстном меню выберите команду **Формат объекта** ► **Элемент управления** (**Format Control** ► **Control**). В поле **Связь с ячейкой** (Cell Link) введите имя `CheckBoxLink` и щелкните кнопку **ОК**. Выберите ячейку A1, а затем перетащите указатель, чтобы выделить диапазон ячеек до H100. Важно, чтобы ячейка A1 была при выделении активной. Выберите команду **Формат** ► **Условное форматирование** (**Format** ► **Conditional Formatting**) и в поле с параметром **Значение** (Value Is) выберите вариант **Формула** (Formula Is). В поле справа введите следующую формулу (рис. 2.7):

```
=AND(COUNTIF($A$1:$H$100,A1)>1,CheckBoxLink)
```

В русской версии Excel

```
=И(СЧЁТЕСЛИ($A$1:$H$100,A1)>1,CheckBoxLink)
```

Щелкните кнопку **Формат** (**Format**), перейдите на вкладку **Вид** (Patterns) и выберите цвет, которым будут выделяться дублирующиеся данные. Щелкните кнопку **ОК**, затем еще раз щелкните кнопку **ОК**.

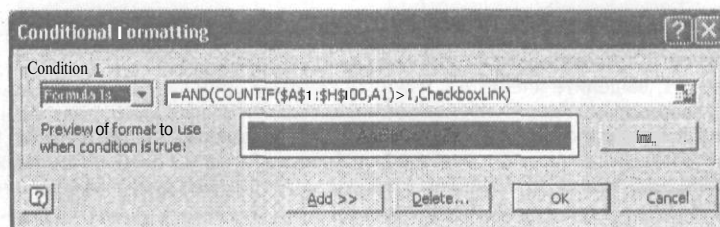


Рис. 2.7. Диалоговое окно с формулой условного форматирования диапазона и выделения повторяющихся данных

Когда флажок, который вы добавили на лист, будет установлен, связь в ячейке K1 (`CheckBoxLink`) возвратит значение **ИСТИНА** (TRUE), и все дублирующиеся значения в диапазоне $\$A\$1:\$H\100 будут подсвечены. Когда вы сбросите флажок, связь в ячейке возвратит значение **ЛОЖЬ** (FALSE) и дубликаты подсвечены не будут.

Этот флажок представляет собой выключатель, при помощи которого вы можете включать и выключать условное форматирование в таблице, не переходя в диалоговое окно Условное форматирование (Conditional Formatting). Тот же принцип можно применять при проверке данных, используя формулы.

Этот способ работает благодаря функции И (AND). Она требует, чтобы произошло два события: функция СЧЁТЕСЛИ(\$A\$1:\$H\$100,A1)>1 (COUNTIF(\$A\$1:\$H\$100,A1)>1) должна вернуть значение ИСТИНА (TRUE), и связь с флажком в ячейке (CheckBox-Link) также должна вернуть значение ИСТИНА (TRUE). Другими словами, чтобы функция И (AND) вернула значение ИСТИНА (TRUE), оба условия должны быть истинными.

ТРЮК
№ 23

Поддержка нескольких списков в поле со списком

При работе с несколькими списками можно переключаться между ними, используя комбинацию переключателей и поля со списком (ComboBox).

В Excel предусмотрено множество способов выбора элементов списка, являются ли они именами, продуктами, днями недели — любыми элементами, составляющими список. Но чтобы получить доступ к вариантам одновременно нескольких списков, обычно требуется несколько элементов управления, например, три поля со списком (ComboBox) с панели инструментов Формы (Forms).

Вместо этого поле со списком можно использовать в комбинации с переключателями (option button, radio button) — они также находятся на панели инструментов Формы (Forms), чтобы список изменялся автоматически в зависимости от того, какой переключатель выбран. Чтобы увидеть, как это работает, введите числа от 1 до 7 в диапазоне A1:A7 на новом рабочем листе. В диапазоне B1:B7 введите дни недели, начиная с понедельника (Monday) и заканчивая воскресеньем (Sunday). В диапазоне C1:C7 введите названия месяцев от января (January) до июля (July).

ПРИМЕЧАНИЕ

Возможности автоматического заполнения Excel могут упростить эту задачу. Введите 1 в ячейке A1, выделите ячейку A1, нажмите клавишу Ctrl и левой кнопкой мыши щелкните маркер заполнения. Удерживая левую кнопку мыши и клавишу Ctrl, перетащите указатель к строке 7. Excel заполнит столбец числами. Затем введите Monday в ячейке B1 и дважды щелкните маркер заполнения этой ячейки. Введите January в ячейке C1 и дважды щелкните маркер заполнения этой ячейки. Excel заполнит столбец днями недели и месяцами.

Выберите команду Вид ► Панели инструментов ► Формы (View ► Toolbars ► Forms) и дважды щелкните элемент управления «переключатель» на панели инструментов. Затем три раза щелкните таблицу в разных местах, чтобы разместить на ней три кнопки выбора.

Теперь щелкните элемент управления Поле со списком (ComboBox) и снова щелкните таблицу, чтобы поместить на нее это поле. Используя маркеры перетаскива-

ния, придайте полю со списком нужный размер и разместите кнопки выбора прямо под полем со списком.

Правой кнопкой мыши щелкните первую кнопку выбора, в контекстном меню выберите команду Изменить текст (Edit Text), а затем замените слова Перекл. 1 (Option Button 1) словом Numbers. То же самое сделайте для Перекл. 2 (Option Button 2), заменив надпись словом Weekdays, и для Перекл. 3 (Option Button 3), заменив фразу словом Months (рис. 2.8).

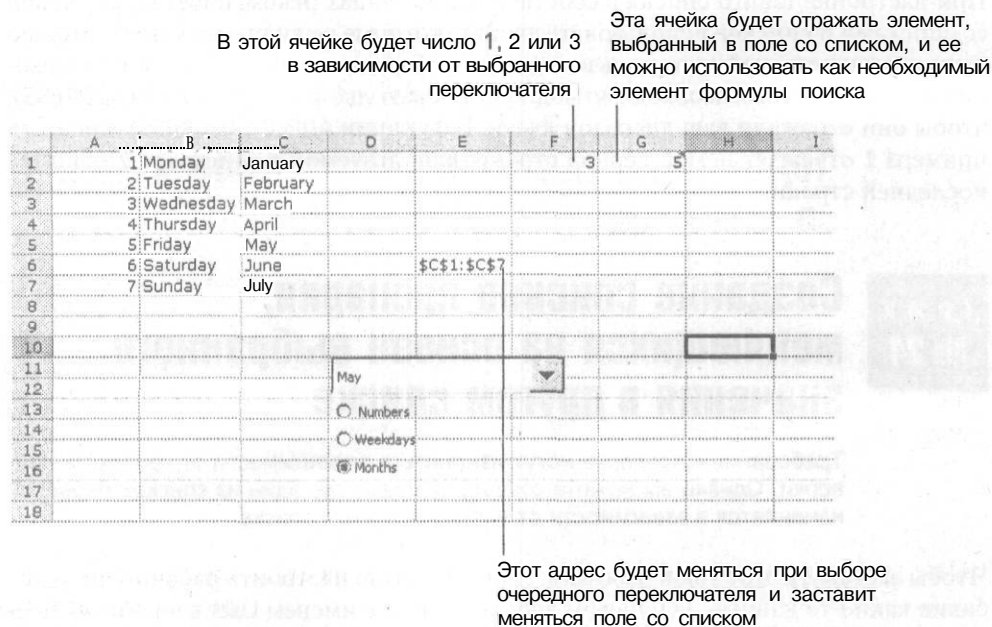


Рис. 2.8. Поле с несколькими списками, управляемое переключателями

Удерживая клавишу Ctrl, щелкните каждый переключатель, чтобы все три были выделены, а затем щелкните правой кнопкой мыши и в контекстном меню выберите команду Формат объекта ▶ Элемент управления (Format Control ▶ Control). Выберите связь с ячейкой \$F\$1 (удостоверьтесь, что связь абсолютная, используйте знак доллара).

В ячейке Е6 введите следующую формулу:

```
=ADDRESS(1,$F$1)&" ":"&ADDRESS(7,$F$1)
```

В русской версии Excel

```
=АДРЕС(1,$F$1)&" ":"&АДРЕС(7,$F$1)
```

Выберите команду Вставка ▶ Имя ▶ Присвоить (Insert ▶ Name ▶ Define). В поле Имя (Names in workbook) введите MyRange, а в поле Формула (Refers to) введите следующее:

```
=INDIRECT($E$6)
```

В русской версии Excel

```
=ДВССЫЛ($E$6)
```

Щелкните кнопку Add (Добавить), затем кнопку ОК. Правой кнопкой мыши щелкните элемент управления «поле со списком» и выберите команду Формат объекта ▶ Элемент управления (Format Control ▶ Control). Для параметра Форматировать список по диапазону (Input range) введите MyRange и укажите связь с ячейкой \$G\$1, затем щелкните кнопку ОК. Теперь можно выбрать один из переключателей, и список в поле со списком автоматически изменится в соответствии с установленным переключателем.

При настройке такого списка в собственных таблицах рекомендуется для полей со списками и списков использовать ячейки, которые не видны на экране. Можно даже скрыть эти ячейки от пользователей, чтобы связи всегда оставались на своем месте. Кроме того, необходимо модифицировать две функции АДРЕС (ADDRESS), чтобы они отражали ваш диапазон ячеек. В функции АДРЕС (ADDRESS) для этого примера 1 отражает номер первой строки, используемой в списках, а 7 — номер последней строки.



Создание списков проверки, меняющихся на основе выбранного значения в другом списке

Требования к проверке могут изменяться в зависимости от контекста проверки. Однако вы можете создать таблицу, где один из списков проверки изменяется в зависимости от выбора в другом списке.

Чтобы заставить этот трюк работать, сначала нужно настроить рабочий лист, добавив какие-то данные. На чистом рабочем листе с именем Lists в ячейке A1 введите заголовок Objects. В ячейке B1 введите заголовок Corresponding List. В ячейках A2:A5 введите слово Sap. В ячейках A6:A9 введите слово Sofa. В ячейках A10:A13 введите слово Shower. В ячейках A14:A17 введите слово Car. Затем, начиная с ячейки B2 и заканчивая ячейкой B17, введите следующие слова (соответствующие списку Objects): Tin, Steel, Opener, Lid, Bed, Seat, Lounge, Cushion, Rain, Hot, Cold, Warm, Trip, Journey, Bonnet и Boot.

В ячейке C1 введите заголовок Validation List. Затем, чтобы создать список уникальных элементов, введите слово Sap в ячейку C2, слово Sofa в ячейку C3, слово Shower в ячейку C4 и слово Car в ячейку C5.

ПРИМЕЧАНИЕ

Список уникальных элементов можно также создать с помощью средства Расширенный фильтр (Advanced Filter). Выделите ячейки A1:A17, выберите команду Данные ▶ Фильтр ▶ Расширенный фильтр (Data ▶ Filter ▶ Advanced Filter) и установите флажки Только уникальные записи (Unique Records Only) и Фильтровать список на месте (Filter the List in Place). Щелкните кнопку ОК, затем выделите ячейки A2:A14 (которые будут включать скрытые ячейки). Скопируйте их и вставьте в ячейку A18. Выберите команду Данные ▶ Фильтр ▶ Отобразить все (Data ▶ Filter ▶ Show All), выберите список уникальных объектов, вырежьте их и вставьте в ячейку C2. Теперь у вас есть список!

Выберите команду Вставка ▶ Имя ▶ Присвоить (Insert ▶ Name ▶ Define) и в поле Имя (Names in Workbook) введите слово Objects. В поле Формула (Refers to) введите следующую формулу и щелкните на кнопке Добавить (Add):

```
=OFFSET($A$2,0,0,COUNTA($A$1:$A$20),1)
```

В русской версии Excel

```
=СМЕЩ($A$2,0,0,СЧЁТЗ($A$1:$A$20),1)
```

В поле Имя (Names in Workbook) введите слово ValList, а в поле Формула (Refers to) введите \$C\$2:\$C\$5. Щелкните на кнопке Добавить (Add). Теперь вставьте еще один рабочий лист, назовите его Sheet1 и приготовьтесь к тому, что вам придется заставить эти странные данные работать.

Сделайте лист Sheet1 активным и выберите пункт меню Вставка ▶ Имя ▶ Присвоить (Insert ▶ Name ▶ Define). В поле Имя (Names in Workbook) введите слово CorrespondingList, а в поле Формула (Refers to) введите следующую длинную формулу и щелкните на кнопке Добавить (Add):

```
=OFFSET(INDIRECT(ADDRESS(MATCH(Val1Cell,Objects,0)+1,2,.."Lists")),  
0,0,COUNTIF(Objects,Val1Cell),1)
```

В русской версии Excel

```
=СМЕЩ(ДВССЫЛ(АДРЕС(ПОИСКПОЗ(Val1Cell,Objects,0)+1,2,.."Lists")),  
0,0,СЧЁТЕСЛИ(Objects,Val1Cell),1)
```

В поле Имя (Names in Workbook) введите слово ValCell, а в поле Формула (Refers to) введите \$D\$6 и щелкните на кнопке Добавить (Add). В поле Имя (Names in Workbook) введите слово Val2Cell, а в поле Формула (Refers to) введите \$E\$6 и снова щелкните на кнопке Добавить (Add). Щелкните на кнопке ОК, чтобы вернуться на лист Sheet1 и затем выберите ячейку \$D\$6.

Это долгий процесс, но мы уже почти закончили.

Выберите команду Данные ▶ Проверка ▶ Параметры (Data ▶ Validation ▶ Settings). В поле Тип данных (Allow) выберите пункт Список (List), а в поле Источник (Source) введите =ValList. Удостоверьтесь, что установлен флажок Список допустимых значений (In-Cell), и щелкните на кнопке ОК.

Выделите ячейку E6 и снова выберите команду Данные ▶ Проверка ▶ Параметры (Data ▶ Validation ▶ Settings). В поле Тип данных (Allow) выберите пункт Список (List), а в поле Источник (Source) напечатайте =CorrespondingList. Затем удостоверьтесь, что установлен флажок Список допустимых значений (In-Cell), и щелкните на кнопке ОК. Выберите один из объектов в списке проверки в ячейке D6, и список проверки в ячейке E6 автоматически изменится в соответствии с выбранным объектом.

Теперь у вас есть очень удобный список проверки (рис. 2.9), содержимое которого меняется автоматически в зависимости от элемента, выбранного в другом списке. В любой из ячеек или диапазоне ячеек можно использовать один список выбора, содержащий до пяти отдельных списков.



Рис. 2.9. Два связанных списка проверки

ТРЮК
№ 25

Как при проверке данных заставить Excel использовать список на другом рабочем листе

Один из параметров, доступных при проверке данных, это параметр Список (List), то есть удобный раскрывающийся список, из которого пользователи могут выбрать определенные элементы. Но есть один недостаток — если вы попытаетесь сослаться на список, находящийся на другом рабочем листе, то получите сообщение, что это невозможно. К счастью, при помощи очередного трюка невозможное можно сделать возможным.

Заставить Excel при проверке данных ссылаться на список на другом рабочем листе можно двумя способами — при помощи именованных диапазонов и функции ДВССЫЛ (INDIRECT).

Способ 1. Именованные диапазоны

Вероятно, самый простой и быстрый способ выполнить эту задачу — присвоить имя диапазону, где размещается список. Для этого упражнения мы предполагаем, что диапазону присвоено имя **MyRange**. Выделите ячейку, в которой должен будет появиться этот раскрывающийся список, и выберите команду Данные ► Проверка (Data ► Validation). В поле Тип данных (Allow) выберите Список (List), а в поле Источник (Source) введите `=MyRange`. Щелкните на кнопке ОК. Теперь список (который находится на другом рабочем листе) можно использовать как список проверки.

Способ 2. Функция ДВССЫЛ (INDIRECT)

Функция ДВССЫЛ (INDIRECT) позволяет ссылаться на ячейку, содержащую текст, который представляет собой адрес ячейки. Ячейку, содержащую функцию ДВССЫЛ (INDIRECT), можно использовать как ссылочную ячейку и применять эту возможность для связи с рабочим листом, где находится нужный список.

Предположим, список находится на листе Sheet1 в диапазоне \$A\$1:\$A\$10. Щелкните любую ячейку на другом рабочем листе, где должен будет появиться этот список проверки. Затем выберите команду Данные ► Проверка (Data ► Validation) и в поле Тип данных (Allow) выберите пункт Список (List). В поле Источник (Source) введите следующую функцию:

```
=INDIRECT("Sheet1!$A$1:$A$10")
```

В русской версии Excel

```
=ДВССЫЛ("Sheet1!$A$1:$A$10")
```

Убедитесь, что флажок Список допустимых значений (In-Cell) установлен, и щелкните на кнопке ОК. Список на листе Sheet1 окажется в вашем раскрывающемся списке проверки.

Если имя рабочего листа, на котором расположен список, содержит пробелы, функцию ДВССЫЛ (INDIRECT) нужно записать так:

```
=INDIRECT("'Sheet 1'!$A$1:$A$10")
```

В русской версии Excel

```
=ДВССЫЛ("'Sheet Г !$A$1:$A$10")
```

Различие заключается в том, что здесь после первой кавычки стоит один апостроф, а второй апостроф находится перед восклицательным знаком. Апострофы ограничивают название листа.

ПРИМЕЧАНИЕ

Полезно использовать одиночные апострофы всегда, независимо от того, содержит имя пробелы или нет. С апострофами вы все так же сможете ссылаться на листы с именами без пробелов, но это и упростит внесение изменений позже.

Преимущества и недостатки обоих методов

У именованных диапазонов и функции ДВССЫЛ (INDIRECT) при использовании их для связи со списком на другом рабочем листе есть преимущества и недостатки.

Преимущество использования именованного диапазона в данном сценарии заключается в том, что изменение названия листа не повлияет на список проверки. Это подчеркивает недостаток функции ДВССЫЛ (INDIRECT) — а именно, любое изменение названия листа не будет автоматически отражаться в функции ДВССЫЛ (INDIRECT), поэтому придется вручную изменить функцию, указав новое название листа.

Преимущество функции ДВССЫЛ (INDIRECT): если из именованного диапазона будет удалена первая ячейка или строка либо последняя ячейка или строка, то именованный диапазон вернет ошибку #REF!. В этом недостаток именованных диапазонов — если удалить ячейки или строки из именованного диапазона, эти изменения не повлияют на список проверки.

Т Р Ю К
№ 26

Удаление ненужных символов при помощи замены

При импортировании или копировании и вставке данных из других источников в Excel в таблице появляются ненужные символы. При помощи этого трюка вы сможете избавиться от необходимости удалять их вручную.

Средство Excel Заменить (Replace) поможет удалить из таблицы ненужные символы — для этого нужно выполнить несколько дополнительных действий. Например, можно заменить ячейки, содержащие ненужные символы, пустым местом, (фактически, удалить их), чтобы они просто перестали существовать. Для этого необходимо знать символьный код символов, которые должны быть удалены. У всех символов есть определенный код, и Excel может сообщить его, если применить к ним функцию КОДСИМВ (CODE). Эта функция возвращает числовой код первого символа в текстовой строке. Возвращенный код соответствует набору символов, который используется на вашем компьютере.

Чтобы заставить трюк работать, выделите одну из ячеек, содержащую ненужный символ. В строке формул выделите символ и скопируйте его в буфер. Затем выделите любую неиспользуемую ячейку, например A1, и вставьте символ в эту ячейку.

В другую ячейку введите следующую формулу:

```
=CODE($A$1)
```

В русской версии Excel

```
=КОДСИМВ($A$1)
```

Функция вернет символьный код ненужного символа.

Выделите все данные и выберите команду Правка ▶ Заменить (Edit ▶ Replace). Щелкните поле Что (Find What), нажмите клавишу Alt (Apple) и введите 0, а затем код, который вернула функция КОДСИМВ (CODE). Если это код 163, нажмите клавишу Alt или Apple и введите 0163. Поле Заменить на (Replace With) должно остаться пустым. Щелкните Заменить все (Replace All). Этот трюк быстро удалит все ненужные символы с соответствующим символьным кодом. Повторите эти действия для всех ненужных символов.

Т Р Ю К
№ 27

Преобразование текстовых чисел в настоящие числа

Содержимое ячейки может выглядеть как число, особенно, если данные были импортированы, но использовать эти числа в вычислениях может быть затруднительно. Вот несколько способов преобразования этих «текстовых» чисел в настоящие числа.

Помните, что числа в Excel по умолчанию выравниваются по правому краю, а текст — по левому. Простой способ идентифицировать проблемные текстовые

числа в столбце, который вы считаете полностью числовым, — выделить столбец, выбрать команду **Формат** ▶ **Ячейки** ▶ **Выравнивание** (Format ▶ Cells ▶ Alignment), убедиться, что выбрано выравнивание по умолчанию По значению (General), и щелкнуть кнопку ОК. Расширьте столбец, и вы увидите, что все настоящие числа будут выровнены по правому краю, а проблемные текстовые числа — по левому. Даты также будут выровнены по правому краю, так как в основе даты лежит не что иное, как число¹.

Теперь, когда вы знаете, что у вас есть числа, отображаемые как текст, мы предлагаем быстрый и простой способ преобразования их в настоящие числа, чтобы Excel смог использовать эти числа в вычислениях. Скопируйте любую пустую ячейку и затем выделите ваш список чисел. Выберите команду **Правка** ▶ **Специальная вставка** (Edit ▶ Paste Special) и в группе **Вставить** (Paste) выберите переключатель **Значения** (Values). В группе **Операция** (Operation) выберите переключатель **Сложить** (Add) и щелкните на кнопке ОК.

Теперь все числа, отображаемые как текст, будут превращены в настоящие числа. Это происходит потому, что пустая ячейка имеет значение 0, и, когда вы добавляете любое число к числу, которое Excel считает текстом, то заставляете Excel преобразовывать текстовое число в настоящее число.

Эту же логику можно применить к некоторым стандартным функциям Excel, в частности, к текстовым функциям, например, **ТЕКСТ** (TEXT). Обычно, когда вы применяете любую текстовую функцию Excel и в качестве результата возвращается число, Excel возвращает это число как текстовое, а не числовое значение.

Предположим, у вас есть диапазон ячеек, начиная с \$A\$1. Каждая ячейка содержит знак доллара и какое-то число, за которым через пробел указано имя человека. Используя следующую формулу, в которой комбинируются две текстовые функции — **ЛЕВСИМВ** (LEFT) и **НАЙТИ** (FIND), можно получить это число:

```
=LEFT(A1,FIND(" ",A1)-1)
```

В русской версии Excel

```
=ЛЕВСИМВ(A1,НАЙТИ(" ",A1)-1)
```

Если ячейка A1 содержит данные \$22.70 Fred, то в результате формула вернет значение \$22.70. Однако оно будет возвращено как текст, а не как настоящее числовое значение; таким образом, по умолчанию в ячейке оно будет выровнено по левому краю.

Можно модифицировать формулу, чтобы результат возвращался не как текст, а как настоящее числовое значение, добавив к нему 0:

```
=LEFT(A1,FIND(" ",A1)-1)+0
```

В русской версии Excel

```
=ЛЕВСИМВ(A1,НАЙТИ(" ",A1)-1)+0
```

¹ В современных версиях для этой цели служит смарт-тег ошибки ввода. Добиться повторения результата автора нам не удалось. Операция сложения (с ненулевым значением!) выполняется, но выравнивание соответствует текстовому формату чисел в ячейке. — *Примеч. ред.*

Это заставит Excel вернуть значение как число, таким образом, по умолчанию оно будет выровнено по правому краю. Все, что нужно после этого сделать, — соответствующим образом отформатировать ячейку.

Еще одна проблема может возникнуть относительно текста и чисел, когда в одной ячейке используются и числа, и текст и не существует удобного способа выделить из нее только числовую часть. В этом случае можно применить пользовательскую функцию, чтобы выделить числовую часть из текстовой строки.

Чтобы создать пользовательскую функцию, нажмите сочетание клавиш Alt/Option+F11, выберите команду Insert ► Module и введите код из листинга 2.1.

Листинг 2.1

```
Function ExtractNumber(rCell As Range)
Dim lCount As Long, l As Long
Dim sText As String
Dim lNum As String

sText = rCell

For lCount = Len(sText) To 1 Step -1
    If IsNumeric(Mid(sText, lCount, 1)) Then
        l = lCount
        lNum = Mid(sText, lCount, 1) & lNum
    End If
Next lCount

If l = 1 Then lNum = CInt(Mid(lNum, 1, l))

ExtractNumber = CLng(lNum)
End Function
```

Нажмите сочетание клавиш Alt/Apple+Q, чтобы сохранить функцию. В диалоговом окне Мастер функций (Paste Function) она появится в области Определенные пользователем (User Defined) (рис. 2.10).

	A	B	C
1	hjl4566klo59	456659	=ExtractNumber(A1)
2	jkh4025lop596	4025596	=ExtractNumber(A2)
3	hqt548kl92	54892	=ExtractNumber(A3)
4			
5			

Рис. 2.10. Выделение числовой части из текстовой строки

На рис. 2.10 столбец А содержит текст вместе с числами, столбец В содержит результат выполнения функции ExtractNumber, а столбец С демонстрирует, как формула выглядит в столбце В.

Т Р Ю К
№ 28

Настройка примечаний ячеек

Примечания ячеек позволяют присваивать электронный эквивалент записки на клейком листике любой ячейке на рабочем листе. Хотя многие пользователи создают примечания к ячейкам, не все знают, что примечания можно настраивать.

При вставке примечания в ячейку командой Вставка ▶ Примечание (Insert ▶ Comment) Excel по умолчанию вставляет в примечание имя пользователя на используемом компьютере. Эту настройку можно изменить, выбрав команду Сервис ▶ Параметры ▶ Общие (Tools ▶ Options ▶ General). Имя пользователя можно изменить внизу открывшегося диалогового окна. Текст, который вы введете, будет отображаться по умолчанию.

Хотя примечания ячеек выполняют простую функцию отображения сообщения для вас или другого пользователя, можно настроить их, чтобы они лучше отражали ваши намерения.

Удостоверьтесь, что открыта панель инструментов Рисование (Drawing) — выберите команду Вид > Панели инструментов ▶ Рисование (View ▶ Toolbars ▶ Drawing). Добавьте примечание к ячейке, выделив ячейку, а затем выбрав команду Вставка ▶ Примечание (Insert ▶ Comment). Будет автоматически включен режим редактирования, и вы сможете ввести текст в поле примечания.

Щелкните вне границы примечания, чтобы выйти из режима редактирования¹. Выделив примечание², на панели инструментов Рисование (Drawing) щелкните кнопку Действия (Draw), затем выберите команду Изменить автофигуру (Change AutoShape). Вы увидите вложенное меню командами Основные фигуры (Basic Shapes), Фигурные стрелки (Block Arrow), Блок-схема (Flow Chart), Звезды и ленты (Stars and Banners) и Выноски (Callouts). Выберите нужную фигуру, и форма примечания изменится (рис. 2.11).

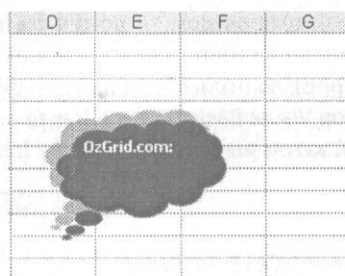


Рис. 2.11. Совершенно другое примечание ячейки

Вы можете пойти еще дальше, наложив на примечание затенение, чтобы сделать его еще более реальным и объемным³. Удостоверьтесь, что примечание все так же

¹ Не нужно этого делать! — Примеч. ред.

² Для этого можно воспользоваться командой контекстного меню ячейки. — Примеч. ред.

³ Для придания примечанию объемного вида нужно воспользоваться соответствующей кнопкой панели рисования. — Примеч. ред.

выделено, но режим правки не **включен**¹. На панели инструментов Рисование (Drawing) щелкните кнопку Тень (Shadow Settings) (рис. 2.12) и выберите нужную форму тени.

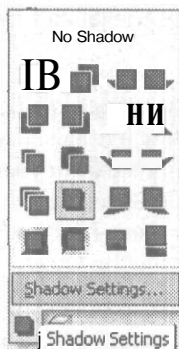


Рис. 2.12. Палитра теней

Еще одно интересное действие, которое можно применить к примечаниям, — заставить их отображать рисунки. Можно, например, вставить в примечание картинку с диаграммой, чтобы лучше иллюстрировать ее, не отображая саму диаграмму.

Чтобы добавить рисунок, удостоверьтесь, что примечание **выделено**, но режим правки **выключен**². Правой кнопкой мыши щелкните примечание и в контекстном меню выберите команду Формат примечания (Format Comment) или дважды щелкните границу примечания. В диалоговом окне Формат примечания (Format Comment) перейдите на вкладку Цвета и линии (Colors and Lines). На палитре Color (Цвет) выберите пункт Способы заливки (Fill Effects) и в диалоговом окне Способы заливки (Fill Effects) перейдите на вкладку Рисунок (Picture). Найдите рисунок для вставки в примечание.

Все, что осталось сделать с примечанием, — получить текст, который был вставлен в примечание, и заставить его появиться в ячейке. Для этого в стандартный модуль нужно поместить простую **пользовательскую** функцию. Выберите команду Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor) (сочетание клавиш **Alt/Option+F11**), затем выберите команду Insert ► Module и введите код, показанный в листинге 2.2.

Листинг 2.2

```
Function GetCommentText(rCommentCell As Range)
Dim strGotIt As String
    On Error Resume Next
    strGotIt = WorksheetFunction.Clean (rCommentCell.Comment.Text)
    GetCommentText = strGotIt
    On Error GoTo 0
End Function
```

¹ Примечание должно находиться в режиме изменения. — *Примеч. ред.*

² Речь о том, чтобы добиться форматирования не только текста... — *Примеч. ред.*

Чтобы вернуться в Excel, щелкните кнопку Close или нажмите сочетание клавиш Alt/Apple+Q. Теперь в любой ячейке введите следующую формулу:

```
=GetCommentText(A1)
```

где A1 — это ячейка с примечанием. Текст примечания появится в ячейке.

ТРЮК
№ 29

Сортировка более чем по трем столбцам

Средство Сортировка (Sort) в Excel ограничено возможностью сортировки только по трем полям. В большинстве случаев этого достаточно, но иногда бывает удобно сортировать более чем по трем столбцам данных. Вот как можно обойти это ограничение.

Для этого примера мы предполагаем, что у вас есть некоторым образом связанные данные в столбцах A, B, C, D и E, и вы хотите отсортировать эти данные сначала по столбцу A, затем по столбцу B, затем по C, по D и по E. Для этого нужно уметь сортировать в обратном порядке, другими словами, сначала сортировать по последнему полю, а затем по очереди продвигаться к первому.

Выделите столбцы с A по E и выберите команду Данные ▸ Сортировка (Data ▸ Sort). Выберите порядок сортировки: сначала должен сортироваться столбец C, затем D, затем E. Щелкните кнопку ОК (Sort). Теперь снова выделите столбцы с A по E и выберите команду Данные ▸ Сортировка (Data ▸ Sort). В этот раз сначала проведите сортировку по A, а затем по B. Щелкните кнопку ОК (Sort), и все будет в порядке. Excel отсортирует столбцы по пяти полям вместо обычных трех.

Если вы хотите автоматизировать эту задачу, можно использовать макрос, который будет сортировать выделенный диапазон и угадывать, есть ли в столбцах данных заголовки, основываясь на форматировании первой строки выделенного диапазона. Если заголовки выделены жирным шрифтом, Excel поймет, что это заголовки столбцов, и не будет сортировать их. Сначала макрос проведет сортировку по самому левому столбцу, затем по следующему и так далее до самого правого. Можно указать до 256 столбцов.

Код этого макроса нужно поместить в стандартный модуль. Чтобы попасть туда, выберите команду Сервис ▸ Макрос ▸ Редактор Visual Basic (Tools ▸ Macro ▸ Visual Basic Editor) (Alt/Option+F11), затем выберите команду Insert ▸ Module и введите код из листинга 2.3.

Листинг 2.3

```
Sub SortByX()  
Dim l As Long
```

```
For l = Selection.Columns.Count To 1 Step -1  
    Selection.Sort Key1:=Selection.Cells(2, l), _  
        Order1:=xlAscending, Header:=xlGuess, Orientation:=xlTopToBottom  
Next l  
End Sub
```

Чтобы вернуться в Excel, закройте окно или нажмите сочетание клавиш Alt/Apple+Q. Создав этот макрос, вы сможете выполнять намного более сложную сортировку, чем предлагают стандартные возможности Excel.



Случайная сортировка

В Excel можно случайным образом выбрать трех победителей — 1-го, 2-го и 3-го — из списка в электронной таблице. Самый простой и честный способ сделать это — использовать функцию СЛЧИС (RAND) Excel совместно с его возможностями сортировки.

Предположим, что в вашей электронной таблице есть таблица из трех столбцов начиная со столбца В с заголовками столбцов Name, Age и ID No. Можно поместить функцию СЛЧИС (RAND) в ячейку А2 и скопировать ее вниз на необходимое количество строк до конца таблицы. Все ячейки в столбце А, в которые вы поместили функцию СЛЧИС (RAND), автоматически вернут случайное число, по которому вы сможете отсортировать таблицу. Другими словами, можно отсортировать столбцы А, В, С и D по столбцу А по возрастанию или убыванию, и тремя победителями будут три верхних имени.

Функция СЛЧИС (RAND) — это пересчитываемая функция, которая автоматически пересчитывается, какое бы действие в Excel ни произошло, — например, ввод данных или принудительный запуск пересчета клавишей F9. Поэтому лучше побыстрее записать имена победителей.

Однако вы можете использовать эту непостоянность в своих целях и записать макрос, который будет сортировать данные сразу после того, как вы пересчитаете функции и выполните функцию СПЧИС (RAND), чтобы вернуть очередной набор случайных чисел. Этот макрос можно назначить какой-либо кнопке, чтобы каждый раз когда понадобится выбрать трех победителей, пришлось бы только нажать кнопку и записать три верхних имени.

Предположим, например, у вас есть данные в столбцах В, С и D, и в строке 1 содержатся заголовки. Сначала в ячейку А1 поместите заголовок RAND. В ячейку А2 введите формулу =RAND() и скопируйте ее вниз на необходимое количество ячеек. После этого выделите любую ячейку и выберите команду Сервис ▶ Макрос ▶ Начать запись (Tools ▶ Macro ▶ Record New Macro).

Выделите столбцы А, В, С и D и нажмите клавишу F9 (чтобы запустить пересчет). Выберите команду Данные ▶ Сортировка (Data ▶ Sort) и отсортируйте данные по столбцу А. Остановите запись макроса.

Теперь выберите команду Вид ▶ Панели инструментов ▶ Формы (View ▶ Toolbars ▶ Forms). На этой панели инструментов выберите кнопку и поместите ее в любое место на рабочем листе. Назначьте этой кнопке только что записанный макрос и щелкните кнопку ОК. (Если хотите, измените текст на кнопке с Кнопка 1 (Button 1) на что-то более осмысленное.) Можно выделить столбец А и полностью скрыть его, так как пользователям не нужно видеть сгенерированные случайные

числа. Каждый раз когда вы будете щелкать кнопку, данные будут сортироваться случайным образом и вы сможете просто прочитать три верхних имени победителей (рис. 2.13).

	B	C	D	E	F	G
1	Name	Age	ID No.	Place	Pick Winner	
2	pave	21	11256	1		
3	Paul	19	11257	2		
4	Anne	23	11135	3		
5	Gemma	30	11248			
6	Peter	52	11356			
7	Fred	30	11236			
8	Nigel	36	11237			
9	Bill	25	11234			
10	Jill	52	11245			

Рис. 2.13. Результат случайной сортировки со скрытым столбцом A

ВНИМАНИЕ

У функции СЛЧИС (RAND) в Excel 2003 есть большой недостаток. Хотя в справке ясно указано, что возвращаются случайные числа от 0 до 1, это не всегда так, в частности, когда функция СЛЧИС (RAND) используется в нескольких ячейках. Иногда она возвращает числа, меньшие 0. Прочитать объяснение Microsoft по поводу изменения алгоритма можно на сайте по адресу: <http://support.microsoft.com/default.aspx?kbid=828795>.

ТРЮК
№ 31

Манипулирование данными при помощи расширенного фильтра

Если вы знакомы с средством Excel Автофильтр (AutoFilter), то вы также должны знать о его ограничениях. Для более гибких возможностей манипулирования данными пригодится инструмент Excel Расширенный фильтр (Advanced Filter).

Хотя инструмент Автофильтр (AutoFilter) в чем-то ограничен, он полезен, если вы хотите отобразить только данные, отвечающие определенным критериям. Иногда, однако, бывает невозможно подобрать необходимую информацию, используя только стандартные параметры автофильтра. Разносторонний инструмент Excel Расширенный фильтр (Advanced Filter) позволяет расширить возможности манипулирования данными.

При работе с инструментом Excel Расширенный фильтр (Advanced Filter) ваша таблица должна быть настроена согласно классическому формату (об этом рассказывается в начале главы 1).

Для применения инструмента Excel Расширенный фильтр (Advanced Filter) понадобится копия заголовков столбцов таблицы где-либо выше данных. Над таблицей данных всегда следует оставлять, по меньшей мере, три пустые строки. Чтобы гарантировать, что заголовки всегда останутся одними и теми же независимо от любых изменений, всегда ссылайтесь на них простыми ссылочными формулами, например, =A4, где A4 содержит заголовок столбца. Копируйте эту формулу столько

раз, сколько заголовков столбцов в вашей таблице. Это гарантирует, что заголовки условий для расширенного фильтра будут динамическими. Непосредственно под скопированными заголовками поместите условия, которые будут использоваться расширенным фильтром. Подробнее об этом — в справке по Excel в разделе Критерии расширенного фильтра (Advanced Filters Criteria).

Нужно всегда помнить, что при использовании расширенного фильтра, если вы поместите два или более критерия непосредственно под соответствующими заголовками, они будут разделены оператором ИЛИ (OR). Если вы хотите использовать оператор И (AND), заголовки столбцов и соответствующие критерии нужно указать дважды бок о бок. На рис. 2.14 показано, как для фильтрации данных и отображения только тех служащих, ставка заработной платы которых больше чем 16,0 долларов ИЛИ меньше чем 15,0 долларов, использовать оператор ИЛИ (OR), а на рис. 2.15 — оператор И (AND) для отображения только тех служащих, имя которых начинается с В и заканчивается на L.

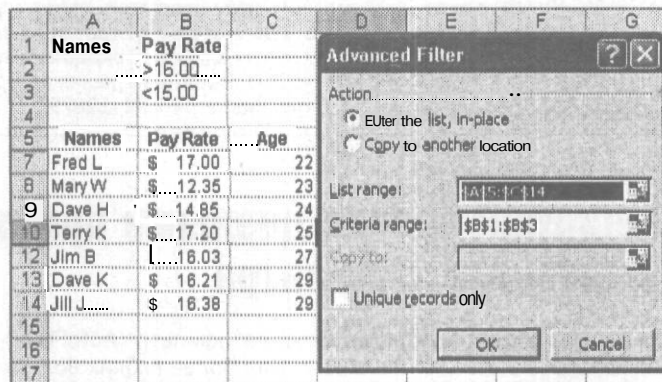


Рис. 2.14. Использование расширенного фильтра с оператором ИЛИ

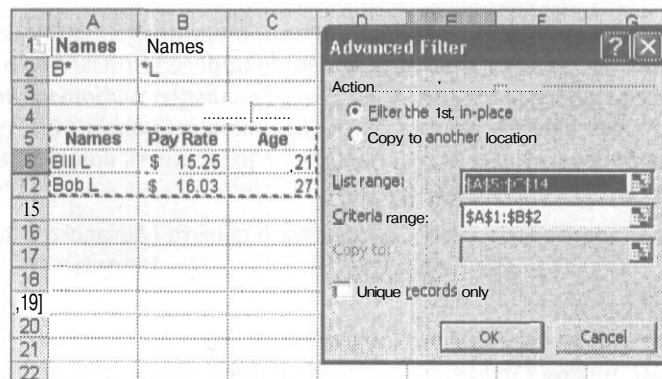


Рис. 2.15. Использование расширенного фильтра с оператором И

Оба предыдущих примера демонстрируют довольно простые случаи использования инструмента Расширенный фильтр (Advanced Filter), и эти задачи при необхо-

димости можно выполнить с помощью автофильтра. Далее мы продемонстрируем несколько примеров применения расширенного фильтра в случаях, когда возможностей автофильтра не хватает.

ПРИМЕЧАНИЕ

Важно помнить, что, когда для критериев используется формула, нельзя над критериями помещать заголовок, идентичный одному из заголовков в таблице. Например, **если** у вас есть список числовых данных в столбце A и список начинается в ячейке A5 (а в ячейке A4 находится заголовок), а вам необходимо получить все числа, большие среднего, условие будет выглядеть так:

```
=A5>AVERAGE($A$5:$A$500)
```

Если этот критерий находится в ячейке A2, диапазон критериев — \$A\$1:\$A\$2, но \$A\$1 не может содержать тот же заголовок, что и заголовок списка. Эта ячейка должна быть либо пустой, либо с другим заголовком.

Важно также помнить, что любая формула должна возвращать значение либо ИСТИНА (TRUE), либо ЛОЖЬ (FALSE). Диапазон, указанный для функции, вычисляющей среднее, указан в абсолютном формате при помощи знаков доллара, тогда как ссылка на ячейку A5 — это относительная ссылка. Это необходимо, поскольку при применении расширенного фильтра Excel увидит, что A5 — это относительная ссылка, и будет перемещаться вниз по списку по одной записи за раз и возвращать значение либо ИСТИНА (TRUE), либо ЛОЖЬ (FALSE). Если будет возвращено значение ИСТИНА (TRUE), данное число будет извлечено из списка. Если возвращено значение ЛОЖЬ (FALSE), число не отвечает критерию и, таким образом, не будет показано.

Предполагаем, что в диапазоне \$A\$5:\$A\$500 с заголовком в ячейке A4 многие имена повторяются. Кроме того, предполагаем, что многие заголовки повторяются несколько раз. Перед нами поставлена задача выделить в списке все повторяющиеся имена. Для этого нужно применить расширенный фильтр и в качестве критерия следующую формулу:

```
=COUNTIF($A$5:$A$500,A5)>1
```

В русской версии Excel:

```
=СЧЁТЕСЛИ($A$5:$A$500,A5)>1
```

Когда вы примените к данным начальным условиям расширенный фильтр с параметром Скопировать результат в другое место (Copy to Another Location), новый список будет содержать все имена, которые повторялись в исходном списке (рис. 2.16). Многие из этих имен и в новом списке будут повторены несколько раз, но легко отфильтровать его при помощи расширенного фильтра, на этот раз выбрав параметр Только уникальные записи (Unique Records Only) (рис. 2.17). Получим список повторяющихся имен.

ПРИМЕЧАНИЕ

Те, кто использует расширенный фильтр, часто спрашивают, как заставить Excel фильтровать данные по указанным точным критериям. Если ваш критерий — это имя Dave и вы применяете расширенный фильтр на длинном списке имен, Excel вернет не только имена Dave, но также такие имена, как Davey, Dave J, Dave K и т. д. Другими словами, любое имя, начинающееся с букв Dave именно в таком порядке, будет считаться удовлетворяющим условию. Чтобы Excel мог найти точные соответствия, в данном случае только имена Dave, введите критерий как **"=Dave"**.

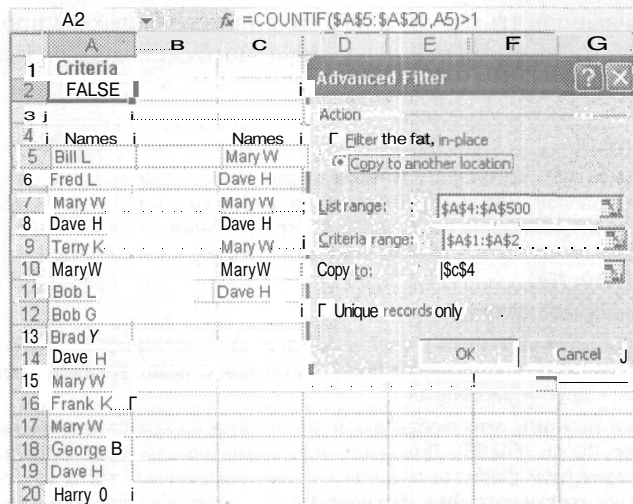


Рис. 2.16. Извлечение повторяющихся имен списка с помощью расширенного фильтра

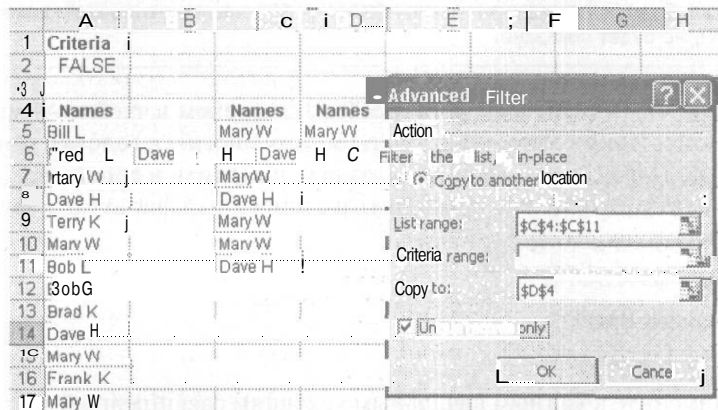


Рис. 2.17. Применение расширенного фильтра для получения уникальных записей

Т Р Ю К № 32

Создание пользовательских числовых форматов

В Excel предусмотрены встроенные числовые форматы, но иногда вам нужно использовать числовой формат, не встроенный в Excel. При помощи трюков этого раздела можно создавать числовые форматы и настраивать в соответствии с вашими требованиями.

Перед тем как приступать к изучению этих трюков, полезно понять, как Excel распознает форматы ячеек. С точки зрения Excel, формат ячейки состоит из четырех

разделов (слева направо): *положительные числа*, *отрицательные числа*, *нулевые значения* и *текстовые значения*. Они отделяются друг от друга точкой с запятой (;).

При создании пользовательского числового формата не обязательно указывать все четыре раздела. Другими словами, если вы создадите только два раздела, первый будет использоваться для положительных чисел и нулевых значений, а второй — для отрицательных чисел. Если создать только один раздел, все числовые типы будут выводиться в этом единственном формате. Пользовательские форматы влияют на текст, только когда вы указываете все четыре раздела; для текста будет применяться последний раздел.

ВНИМАНИЕ

Не следует считать, что пользовательские форматы применяются только к числовым данным. Числовые форматы применяются и к тексту.

Пользовательский числовой формат, показанный на рис. 2.18, — это стандартный формат валюты Excel, который отображает отрицательные значения валюты красным цветом. Мы модифицировали его, добавив отдельный формат для нулевых значений и текста. Если в качестве значения валюты вы введете положительное число, Excel автоматически отформатирует его, чтобы на месте разделительного знака для разряда тысяч вставлялась запятая, и в числе использовалось два десятичных разряда. То же самое будет сделано и для отрицательных значений, но они будут отображаться красным цветом. Для любого нулевого значения символ валюты отображаться не будет, но будут присутствовать два десятичных разряда. Если вы введете в ячейку текст, Excel отобразит сообщение «No Text Please», независимо от того, какой это будет текст.

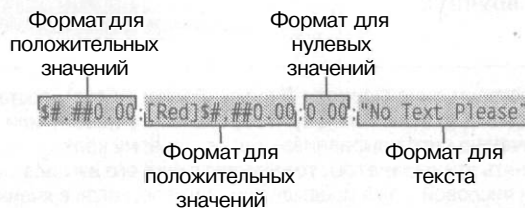


Рис. 2.18. Разделы пользовательского числового формата

Важно помнить, что форматирование значения в ячейке не влияет на его настоящее значение. Например, введите любое число в ячейку A1. Выберите команду **Формат > Ячейки > Число > Все форматы (Format > Cells > Number > Custom)** и, выбрав любой формат, введите «Hello» (в кавычках). Затем щелкните кнопку ОК.

Хотя в ячейке появится слово Hello, вы можете увидеть ее настоящее значение, выделив ячейку и взглянув на строку формул или нажав клавишу F2. Если вы хотите сослаться на эту ячейку в формуле, например, =A1+20, то в результирующей ячейке также будет применяться пользовательский формат. Если вы хотите сослаться на ячейку A1 вместе со многими другими ячейками со стандартным

форматом Excel, например, =SUM(A1:A10), то в результирующей ячейке также будет применяться пользовательский формат ячейки A1. Excel считает, что вы хотите, чтобы результирующая ячейка была отформатирована так же, как и ячейки, на которые ссылается формула. Если эти ячейки содержат несколько типов форматирования, любой пользовательский формат будет иметь преимущественное значение.

Это означает, что всегда необходимо помнить: Excel использует для вычислений настоящее значение ячейки, а не отображаемое. При этом могут возникнуть сюрпризы, если вычисления проводятся с использованием ячеек, отформатированных, например, без десятичных разрядов или всего с несколькими. Чтобы увидеть, как это работает, введите (в английской версии) 1.4 в ячейке A1 и 1.4 в ячейке A2, отформатируйте обе ячейки так, чтобы они не показывали десятичные разряды, а затем в другую ячейку поместите формулу =A1+A2. В результате, конечно же, получится 3, так как Excel округляет значение.

ПРИМЕЧАНИЕ

В Excel предусмотрен флажок **Точность**, как на экране (Precision as Displayed), который можно установить, выбрав команду Сервис ▶ Параметры ▶ Вычисления (Tools ▶ Options ▶ Calculation), но необходимо помнить, что этот параметр навсегда изменяет хранимые в ячейках значения с полной точности (15 цифр) до того формата, включая десятичные разряды, в котором значение выводится на экран. Другими словами, если вы установите этот флажок и щелкните кнопку ОК, пути назад уже не будет. (Можно попробовать, но дополнительная информация о точности значения исчезнет навсегда.)

Формат по умолчанию для любой ячейки — **Общий (General)**. Если вы вводите в ячейку число, Excel часто угадывает наиболее подходящий числовой формат. Например, если вы введете 10 %, Excel отформатирует эту ячейку процентным форматированием. Чаще всего Excel угадывает правильно, однако иногда формат приходится менять вручную.

ВНИМАНИЕ

Находясь в диалоговом окне **Формат ▶ Ячейки (Format ▶ Cells)**, постарайтесь избежать искушения принудительно включить форматирование по левому или правому краю или по центру. По умолчанию числа выравниваются по правому краю, а текст — по левому. Если вы не будете менять эти параметры, то сможете с первого взгляда сказать, является ячейка текстовой или числовой, как в предыдущем примере, когда в ячейке A1 выводился текст, но на самом деле она содержала число.

В каждом разделе данного формата используется собственный набор кодов форматов. Эти коды заставляют Excel выводить данные так, как вам это нужно. Так, например, предположим, что вы хотите, чтобы отрицательные числа выводились в круглых скобках, а для всех чисел (положительных, отрицательных и нулевого значения) выводились два десятичных разряда. Для этого используйте следующий пользовательский формат:

0.00_ :(-0.00)

Если вы также хотите, чтобы отрицательные числа выводились красным цветом, используйте формат

0.00_ : [Red](-0.00)

В русской версии Excel

0.00_ : [КРАСНЫЙ](-0.00)

Обратите внимание на квадратные скобки в предыдущем коде. Этот код формата заставляет Excel выводить число красным цветом.

В разделах пользовательского формата можно применять множество кодов форматов. В таблицах 2.1–2.5, полученных из документации Microsoft, объясняется значение этих кодов (в английской версии).

Таблица 2.1. Коды форматов

Числовой код	Описание
General	Общий числовой формат
0 (нуль)	Заполнитель разрядов числа, который заполняет значение нулями в соответствии с форматом
#	Заполнитель разрядов числа, не требующий вывода лишних нулей
?	Заполнитель разрядов числа, который оставляет пробелы для незначащих нулей, но не выводит их
%	Процент. Excel умножает число на 100 и выводит после числа символ %
, (запятая)	Разделитель тысяч (групп разрядов). Запятая, за которой следует заполнитель, увеличивает число на 1 000
E+ E- e+ e-	Экспоненциальное представление

Таблица 2.2. Текстовые коды

Текстовый код	Описание
\$ - + / () : и пробел	Эти символы в числе отображаются. Чтобы вывести любые другие символы, заключите их в кавычки или поставьте перед ними обратную косую черту
\символ	При помощи этого кода указанный символ можно вывести. Обратите внимание, что при вводе символов !, ^, &, ', ~, {, }, =, < или > перед символом автоматически появляется обратная косая черта
«текст»	Этот код отображает текст, заключенный в кавычки (без самих кавычек)
*	Этот код повторяет следующий символ в формате, заполняя столбец на всю ширину. В каждом разделе формата можно использовать только одну звездочку
_ (подчеркивание)	При помощи этого кода можно пропустить пространство, равное ширине следующего символа. Обычно он используется как «_»), чтобы оставить пространство для закрывающей скобки в формате положительного числа, когда формат отрицательного числа включает скобки. Это позволяет выравнивать положительные и отрицательные числа по десятичной точке
@	Заполнитель разряда для текста

Таблица 2.3. Коды даты

Код даты	Описание
M (M)	Месяц, представляемый как число без ведущих нулей (1-12)
Mm (Mm)	Месяц, представляемый как число с ведущими нулями (01-12)
Mmm (Mmm)	Месяц, отображаемый в виде аббревиатуры (Jan (январь) - Dec (декабрь))
D (D)	День, представляемый без ведущих нулей (1-31)
Dd (Dd)	День, представляемый с ведущими нулями (01-31)
Ddd (Ddd)	День недели, отображаемый в виде аббревиатуры (Sun (воскресенье) - Sat (суббота))
Dddd (Dddd)	День недели полностью (Sunday (воскресенье) - Saturday (суббота))
Yy (Yy)	Год двумя цифрами (например, 96)
Yyyy (Yyyy)	Год четырьмя цифрами (например, 1996)

Таблица 2.4. Коды времени

Код времени	Описание
H (H)	Часы как число без ведущих нулей (0-23)
Hh (Hh)	Часы как число с ведущими нулями (00-23)
m (m)	Минуты как число без ведущих нулей (0-59)
mm (mm)	Минуты как число с ведущими нулями (00-59)
s (s)	Секунды как число без ведущих нулей (0-59)
ss (ss)	Секунды как число с ведущими нулями (00-59)
AM/PMат/рм	Время дня в 12-часовом формате

Таблица 2.5. Прочие коды

Код	Описание
[BLACK], [BLUE], [CYAN], [GREEN], [MAGENTA], [RED], [WHITE], [YELLOW], [COLOR n] ([ЧЕРНЫЙ], [СИНИЙ], [ГОЛУБОЙ], [ЗЕЛЕНый], [РОЗОВый], [КРАСный], [БЕЛый], [ЖЕЛтый], [Цвет n])	Эти коды отображают символы определенным цветом. Обратите внимание, что n — это значение от 1 до 56 и обозначает n-й цвет цветовой палитры
[условие значение]	Условия: <, >, =, >=, <= и o; значение — любое число. Числовой формат может содержать до двух условий

Обратите особое внимание на последний код формата в табл. 2.5 — операторы сравнения. Предположим, вы хотите, чтобы пользовательский числовой формат 0.00_ ;[Red] (-0.00) (0.00_ ;[Красный] (-0,00)) отображал отрицательные числа красным цветом и в скобках, только если они меньше -100. Для этого укажите следующий формат:

0.00_ ;[Red][<-100](-0.00);0.00

В русской версии Excel

0.00_ ;[Красный][<-100](-0.00);0.00

Добиться нужного результата помогают коды формата `[Red][<-100](-0.00)` ([Красный][<-100](-0,00)) в разделе для отрицательных чисел. При помощи этого метода и условного форматирования вы можете увеличить количество условий в условном формате с трех до шести.

Часто пользователи хотят отобразить знак доллара словами. Для этого используйте следующий пользовательский формат:

О "Dollars and" .00 "Cents"

Этот формат позволит числа, например, 55.25 отобразить как 55 Dollars and .25 Cents. Если вы хотите преобразовывать числа в доллары и центы, то, чтобы узнать об этих двух пользовательских функциях Microsoft, обратитесь на сайты <http://www.ozgrid.com/VBA/ValueToWords.htm> и <http://www.ozgrid.com/VBA/CurrencyToWords.htm>.

Кроме того, при помощи пользовательского формата можно отображать слова Low, Average или High вместе с введенными словами. Укажите следующий код формата:

`[<11]"Low"* 0:[>20]"High"* 0;"Average"* 0`

Обратите внимание на звездочку. Она повторяет следующий символ в формате, заполняя столбец на всю ширину, благодаря чему слова Low, Average или High будут отформатированы по левому краю, а числа — по правому.



Увеличение количества уровней отмены действий в Excel для Windows

Все знакомы со знаменитым средством Отмена (Excel Undo), которая позволяет пользователям отменять только что сделанные ошибки. К сожалению, уровень отмены по умолчанию равен всего лишь 16 изменениям. При помощи трюка из этого раздела вы сможете изменить системный реестр так, чтобы отменить можно было до 100 ошибок.

При использовании средства Excel Отмена (Undo), когда вы достигаете 16-го действия, на место 17-го помещается первое и так далее. Кроме того, когда вы сохраняете рабочую книгу, стек отмены очищается и история отмены действий теряется. Это происходит потому, что, нажимая кнопку Сохранить (Save), вы сообщаете Excel, что удовлетворены сделанными изменениями, и он решает, что история отмены действий больше не нужна.

Иногда вы, вероятно, обнаруживали, что 16 изменений, хранящихся в стеке отмены, недостаточно. Вместо того чтобы смириться с этим, вы можете изменить этот параметр, отредактировав реестр (это работает только в Windows). Для этого полностью закройте Excel. Выберите команду Пуск ▶ Выполнить (Start ▶ Run) и в поле Открыть (Open) введите Regedit.exe. Щелкните кнопку ОК. В открывшемся окне редактора реестра разверните папку HKEY_CURRENT_USER. Затем разверните папку Software ▶ Microsoft ▶ Office, затем, например, папку 10.0 (название последней папки разное для различных версий Excel, например, 10.0 — это название для Excel 2002). Разверните папку Excel и, наконец, папку Options.

Выберите команду Правка ▶ Создать ▶ Параметр DWORD (Edit ▶ New ▶ DWORD Value), введите слово UndoHistory и нажмите клавишу Enter. Дважды щелкните только что созданный пункт UndoHistory и введите значение от 16 до 100.

Надеемся, что 100 отмен будет достаточно даже для самых требовательных пользователей, вопрос с очисткой стека отмены при сохранении, однако, остается нерешенным.

Т Р Ю К
№ 34

Создание пользовательских списков

Добавив в Excel пользовательский список, можно ввести первый элемент списка, перетащить его вниз при помощи маркера заполнения, и список будет автоматически заполнен.

Одной из самых популярных возможностей Excel, позволяющих экономить время, является умение автоматически вводить не только числа, но и определенного вида текст. В Excel есть пара встроенных списков, например, дни недели или месяцы года. Вы вводите первый элемент списка, а затем при помощи маркера заполнения заполняете ячейки следующими элементами списка. Можно легко создавать собственные списки для часто используемых элементов.

Самый гибкий способ создания пользовательского списка — это ввод содержимого списка в диапазон ячеек. Например, у вас есть список из 100 имен сотрудников. Введите все имена, начиная с ячейки A1 и заканчивая ячейкой A100, и, если необходимо, отсортируйте список. Затем выберите команду Сервис ▶ Параметры (Tools ▶ Options) и на вкладке Списки (Custom Lists) щелкните поле Импорт списка из ячеек (Import List from the Cells). Щелкните ячейку A1 и протащите указатель вниз на ячейку A100. После этого щелкните кнопку Импорт (Import). Теперь этот пользовательский список можно будет использовать во всех рабочих книгах на этом компьютере.

Теперь, когда у вас есть пользовательский список, можно перевернуть его. Для этого вернитесь к столбцу рядом со списком и поместите последнюю запись списка в верхнюю ячейку. В ячейку снизу поместите предпоследнюю запись. Выделите обе ячейки и дважды щелкните маркер заполнения. Созданный вами список должен перевернуться. Можно отсортировать исходный список командой Данные ▶ Сортировка ▶ Параметры (Data ▶ Sort ▶ Options), присвоить ему имя и затем отсортировать его по-другому.

ПРИМЕЧАНИЕ

Если список отсортирован и вы хотите перевернуть его, вероятно, будет легче отсортировать его в обратном порядке, от Z до A, если исходный список был отсортирован от A до Z.

Т Р Ю К
№ 35

Промежуточные итоги жирным шрифтом

Было бы здорово, если бы можно было легко находить и идентифицировать промежуточные итоги на рабочих листах.

Когда вы работаете с таблицей, в которой есть промежуточные итоги, созданные командой Данные ► Итоги (Data ► Subtotals), иногда бывает трудно отличить итоги от других данных, поэтому читать таблицу тяжело. Это особенно часто случается, если промежуточные итоги были созданы для таблицы данных с множеством столбцов.

Обычно результирующие промежуточные итоги появляются справа, а соответствующие заголовки — в первом столбце. Поскольку значения промежуточных итогов не выделяются жирным шрифтом, бывает трудно визуально сопоставить их с соответствующими заголовками строк. Применяв к значениям промежуточных итогов форматирование жирным шрифтом, можно облегчить распознавание их среди прочих данных.

Чтобы проверить эту проблему, введите некоторые данные, схожие с данными на рис. 2.19.

	A	B
1	Quarter	Cost
2	Quart1	\$10.00
3	Quart1	\$20.00
4	Quart2	\$10.00
5	Quart2	\$10.00
6	Quart2	\$10.00
7	Quart3	\$15.00
8	Quart3	\$10.00
9	Quart3	\$25.00
10		

Рис. 2.19. Данные на рабочем листе перед добавлением промежуточных итогов

Теперь добавьте промежуточные итоги, выбрав команду Данные ► Итоги (Data ► Subtotals) и приняв в диалоговом окне Промежуточные итоги (Subtotals) значения параметров по умолчанию (просто щелкните кнопку **OK**).

На рис. 2.20 заголовки промежуточных итогов выделены жирным шрифтом, а соответствующие значения — нет. Так как в этой таблице только два столбца, ее нетрудно читать и выделять промежуточные итоги.

	A	B
1	Quarter	Cost
1	Quart1	\$10.00
1	Quart1	\$20.00
4	Quart1 Total	\$30.00
5	Quart2	\$10.00
6	Quart2	\$10.00
7	Quart2	\$10.00
7	Quart2 Total	\$30.00
9	Quart3	\$15.00
9	Quart3	\$10.00
11	Quart3	\$25.00
11	Quart3 Total	\$50.00
13	Grand Total	\$110.00

Рис. 2.20. Данные на рабочем листе после добавления промежуточных итогов

Однако чем больше столбцов в таблице, тем труднее визуально выделять промежуточные итоги. Эту проблему можно решить при помощи условного форматирования Excel. Используя в качестве примера рис. 2.19, попробуйте перед добавлением промежуточных итогов сделать следующее. Выделите ячейки A1:B9, следя за тем, чтобы ячейка A1 была активной. Выберите команду **Формат** ► **Условное форматирование** (Format ► Conditional Formatting), выберите пункт **Формула** (Formula Is) и введите следующую формулу:

```
=RIGHT($A1,5)="Total"
```

В русской версии Excel

```
=ПРАВСИМВ($A1,5)="Total"
```

Теперь щелкните кнопку **Формат** (Format), перейдите на вкладку **Шрифт** (Font) и в поле **Начертание** (Font Style) выберите **Полужирный** (Bold). Щелкните кнопку **ОК**, затем еще раз щелкните кнопку **ОК**.

Важным свойством формулы является использование абсолютной ссылки на столбец (\$A) и относительной ссылки на строку (1). Так как выделение вы начали с ячейки A1, Excel автоматически изменит формулу для каждой ячейки. Например, в ячейках A2 и B2 формула условного форматирования будет выглядеть как `=RIGHT($A2,5)=«Total»` (`=ПРАВСИМВ($A2,5)=«Total»`), а в ячейках A3 и B3 - как `=RIGHT($A3,5)=«Total»` (`=ПРАВСИМВ($A3,5)=«Total»`).

Добавьте промежуточные итоги; результат будет выглядеть, как на рис. 2.21.

	A	B	C
1	Quarter	Cost	
2	Quart1	\$10.00	
3	Quart1	\$20.00	
4	Quart1 Total	\$30.00	
5	Quart2	\$10.00	
6	Quart2	\$10.00	
7	Quart2	\$10.00	
8	Quart2 Total	\$30.00	
9	Quart3	\$15.00	
10	Quart3	\$10.00	
11	Quart3	\$25.00	
12	Quart3 Total	\$50.00	
13	Grand Total	\$110.00	

Рис. 2.21. Данные на рабочем листе после форматирования промежуточных итогов

Необходимо помнить, что после удаления промежуточных итогов жирный шрифт также исчезнет.

Трюк в трюке

Единственный подвох в этом способе — то, что общий итог (Grand Total) выводится в том же стиле, что и промежуточные итоги. Было бы удобно отформатировать общий итог по-другому, чтобы он отличался от промежуточных итогов и был легко заметен. Мы сделаем это, используя тот же пример.

Начните с необработанных данных. Выделите ячейки A1:B9, следя за тем, чтобы ячейка A1 была активной при выделении. Выберите команду **Формат > Условное форматирование (Format > Conditional Formatting)**. Выберите пункт **Формула (Formula Is)** и введите следующую формулу:

```
=A1="Grand Total"
```

Щелкните кнопку **Формат (Format)**, перейдите на вкладку **Шрифт (Font)** и в поле **Начертание (Font Style)** выберите **Полужирный (Bold)**. Щелкните кнопку **ОК**, а затем кнопку **Добавить (Add)**, чтобы добавить второе условие форматирования. Выберите пункт **Формула (Formula Is)** и введите следующую формулу:

```
=RIGHT($A1,5)="Total"
```

В русской версии Excel

```
=ПРАВСИМВ($A1,5)="Total"
```

Щелкните кнопку **Формат (Format)**, перейдите на вкладку **Шрифт (Font)** и в поле **Начертание (Font Style)** выберите **Полужирный курсив (Bold Italic)**. В поле **Подчеркивание (Underline)** выберите **Одинарное, по значению (Single)**, щелкните кнопку **ОК**, а затем еще раз щелкните кнопку **ОК**.

Теперь выберите команду **Данные > Итоги (Data > Subtotals)** и щелкните кнопку **ОК**, приняв значения по умолчанию. Теперь данные на вашем рабочем листе будут выглядеть, как на рис. 2.22.

	Quarter	Cost
2	Quart1	\$10.00
3	Quart1	\$20.00
4	Quart1 Total	<u>\$30.00</u>
5	Quart2	\$10.00
6	Quart2	\$10.00
7	Quart2	\$10.00
8	Quart2 Total	<u>\$30.00</u>
9	Quart3	\$15.00
10	Quart3	\$10.00
11	Quart3	\$25.00
12	Quart3 Total	<u>\$50.00</u>
13	GrandTotal	<u>\$110.00</u>
14		

Рис. 2.22. Данные на рабочем листе с более заметным общим итогом

Чтобы улучшить читаемость промежуточных итогов, можно использовать любое форматирование.

Т Р Ю К
№36

Преобразование формул и функций Excel в значения

Большинство таблиц Excel содержат формулы. Иногда нужно, чтобы в ячейку был помещен только результат вычисления формулы, а не сама формула, которая изменяется, когда изменяются данные, на которые она ссылается.

Это можно сделать вручную двумя способами или использовать макрос, который автоматически выполнит эту задачу. Давайте сначала изучим ручные методы.

Специальная вставка

При помощи инструмента Excel Специальная вставка (Paste Special) можно копировать результаты вычисления формулы, оставляя исходные формулы на месте. Предположим, у вас есть формулы в ячейках A1:A100. Выделите этот диапазон, выберите команду Копировать (Copy), а затем выделите начальную ячейку, куда будут помещены результаты. Выберите команду Правка ▸ Специальная вставка ▸ Значения (Edit ▸ Paste Special ▸ Values) и щелкните кнопку ОК.

Если вы хотите на место исходных формул поместить только результаты их вычисления, выделите диапазон с формулами и выберите команду Копировать (Copy). Теперь, когда диапазон все так же выделен, выберите команду Правка ▸ Специальная вставка ▸ Значения (Edit ▸ Paste Special ▸ Values) и щелкните кнопку ОК.

Копирование только значений

Скопировать результаты вычисления формул, оставив исходные формулы на месте, можно при помощи контекстного меню, о котором многие пользователи даже не подозревают.

Выделите диапазон ячеек с формулами и правой кнопкой мыши щелкните правую или левую границу выделения (другими словами, любое место, кроме маркера заполнения). Не отпуская правую кнопку мыши (или удерживая клавишу Ctrl и нажав кнопку мыши на Macintosh), перетащите диапазон в желаемое место, отпустите кнопку мыши и в раскрывшемся контекстном меню выберите команду Копировать только значения (Copy Here as Values Only).

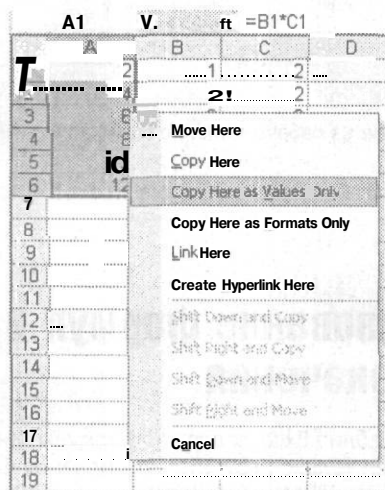


Рис. 2.23. Контекстное меню

Результаты вычисления формул можно также поместить вместо самих формул. Выделите диапазон ячеек с формулами, затем снова правой кнопкой мыши щелкните правую или левую границу выделения (другими словами, любое место, кроме маркера заполнения). Не отпуская правую кнопку мыши (или удерживая клавишу **Ctrl** и нажав кнопку мыши на Macintosh), перетащите диапазон на один столбец вправо или влево, а затем обратно на исходное место, отпустите кнопку мыши и в раскрывшемся контекстном меню выберите команду Копировать только значения (Copy Here as Values Only) (рис. 2.23).

Макрос

Если вы часто преобразовываете ячейки, содержащие формулы и функции, в значения, можете использовать этот простой макрос, код которого приведен в листинге 2.4.

Листинг 2.4

```
Sub ValuesOnly()  
Dim rRange As Range  
On Error Resume Next  
Set rRange = Application.InputBox(Prompt:="Выберите формулы",  
Title:="VALUES ONLY", Type:=8)  
If rRange Is Nothing Then Exit Sub  
rRange = rRange.Value  
End Sub
```

Чтобы применить его, выберите команду Сервис ▸ Макрос ▸ Редактор Visual Basic (Tools ▸ Macro ▸ Visual Basic Editor) (**Alt/Option+F11**). В VBE выберите команду Insert ▸ Module, чтобы вставить стандартный модуль. В этот модуль введите предыдущий код. Щелкните кнопку Close этого окна или нажмите сочетание клавиш **Alt/Apple+Q**, чтобы вернуться обратно в Excel. Выберите команду Сервис ▸ Макрос ▸ Макросы (Tools ▸ Macro ▸ Macros) (**Alt/Option+F8**), выберите ValuesOnly, а затем щелкните кнопку Параметры (Options), чтобы назначить макросу сочетание клавиш. Запустив этот макрос, вы увидите окно InputBox, в котором необходимо выбрать диапазон, содержащий формулы. Адрес выбранного диапазона автоматически появится в этом окне, и все, что останется сделать, — щелкнуть кнопку OK.



Автоматическое добавление данных в список проверки

Возможность проверки в Excel — это замечательная вещь, но она не умеет выполнять одно важное действие (без следующего трюка): автоматически добавлять новую запись в список, который используется в качестве источника для списка проверки.

Если вы уже работали с проверкой, то знаете, что это весьма искусная возможность. Вероятно, наиболее впечатляющим является ее умение добавлять в любую

ячейку список, в котором пользователь может выбирать значения. Было бы здорово, если бы при вводе нового имени в проверенную ячейку Excel автоматически добавлял ее в список. Это возможно благодаря следующему трюку.

Пусть есть список имен в диапазоне A1:A10 (рис. 2.24).

	A	B
1	Robyn Fenlon	
2	Joe Smith	
3	Bill Bloggs	
4	Fred Stone	
5	Neil Watts	
6	Judith Thurley	
7	Jacqui Jones	
8	Connie Eldon	
9	Kim Holmes	
10	Andrea Brooks	
11		

Рис. 2.24. Рабочая книга, настроенная для списка проверки

Эти имена представляют сотрудников компании. Не так уже редко в подобный список добавляются имена новых сотрудников, но сейчас единственный способ сделать это — добавить новые имена в конце списка, а затем выбрать их в списке в проверенной ячейке.

Чтобы обойти это ограничение, сделайте следующее. В ячейке A11 введите следующую формулу и скопируйте ее вниз до строки 20 (рис. 2.25) — обратите внимание на относительную ссылку на ячейку A10:

=IF(OR(\$D\$1="",COUNTIF(\$A\$1:A10,\$D\$1)), "x", \$D\$1)

В русской версии Excel

=ЕСЛИ(ИЛИ(\$D\$1="",СЧЁТЕСЛИ(\$A\$1:A10,\$D\$1)), "x", \$D\$1)

	A	B
1	Robyn Fenlon	j
2	Joe Smith	
3	Bill Bloggs	
4	Fred Stone	
5	Neil Watts	
6	Judith Thurley	
7	Jacqui Jones	
8	Connie Eldon	
9	Kim Holmes	
10	Andrea Brooks	
11	x	
12	x	
13	x	
14	x	
15	x	
16	x	
17	x	
18	x	
19	X	!
20	x	

Рис. 2.25. Список с формулой, добавленной в строки A11:A20

Теперь выберите команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define) и в поле Имя (Names in workbook) введите MyNames. В поле Формула (Refers to) введите следующую формулу (рис. 2.26), щелкните кнопку Добавить (Add), затем щелкните кнопку ОК:

```
=OFFSET(Sheet1!$A$1,0,0,COUNTIF(Sheet1!$A:$A,"<>x"),1)
```

В русской версии Excel:

```
=СМЕЩ(Sheet1!$A$1,0,0,СЧЁТЕСЛИ(Sheet1!$A:$A,"<>x"),1)
```

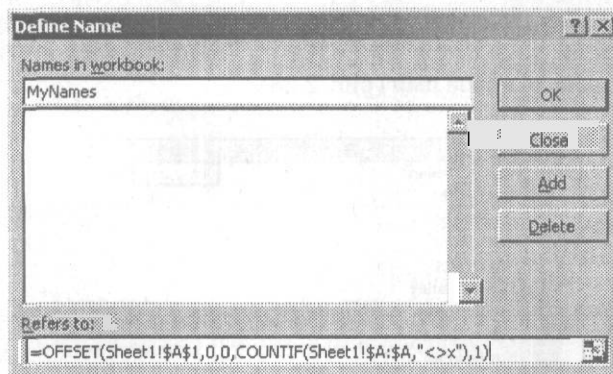


Рис. 2.26. Превращение списка в динамический список

Выделите ячейку D1 и выберите команду Данные ► Проверка (Data ► Validation). В поле Тип данных (Allow) выберите пункт Список (List), а в поле Источник (Source) введите =MyNames; удостоверьтесь, что установлен флажок Список допустимых значений (In-Cell). Перейдите на вкладку Сообщение об ошибке (Error Alert) и сбросьте флажок Выводить сообщение об ошибке (Show error alert after invalid data is entered). Щелкните кнопку ОК. Вы увидите результат, как на рис. 2.27.

	A	B	C	D	E
1	Robyn Fenlon				
2	Joe Smith				
3	Bill Bloggs				
4	Fred Stone				
5	Neil Watts				
6	Judith Thurley				
7	Jacqui Jones				
8	Connie Eldon				
9	Kim Holmes				
10	Andrea Brooks				
11	x				
12	x				
13	x				

Рис. 2.27. Список с проверкой, добавленный в ячейку D1

Правой кнопкой мыши щелкните ярлычок листа и в контекстном меню выберите команду Исходный текст (View Code). Введите код, приведенный в листинге 2.5.

Листинг 2.5

```
Private Sub Worksheet_Calculate()
On Error Resume Next
    Application.EnableEvents = False
    Range("MyNames") = Range("MyNames").Value
    Application.EnableEvents = True
On Error GoTo 0
End Sub
```

Чтобы вернуться обратно в Excel и сохранить рабочую книгу, закройте окно. Теперь выделите ячейку D1, введите любое имя, не входящее в список, и нажмите клавишу Enter. Снова выделите ячейку D1 и взгляните на список. Теперь в списке должно присутствовать новое имя (рис. 2.28).

	A	B	C	D	E
1	Robyn Fenlon			Jim Jones	
2	Joe Smith				
3	Bjil Bloggs				
4	Fred Stone				
5	Neil Watts				
6	Judith Thurley				
7	Jacqui Jones				
8	Connie Eldon				
9	Kim Holmes				
10	Andrea Brooks				
11	Jim Jones				
12	x				
13	x				

Рис. 2.28. Список после добавления новой записи в ячейку D1

Если вы хотите добавить больше 10 новых имен в список, скопируйте формулу ниже строки 20.

ТРЮК
№ 38

Трюки с возможностями даты и времени в Excel

Возможности даты и времени Excel при создании простых таблиц работают прекрасно, но могут вызывать проблемы в более сложных проектах. К счастью, когда допущения Excel не отвечают вашим требованиям, всегда есть способ обойти их.

По умолчанию в Excel используется система дат 1900. Это означает, что числовое значение, лежащее в основе даты 1 января 1900 года, равно 1, у 2 января 1900 года — 2 и так далее. В Excel эти значения называются *последовательными значениями*, позволяющими использовать даты в вычислениях.

Формат времени очень похож, но Excel обрабатывает время как десятичные доли, где 1 — это 24:00, или 00:00. Числовое значение 18:00 (в русской версии) равно 0,75, так как это три четверти от 24 часов.

Чтобы узнать числовое значение даты или времени, отформатируйте ячейку, содержащую значение, форматом Общий (General). Например, у даты и времени (в английской версии) 3/July/2002 3:00:00 PM числовое значение 37440.625, где число после десятичной точки представляет время, а 37440 — последовательное значение для 3 июля 2002 года.

Сложение за пределами 24 часов

При помощи функции СУММ (SUM) или просто знака плюс время можно складывать. Таким образом, `=SUM(A1:A5)` даст нам общее количество часов в диапазоне A1:A5, если эти ячейки содержат допустимые значения времени. Однако если Excel не дано никаких специальных указаний, он не складывает время за пределами 24 часов. Это происходит потому, что, когда значение времени превышает 24 часа (настоящее значение равно 1), оно переходит в новый день и отсчет начинается заново. Чтобы заставить Excel не переходить в новый день после каждых 24 часов, можно использовать формат ячеек 37:30:55 или пользовательский формат [ч]:мм:сс ([h]:mm:ss).

Схожий формат можно применять для получения общего количества минут или секунд. Чтобы узнать полное количество минут, когда время равно 24:00, отформатируйте ячейку как [m] ([m]), и вы получите 1440. Чтобы получить общее количество секунд, используйте пользовательский формат [c] ([s]) и вы получите 86400.

Вычисление времени и даты

Если вы хотите использовать фактические значения времени в других вычислениях, помните о следующих «магических» числах: 60 — 60 минут или 60 секунд; 3600 - 60 секунд * 60 минут; 24 - 24 часа; 1440 - 60 минут * 24 часа; 86400 - 24 часа * 60 минут * 60 секунд.

Вооружившись этими магическими числами и предыдущей информацией, намного проще манипулировать временем и датами. Взглянем на следующие примеры и посмотрим, что они обозначают (предполагаем, что время записано в ячейке A1).

Если у вас есть число 5.50 и вы хотите получить 5:30 или 5:30 a.m., используйте следующую формулу:

`-A1/24`

Необходимо указать подходящий формат!

Если время должно выглядеть как 17:30 или 5:30 p.m., используйте следующую формулу:

`=(A1/24)+0.5`

Чтобы получить противоположное значение, то есть десятичное время из настоящего времени, воспользуйтесь формулой

`=A1*24`

Если в ячейке содержится настоящая дата и настоящее время (например, 22/Jan/03 15:36), а вы хотите получить только дату, используйте следующую формулу:

`-INT(A1)`

В русской версии Excel

`=ЦЕЛОЕ(A1)`

Чтобы получить только время

`-A1-INT(A1)`

В русской версии Excel

`=A1-ЦЕЛОЕ(A1)`

или

`=MOD(A1,1)`

В русской версии Excel

`=ОСТАТ(A1,1)`

И вновь необходим подходящий формат.

Чтобы найти разность между двумя датами, воспользуйтесь формулой

`=DATEDIF(A1,A2,"d")`

где A1 — это более ранняя дата.

Получим количество дней между двумя датами. В качестве результата можно также указать «т» или «у», то есть месяцы или годы. (В действительности функция DATEDIF в Excel 97 не документирована и является функцией Lotus 123.)

Если более ранняя дата или время неизвестны, помогут функции МИН (MIN) и МАКС (MAX). Например, чтобы наверняка получить правильный результат, можно воспользоваться такой функцией:

`=DATEDIF(MIN(A1,A2),MAX(A1,A2),"d")`

В русской версии Excel:

`=DATEDIF(МИН(A1,A2),МАКС(A1,A2),"d")`

При работе со временем может также понадобиться учитывать начальное и конечное время. Например, начальное время — это 8:50 p.m. в ячейке A1, а конечное время — 9:50 a.m. в ячейке A2. Если вы вычтете начальное время из конечного (`=A2-A1`), получите в ответе #####, так как Excel по умолчанию не работает с отрицательными значениями времени. Подробнее о том, как работать с отрицательными значениями времени, — в разделе «Трюк № 74. Отображение отрицательных значений времени».

Иначе это ограничение можно обойти двумя способами, гарантировав положительный результат:

`=MAX(A1,A2)-MIN(A1:A2)`

В русской версии Excel

`=МАКС(A1,A2)-МИН(A1:A2)`

или

```
=A1-A2+IF(A1>A2,1)
```

В русской версии Excel

```
=A1-A2+ЕСЛИ(A1>A2,1)
```

Можно также приказать Excel прибавить любое количество дней, месяцев или лет к любой дате

```
=DATE(YEAR(A1)+value1,MONTH(A1)+value2,DAY(A1)+value3)
```

В русской версии Excel

```
=ДАТА(ГОД(A1)+value1,МЕСЯЦ(A1)+value2,ДЕНЬ(A1)+value3)
```

Чтобы добавить один месяц к дате в ячейке A1, воспользуйтесь формулой

```
=DATE(YEAR(A1),MONTH(A1)+1,DAY(A1))
```

В русской версии Excel

```
=ДАТА(ГОД(A1),МЕСЯЦ(A1)+1,ДЕНЬ(A1))
```

В Excel реализовано и несколько дополнительных функций, являющихся частью надстройки Analysis ToolPak. Выберите команду Сервис ► Надстройки (Tools ► Add-Ins) и установите флажок Пакетанализа (Analysis ToolPak) и, если появится сообщение с **вопросом**, нужно ли установить эту надстройку, ответьте **согласием**. Станут доступны дополнительные функции, такие, как **ДАТАМЕС** (EDATE), **КОНЕЦМЕСЯЦА** (EOMONTH), **ЧИСТРАБДНИ** (NETWORKDAYS) и WEEKNUM.

Все эти функции можно найти в категории Дата и время (Date & Time) диалогового окна мастера функций. Их легко применять, сложнее узнать, что эти функции существуют, и привлечь их к делу.

Настоящие даты и время

Иногда в таблицах с импортированными данными (или данными, введенными неправильно) даты и время отображаются как текст, а не как настоящие числа. Это можно легко распознать, немного расширив столбцы, выделив столбец, выбрав команду Формат ► Ячейки ► Выравнивание (Format ► Cells ► Alignment) и для параметра По горизонтали (Horizontal) выбрав значение По значению (General) (это формат ячеек по умолчанию). Щелкните кнопку ОК и внимательно просмотрите даты и время. Если какие-либо значения *не* выровнены по правому краю, Excel не считает их датами.

Чтобы исправить эту ошибку, сначала скопируйте любую пустую ячейку, затем выделите столбец и отформатируйте его, выбрав любой формат даты или времени. Не снимая выделение столбца, выберите команду Правка ► Специальная вставка ► Значение ► ОК (Edit ► Paste Special ► Value ► Add). Теперь Excel будет преобразовывать все текстовые даты и время в настоящие даты и время. Возможно, вам придется еще раз изменить форматирование. Еще один простой способ — ссылаться на ячейки так: =A1+0 или =A1*1.

Ошибка даты?

Excel ошибочно предполагает, что 1900 год был високосным годом¹. Это означает, что внутренняя система дат Excel считает, что существовал день 29 февраля 1900 года, тогда как его не было! Самое невероятное — Microsoft сделала это намеренно, по крайней мере, они так утверждают. Более подробную информацию вы найдете на сайте <http://support.microsoft.com/default.aspx?scid=kb;EN-US;q181370>.

Вот несколько хороших ссылок на информацию о дате и времени:

КАК СДЕЛАТЬ: даты и время в Excel 2000

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q214094#6>;

текст или число, преобразованное в непреднамеренный числовой формат:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q214233>;

максимальное значение времени в Microsoft Excel

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q214386>;

Даты и время, выводимые при просмотре формул как последовательные числа:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q241072>;

настройки в диалоговом окне Формат ячеек (Format Cells):

<http://support.microsoft.com/default.aspx?scid=kb;en-us;Q264372>.

Как использовать даты и время в Microsoft Excel:

<http://support.microsoft.com/default.aspx?scid=kb;en-us;214094>.

Даты и время, вероятно, представляют собой одну из наиболее запутанных сфер Excel. Вооружившись этой информацией, вы, вероятно, лучше поймете причуды Excel и упростите себе работу в этой области.

¹ Добавим, он был последним годом XIX века, а не первым XX. — *Примеч. ред.*

Трюки с именованием

Трюки с 39 по 44

Вероятно, самое большое преимущество использования именованных диапазонов заключается в том, что формулы становятся намного проще читать и понимать, причем не только вам, но и другим пользователям, которым приходится работать с вашими электронными таблицами. Используя именованные диапазоны (несомненно, одну из самых полезных возможностей Excel), можно сослаться на диапазон ячеек и назначить ему определенное имя. После этого можно сослаться на этот диапазон по имени, а не по адресу. Хотя именованные диапазоны и представляют собой мощный инструмент, можно выйти за границы стандартных имен диапазонов несколькими способами.

**ТРЮК
№ 39**

Адресация данных по имени

Хотя номера ячеек лежат в основе всего, что делает Excel, намного проще запоминать имена, например, *Item Number* или *Quantity*, чем номера ячеек, например, *A1:A100*. В Excel это сделать просто.

В Excel используется одна и та же техника определения именованных ячеек и именованных диапазонов — поле имени в левой части строки формул. Чтобы присвоить имя ячейке, выделите ее, введите нужное имя в поле имени (рис. 3.1) и нажмите клавишу **Enter**. Чтобы присвоить имя диапазону ячеек, выделите диапазон, введите нужное имя для диапазона в поле имени и нажмите клавишу **Enter**.

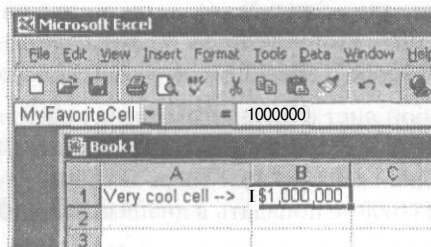


Рис. 3.1. Присваивание ячейке имени MyFavoriteCell

Раскрывающийся список справа в поле имени позволяет находить именованные диапазоны и ячейки (дополнительные способы поиска диапазонов вы найдете

в разделе «Трюк № 44. Поиск именованных диапазонов на рабочем листе» в конце этой главы). Если точно выделить диапазон, вместо обычных ссылок на ячейки в поле имени появится его имя.

В формулах эти имена можно применять вместо идентификаторов ячеек или диапазонов. Если вы назовете ячейку E4 «date», то сможете использовать формулу =date вместо =E4. Схожим образом если вы создадите диапазон с названием «quantity» в ячейках A3:A10, то сможете написать формулу =СУММ(quantity) (=SUM(quantity)) вместо =СУММ(A3:A10) (=SUM(A3:A10)).

По мере того как электронные таблицы становятся больше и сложнее, именованные ячейки и диапазоны становятся жизненно необходимым инструментом для поддержания понятности и управляемости таблиц.

ТРЮК № 40

Использование одного имени для диапазонов на разных рабочих листах

Иногда бывает удобно использовать одно и то же имя для данных на одном и том же месте на разных рабочих листах одной рабочей книги. Для этого в Excel нужно сделать несколько дополнительных шагов.

Обычно, когда вы именуete диапазон, это имя хранится на уровне рабочей книги, то есть одно имя всегда относится к указанному диапазону на указанном листе, где бы оно ни упоминалось в рабочей книге. Если какое-то имя уже использовано, его нельзя применить для именованния диапазона на другом рабочем листе. Однако это ограничение можно обойти.

Предположим, у вас есть рабочая книга с тремя рабочими листами. У этих трех листов простые имена: Sheet1, Sheet2 и Sheet3. Вы хотите создать именованный диапазон с именем MyRange (можно использовать любое допустимое имя), который будет относиться к диапазону A1:A10 на листе Sheet1, когда активен лист Sheet1, A1:A10 на листе Sheet2, когда активен лист Sheet2, и A1:A10 на листе Sheet3, когда активен лист Sheet3.

Для этого активируйте лист Sheet1, выделите диапазон A1:A10 и щелкните в поле имени, как описано в разделе «Трюк № 39. Адресация данных по имени». Введите Sheet1!MyRange и нажмите клавишу Enter. То же самое сделайте на листах Sheet2 и Sheet3, введя Sheet2!MyRange и Sheet3!MyRange соответственно.

Теперь активируйте любой лист и щелкните стрелку раскрывающего списка поля имени. Вы должны увидеть имя MyRange только один раз. Выберите его и вы попадете непосредственно в диапазон A1:A10. Теперь активируйте другой лист и сделайте то же самое. Вы всегда будете попадать в диапазон A1:A10 на активном листе.

Этот трюк работает благодаря тому, что перед именем диапазона вы поставили имя листа и восклицательный знак. Выбрав команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define), вы увидите только одно имя — имя, относящееся к активному в данный момент листу.

Если в имени рабочего листа есть пробелы, нельзя просто сослаться на диапазон A1:A10 на листе Sheet1 как Sheet1!MyRange. Нужно назвать его 'Sheet1!MyRange, поместив слово Sheet1 в одиночные кавычки, апострофы. В действительности, апострофы можно использовать и с именем рабочего листа, не содержащим пробелы, поэтому лучше вставлять их при каждой ссылке на рабочие листы.

В именованных диапазонах можно использовать и относительные ссылки. По умолчанию именованные диапазоны — абсолютные, но вы можете исправить это. Сделайте следующее.

Выделите ячейку A11 на любом рабочем листе и выберите команду Вставка ► Имя ► Присвоить (Name ► Insert ► Define). В поле Имя (Names in workbook) введите MyNumbers. В поле Формула (Refers to) введите =A\$1:A\$10, щелкните на кнопке Добавить (Add), затем щелкните на кнопке ОК.

Теперь в ячейку A1 введите число 1. Выделите ячейку A1, подведите указатель к маркеру заполнения и нажмите левую кнопку мыши. Удерживая клавишу Ctrl, перетащите маркер к ячейке A10. Так вы создадите список с арифметической прогрессией с шагом 1.

В ячейку B1 введите 1 и перетащите маркер на ячейку B10, в этот раз не удерживая клавишу Ctrl. В ячейке A11 введите следующую формулу:

=SUM(MyNumbers)

В русской версии Excel

=СУММ(MyNumbers)

В ячейке B11 введите следующую формулу:

=SUM(MyNumbers)

В русской версии Excel

=СУММ(MyNumbers)

Вы должны получить ответы 55 и 10 соответственно, так как ячейка A11 была активной, когда вы использовали команду Вставка ► Имя ► Присвоить (Name ► Insert ► Define) и присвоили имя диапазону A\$1:A\$10, то есть диапазону с относительной ссылкой на столбец и абсолютной ссылкой на строку.

ПРИМЕЧАНИЕ

Чтобы создать абсолютную ссылку на диапазон, воспользуйтесь знаком доллара (\$).

Используемое в формуле имя MyNumbers всегда обозначает 10 ячеек непосредственно над формулой. Если вы вставите формулу =СУММ(MyNumbers) (=SUM(MyNumbers)) в ячейку A11 на другом рабочем листе, она все так же будет обозначать ячейки A1:A10 на листе, который был активен при создании имени диапазона.

Предположим, вы хотите упростить суммирование значений в десяти ячейках. Выделите ячейку A11 на любом листе. Выберите команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define) и в поле Имя (Name) введите MySum. Затем в поле Формула (Refers to) введите следующую формулу:

=SUM(A\$1:A\$10)

В русской версии Excel

=СУММ(A\$1:A\$10)

Щелкните на кнопке **Добавить (Add)**, затем щелкните на кнопке **ОК**. Теперь в ячейке A1 введите число 1. Выделите ячейку A1, подведите указатель мыши к маркеру заполнения и нажмите левую кнопку мыши. Удерживая клавишу **Ctrl**, перетащите маркер к ячейке A10. После этого в ячейке B1 введите 1 и перетащите маркер к ячейке B10, не удерживая клавишу **Ctrl**.

В ячейке A11 введите следующую формулу:

=MySum

В ячейке B11 также введите формулу

=MySum

Вы получите такой же результат, что и в предыдущем примере, не используя функцию **СУММ (SUM)**. Сочетание абсолютных и относительных ссылок и вложение нескольких функций друг в друга может быть очень полезным и экономит много времени.

ТРЮК
№ 41

Создание пользовательских функций при помощи имен

Хотя ссылаться на данные по имени очень удобно, иногда бывает полезнее хранить под определенным именем постоянное значение или даже формулу, особенно, если вы создавали пользовательские функции в **VBA**.

Предположим, у вас есть налоговая ставка 10 %, которую вы используете в рабочей книге для различных вычислений. Вместо того чтобы вводить значение 10 % (0,1) во всех формулах, для которых необходима эта налоговая ставка, можно просто ввести слово **TaxRate**, и Excel автоматически подставит значение 0,1, которому присвоено это имя. Вот как можно это сделать.

Выберите команду **Вставка > Имя > Присвоить (Insert > Name > Define)** и в поле **Имя (Names in workbook)** введите **TaxRate**. В поле **Формула (Refers To)** введите =0,1 и щелкните на кнопке **Добавить (Add)**.

Теперь в любую ячейку можно ввести любую формулу, а вместо значения 10 % можно использовать слово **TaxRate**. Вероятно, одно из самых больших преимуществ этого метода заключается в том, что, когда налоговая ставка увеличивается или уменьшается и необходимо отразить в формулах новое значение, можно просто выбрать команду **Вставка > Имя > Присвоить (Insert > Name > Define)**, а затем выбрать имя **TaxRate** и изменить соответствующее значение.

Чтобы сделать очередной шаг в этой концепции, в качестве диапазона в поле **Формула (Refers To)** можно указывать формулы, а не адреса ячеек или постоянные значения. Предположим, вы хотите создать имя, которое, если ввести его в ячейку,

будет автоматически возвращать сумму значений в десяти ячейках непосредственно над этой ячейкой.

Выделите ячейку А1 на любом рабочем листе и выберите команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define). В поле Имя (Names in workbook) введите имя Total. В поле Формула (Refers to) введите следующую формулу:

```
=SUM(A1:A10)
```

В русской версии Excel

```
=СУММ(A1:A10)
```

Щелкните на кнопке Добавить (Add), затем щелкните на кнопке ОК.

В любой столбец, начиная со строки 1, введите 10 чисел. Теперь перейдите к строке И в том же столбце и введите формулу

```
=Total
```

Имя Total автоматически вернет сумму значений, которые вы только что ввели в 10 ячеек А1:А10.

Чтобы создать похожую именованную формулу, которая не ограничивается 10 ячейками, а использует все ячейки непосредственно над той строкой, где вы ввели =Total, сделайте следующее.

Выделите ячейку В11 и выберите команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define). Щелкните имя Total.

Посмотрите в поле Формула (Refers to): вы должны увидеть формулу =СУММ(В1:В10) (=SUM(B1:B10)). Так вы создаете именованные формулы. Другими словами, поскольку вы не указали абсолютные ссылки на столбцы для исходного имени Total, эта формула всегда ссылается на тот столбец, в котором находится.

Теперь щелкните поле Формула (Refers to) и измените формулу следующим образом:

```
=SUM(B$1:B10)
```

В русской версии Excel

```
=СУММ(B$1:B10)
```

Щелкните на кнопке Добавить (Add), затем щелкните на кнопке ОК.

Выберите любую строку в любом столбце, кроме строки 1, и введите =Total и вы автоматически получите сумму значений во всех ячейках выше данной, независимо от того, сколько там строк. Так происходит потому, что вы поставили якорь на строку 1, сделав ссылку на нее абсолютной, но оставили ссылку на ячейку В10 относительной, что означает, что она всегда будет указывать на ячейку непосредственно над строкой, где вы ввели именованную формулу =Total.

Комбинируя этот трюк с одной из стандартных, хотя и малоизвестных возможностей Excel — методом пересечения, — можно создавать сложные функции поиска. Если вы не знаете, как работает метод пересечения, вот небольшой пример.

В ячейку А1 введите заголовок Name, в ячейку В1 — Ray, в ячейку С1 — Title. В ячейку А2 введите Bill, в ячейку А3 — Fred.

В ячейку B2 введите 10, в ячейку B3 — 20, в ячейку C2 — Mr, в ячейку C3 — Dr. Теперь выделите диапазон A1:C3 и выберите команду Вставка ► Имя ► Создать (Insert ► Name ► Create). Удостоверьтесь, что установлены флажки для строки выше и столбца слева, и щелкните на кнопке ОК.

Выделите любую ячейку за пределами таблицы и введите =Fred Title. Вы должны увидеть правильное обращение для имени Fred.

ПРИМЕЧАНИЕ

Важно, чтобы между словами Fred и Title был пробел, так как именно его Excel понимает как оператор пересечения.

Основываясь на таком принципе, можно сочетать эту возможность с возможностями именованных формул Excel, чтобы делать электронные таблицы не только простыми в работе, но и легко читаемыми и удобными для восприятия.

Предположим, в вашей электронной таблице есть таблица, настроенная, как на рис. 3.2, и на основе этой таблицы вы создаете имена в Excel.

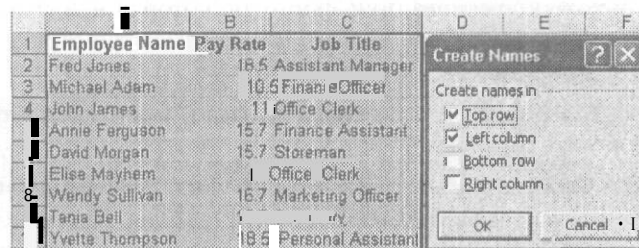


Рис. 3.2. Таблица и диалоговое окно создания имен

Создав имена для этой таблицы, вы увидите, что Excel автоматически вставляет символ подчеркивания вместо пробелов между двумя или более словами. Это происходит потому, что имена именованных диапазонов не могут содержать пробелы.

Выберите команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define) и в поле Имя (Names in workbook) введите FredsPayRate. В поле Формула (Refers to) введите =Fred_Jones Pay_Rate и щелкните на кнопке Добавить (Add) (рис. 3.3).

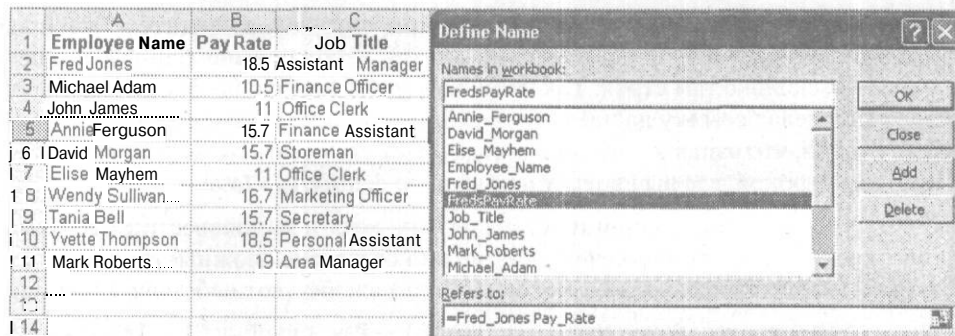


Рис. 3.3. Функция поиска по имени

Теперь в любой ячейке за пределами таблицы введите формулу

```
=Fred'sPayRate
```

Будет автоматически возвращена ставка заработной платы для сотрудника Fred.

Поэкспериментируйте с пересечениями, чтобы узнать, как они наилучшим образом будут работать в ваших проектах.

ТРЮК
№42

Создание увеличивающихся и уменьшающихся диапазонов

Если вам необходимо постоянно обновлять и добавлять данные или если вы работаете с диаграммами и сводными таблицами, вам понадобятся динамические именованные диапазоны, которые увеличиваются и уменьшаются в зависимости от ваших данных.

Чтобы понять, как работают динамические именованные диапазоны, сначала нужно познакомиться с функцией Excel **СМЕЩ**(OFFSET), если вы еще не знакомы с ней. Функция **СМЕЩ**(OFFSET) относится к ссылочным функциям и функциям поиска Excel.

Начнем с простейшего динамического именованного диапазона, который будет расширяться вниз по одному столбцу, но только до тех пор, пока в этом столбце есть записи. Например, если в столбце A есть 10 последовательных строк с данными, динамический именованный диапазон будет охватывать диапазон A1:A10. Чтобы создать базовый динамический именованный диапазон, сделайте следующее.

Выберите команду Вставка ▶ Имя ▶ Присвоить (Insert ▶ Name ▶ Define) и в поле Имя (Names in workbook) введите MyRange. В поле Формула (Refers to) введите следующую формулу:

```
=OFFSET($A$1,0,0,COUNTA($A$1:$A$100),1)
```

В русской версии Excel

```
=СМЕЩ($A$1,0,0,СЧЁТЗ($A$1:$A$100),1)
```

Теперь щелкните на кнопке Добавить (Add), затем щелкните на кнопке ОК.

ПРИМЕЧАНИЕ

При определении диапазона для функции COUNTA (СЧЁТЗ) не следует включать столбец данных полностью, чтобы функции не пришлось при вычислении учитывать, возможно, тысячи ненужных ячеек.

Теперь, если вы добавите некоторые данные в столбец A, этот диапазон будет содержать все данные в соседних строках, начиная с ячейки A1. Если вы хотите проверить динамический именованный диапазон, это можно сделать несколькими способами.

К сожалению, динамические именованные диапазоны нельзя проверить в стандартном поле имени слева от строки формул. Несмотря на это, можно щелкнуть

в поле имени, ввести имя **MyRange** и нажать клавишу Enter. Excel автоматически выделит диапазон. Конечно, можно воспользоваться и диалоговым окном **Переход (Go To)**, выбрав команду **Правка ▶ Перейти (Edit ▶ Go To)** (сочетание клавиш **Ctrl/Apple+G**). В поле **Ссылка (Reference)** введите **MyRange** и щелкните на кнопке **ОК**.

В динамическом именованном диапазоне, который вы создали в предыдущем примере, функция **СЧЁТЗ (COUNTA)** стоит на месте аргумента **Высота (Height)** функции **СМЕЩ (OFFSET)**.

ВНИМАНИЕ

Помните, что функция **СЧЁТЗ (COUNTA)** подсчитывает все непустые ячейки: будут учитываться и формулы в ячейках, возвращающих пустой текст (" ").

Если у вас есть список, который содержит только числовые данные, а в конце списка вы хотите записать текстовое значение, но не помещать его в динамический именованный диапазон, можно заменить функцию **СЧЁТЗ (COUNTA)** стандартной функцией Excel **СЧЁТ (COUNT)**. Она подсчитывает только ячейки, содержащие числовые данные.

В следующем примере динамический именованный диапазон мы применим для определения таблицы данных, которая должна быть динамической. Для этого в поле **Формула (Refers to)** введите следующую формулу:

```
=OFFSET($A$1,0,0,COUNTA($A$1:$A$100),COUNTA($1:$1))
```

В русской версии Excel

```
=СМЕЩ($A$1,0,0,СЧЁТЗ($A$1:$A$100),СЧЁТЗ($1:$1))
```

Теперь динамический именованный диапазон будет расширяться на столько записей, сколько есть в столбце A, и на столько строк, сколько заголовков в строке 1. Если вы уверены, что количество столбцов в таблице данных меняться не будет, можете заменить вторую функцию **СЧЁТЗ (COUNTA)** постоянным числом, например, 10.

Единственная проблема при использовании динамического именованного диапазона для таблицы данных заключается в том, что предполагается, что столбец A определяет максимальную длину таблицы. Чаще всего это так, однако иногда самым длинным столбцом может быть другой столбец таблицы.

Преодолеть эту проблему можно при помощи функции Excel **МАКС (MAX)**, которая возвращает самое большое число в диапазоне ячеек. В качестве примера создайте таблицу, как на рис. 3.4.

В строке 1 хранятся функции **СЧЁТЗ (COUNTA)**, которые ссылаются вниз на соответствующий столбец и, таким образом, возвращают количество записей в каждом столбце. Функция **МАКС (MAX)** будет использоваться в качестве аргумента **Высота (Height)** функции **СМЕЩ (OFFSET)**. Это гарантирует, что **динамический** именованный диапазон для этой таблицы всегда будет расширяться вниз на столько ячеек, сколько их содержится в самом длинном столбце в таблице. Конечно же, можно скрыть строку 1, так как пользователю совершенно не нужно ее видеть.

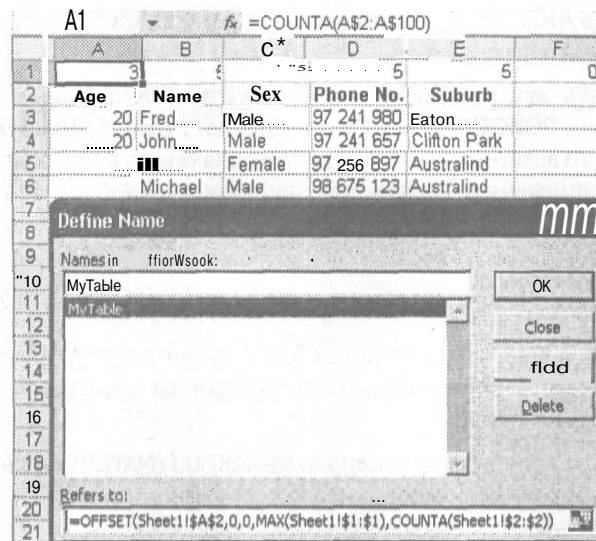


Рис. 3.4. Динамическая таблица данных и диалоговое окно присвоения имени

Во всех этих примерах мы предполагали, что данные всегда будут размещаться в соседних строках, между которыми не будет пустых ячеек. Хотя именно этот способ настройки списка или таблицы данных является правильным, иногда он становится неуправляемым.

В следующем примере список чисел в столбце А также содержит пустые ячейки. Это означает, что, если вы попытаетесь воспользоваться функцией СЧЁТ (COUNT) или СЧЁТЗ (COUNTA), динамический именованный диапазон закончится раньше, чем последняя ячейка с данными. Взгляните, например, на рис. 3.5.

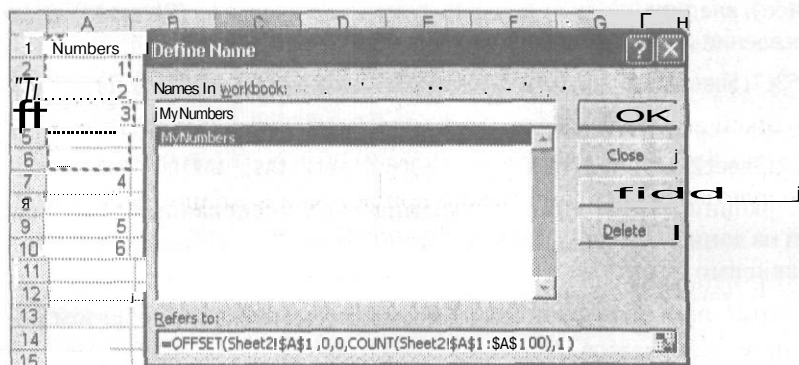


Рис. 3.5. Диапазон чисел и диалоговое окно присвоения имени

В этом случае, хотя последнее число диапазона в действительности находится в строке 10, динамический диапазон расширяется вниз только до шестой строки. Причина этого лежит в функции СЧЁТ (COUNT), которая считает значения

в ячейках от A1 до A100. Так как в списке только шесть числовых значений, диапазон содержит только шесть строк.

Чтобы преодолеть эту проблему, воспользуйтесь функцией Excel ПОИСКПОЗ (MATCH). Функция ПОИСКПОЗ (MATCH) возвращает относительную позицию элемента массива, соответствующего указанному значению в указанном порядке. Например, если вы примените следующую функцию ПОИСКПОЗ (MATCH):

```
=MATCH(6,$A$1:$A$100,0)
```

В русской версии Excel

```
=ПОИСКПОЗ(6,$A$1:$A$100,0)
```

к тому же набору чисел, что и на рис. 3.5, она вернет число 10, представляющее строку 10 столбца A. Она возвращает 10, так как вы приказали функции найти число 6 в диапазоне A1:A100.

Очевидно, когда вы используете функцию ПОИСКПОЗ (MATCH) в динамическом именованном диапазоне, последнее число диапазона, вероятно, заранее неизвестно. Таким образом, понадобится задать в функции поиск слишком большого числа, которое никогда не появится в диапазоне, и изменить ее последний аргумент с 0 на 1.

В предыдущем примере вы приказали функции ПОИСКПОЗ (MATCH) найти в точности число 6, не больше и не меньше. Заменяв 0 на 1, вы заставите функцию искать самое большое значение, меньшее или равное указанному.

Для этого воспользуйтесь формулой

```
=MATCH(1E+306,$A$1:$A$100,1)
```

В русской версии Excel

```
=ПОИСКПОЗ(1E+306,$A$1:$A$100,1)
```

Чтобы создать динамический именованный диапазон, который будет расширяться до последней строки, содержащей число (независимо от наличия пустых ячеек до нее), введите следующую формулу в поле Формула (Refers to) диалогового окна Присвоение имени (Define Name) (рис. 3.6):

```
=OFFSET(Sheet2!$A$1,0,0,MATCH(1E+306,Sheet2!$A$1:$A$100,1),1)
```

В русской версии Excel

```
=СМЕЩ(Sheet2!$A$1,0,0,ПОИСКПОЗ(1E+306,Sheet2!$A$1:$A$100,1),1)
```

Следующий логический тип динамических именованных диапазонов, вытекающий из данного, — это диапазон, расширяющийся до последней текстовой записи независимо от пустых ячеек в списке или таблице.

Для этого измените функцию ПОИСКПОЗ (MATCH) таким образом:

```
MATCH("*",$A$1:$A$100,-1)
```

В русской версии Excel

```
ПОИСКПОЗ("*",$A$1:$A$100,-1)
```

Такая функция всегда возвращает номер строки, где содержится последняя текстовая запись в диапазоне \$A\$1:\$A\$100.

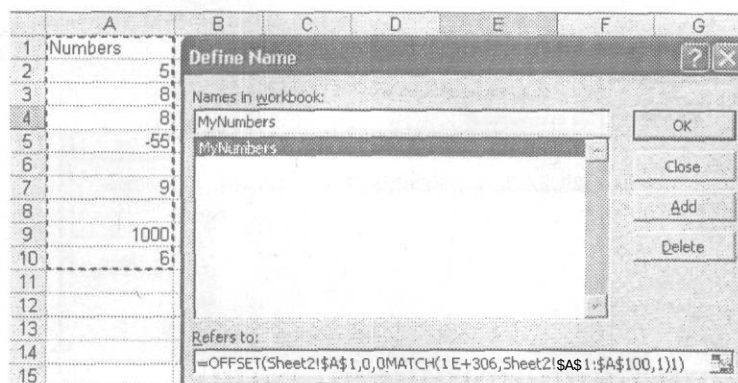


Рис. 3.6. Динамический диапазон, расширяющийся до последней записи, содержащей число

Теперь, когда известно, как выполнить эту задачу для числовых записей и текстовых значений, единственное, что осталось сделать, — каким-то образом определить динамический именованный диапазон, который будет расширяться, не обращая внимания на пустые ячейки в списке, содержащем и текстовые, и числовые данные.

Для этого сначала вставьте две пустые строки выше списка. Выделите строки 1 и 2 и выберите команду Вставка ▸ Строки (Insert ▸ Row). В первой строке (строка 1) введите следующую функцию:

```
=MAX(MATCH"*", $A$3:$A$100, -1), MATCH(1E+306, $A$3:$A$100, 1))
```

В русской версии Excel

```
=МАКС(ПОИСКПОЗ"*", $A$3:$A$100, -1), ПОИСКПОЗ(1E+306, $A$3:$A$100, 1))
```

В ячейке под ячейкой с формулой введите число 1. Ячейка еще ниже, под ячейкой с числом 1, должна содержать текстовый заголовок списка. Число 1 было добавлено, чтобы вторая функция ПОИСКПОЗ (MATCH) не вернула ошибку #N/A, если в диапазоне A3:A100 не окажется чисел. Первая функция ПОИСКПОЗ (MATCH) всегда найдет текст — в заголовке.

Дайте ячейке A1 имя MaxRow. Выберите команду Вставка > Имя ▸ Присвоить (Insert ▸ Name ▸ Define), присвойте динамическому диапазону имя, например, MyList и в поле Формула (Refers to:) (рис. 3.7) введите следующую формулу:

```
=OFFSET(Sheet2!$A$3,0,0,MaxRow,1)
```

В русской версии Excel

```
=СМЕЩ(Sheet2!$A$3,0,0,MaxRow,1)
```

В следующем списке перечислены типы динамических именованных диапазонов, которые могут оказаться полезными. Для *всех* этих примеров понадобится заполнить столбец A и текстом, и числовыми значениями. Кроме того, выберите команду Вставка ▸ Имя ▸ Присвоить (Insert ▸ Name ▸ Define) и в поле Имя (Names in workbook) введите имя, состоящее из одного слова (например, MyRange). Все, что будет изменяться, — это формула в поле Формула (Refers to).

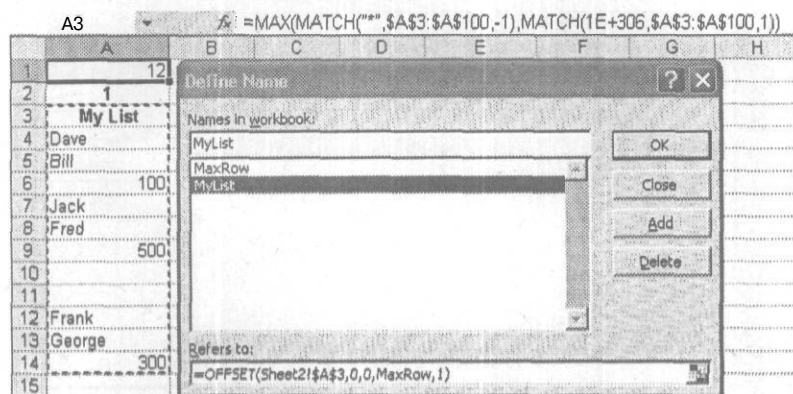


Рис. 3.7. Динамический список для числовых и текстовых записей, между которыми могут быть пустые ячейки

Расширить диапазон на столько строк, сколько существует числовых записей

В поле Формула (Refers to) введите следующее:

`=OFFSET(A1,0,0,COUNT($A:$A),1)`

В русской версии Excel

`=СМЕЩ(A1,0,0,СЧЁТ($A:$A),1)`

Расширить диапазон на столько строк, сколько существует числовых и текстовых записей

В поле Формула (Refers to) введите следующее:

`=OFFSET(A1,0,0,COUNTA($A:$A),1)`

В русской версии Excel

`=СМЕЩ(A1,0,0,СЧЁТ3($A:$A),1)`

Расширить до последней числовой записи

В поле Формула (Refers to) введите следующее:

`=OFFSET(A1,0,0,MATCH(1E+306,$A:$A))`

В русской версии Excel

`=СМЕЩ(A1,0,0,ПОИСКПОЗ(1E+306,$A:$A))`

Если вы ожидаете, что может встретиться число, большее 1E+306 (1 с 306 нулями), введите еще большее число.

Расширить до последней текстовой записи

В поле Формула (Refers to) введите следующее:

`=OFFSET(A1,0,0,MATCH("*",$A:$A,-1))`

В русской версии Excel

`=СМЕЩ(A1,0,0,ПОИСКПОЗ("*",$A:$A,-1))`

Расширить вниз в зависимости от значения в другой ячейке

В ячейку B1 введите число 10, а затем в поле Формула (Refers to) введите следующее:

```
=OFFSET($A$1,0,0,$B$1,1)
```

В русской версии Excel

```
=СМЕЩ($A$1,0,0,$B$1,1)
```

Теперь измените число в ячейке B1, и диапазон изменится соответствующим образом.

Расширять вниз по одной строке каждый месяц

В поле Формула (Refers to) введите следующее:

```
=OFFSET($A$1,0,0,MONTH(TODAY()),1)
```

В русской версии Excel

```
=СМЕЩ($A$1,0,0,МЕСЯЦ(СЕГОДНЯ()),1)
```

Расширять вниз по одной строке каждую неделю

В поле Формула (Refers to) введите следующее:

```
=OFFSET($A$1,0,0,WEEKNUM(TODAY()),1)
```

В русской версии Excel

```
=СМЕЩ($A$1,0,0,ДЕНЬНЕД(СЕГОДНЯ()),1)
```

Для последней формулы необходимо установить Пакет анализа (Analysis ToolPak). Это можно сделать командой Сервис ► Надстройки (Tools ► Add-ins).

ТРЮК № 43 Вложение динамических диапазонов для максимальной гибкости

Динамический именованный диапазон, который находится в другом динамическом именованном диапазоне, может оказаться полезным для таких случаев, когда, например, есть очень длинные списки имен.

Например, можно создать именованный диапазон с именем Jnames, который ссылается на все имена в отсортированном списке, начинающиеся с буквы J.

Начните со списка имен в столбце A, например, таких, как на рис. 3.8, где ячейка A1 — это заголовок, а список отсортирован. Выберите команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define). В поле Имя (Names in workbook) введите Names, а в поле Формула (Refers to) введите следующую формулу:

```
=OFFSET($A$2,0,0,COUNTA($A$2:$A$1000),1)
```

В русской версии Excel

```
=СМЕЩ($A$2,0,0,СЧЁТЗ($A$2:$A$1000),1)
```

Щелкните на кнопке Добавить (Add). Теперь еще раз щелкните поле Имя (Names in workbook) и введите Jnames (вместо J можно подставить любую другую букву).

В поле Формула (Refers to) введите следующее:

=OFFSET(INDIRECT(ADDRESS(MATCH("J*",Names,0)+1,1)),0,0,COUNTIF(Names,"J*"),1)

В русской версии Excel

=СМЕЩ(ДВССЫЛ(АДРЕС(ПОИСКПОЗ("J*",Names,0)+1,1)),0,0,СЧЁТЕСЛИ(Names,"J*"),1)

где «J*» — это критерий для данных, которые нужно найти, в данном случае — имен, начинающихся с J. Щелкните на кнопке Добавить (Add). Когда вы снова щелкнете поле Формула (Refers to), где находится формула, вокруг всех имен, начинающихся с буквы J, появится рамка выделения (см. рис. 3.8).

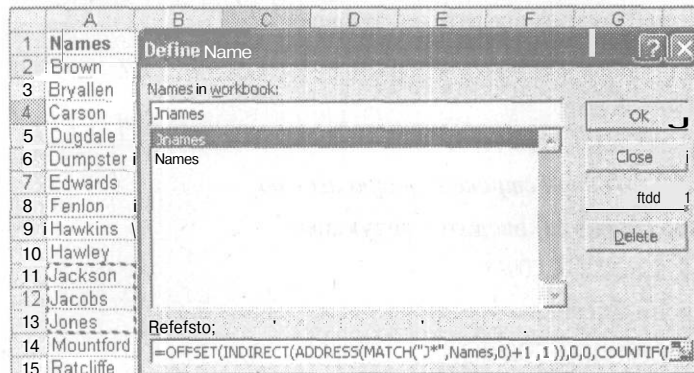


Рис. 3.8. Динамический именованный диапазон в пределах другого динамического именованного диапазона

При желании можно создать по одному именованному диапазону для каждой буквы алфавита, но, возможно, лучше, чтобы именованный диапазон изменялся в зависимости от того, какую букву вы вводите в ячейку на рабочем листе. Для этого просто введите любую букву в любую неиспользуемую ячейку и присвойте этой ячейке имя Letter.

Теперь выберите команду Данные ► Проверка (Data ► Validation) и в списке поля Тип данных (Allow) выберите пункт Список (List). Щелкните поле Источник (Source) и введите A*,B*,C* и т. д. для всех 26 букв алфавита (рис. 3.9). Закончив, щелкните на кнопке ОК.

Выберите команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define) и в поле Имя (Names in workbook) введите имя «Names». В поле Формула (Refers to) введите следующую формулу и щелкните на кнопке Добавить (Add):

=OFFSET(\$A\$2,0,0,COUNTA(\$A\$2:\$A\$1000),1)

В русской версии Excel

=СМЕЩ(\$A\$2,0,0,СЧЁТЗ(\$A\$2:\$A\$1000),1)

Снова щелкните поле Имя (Names in workbook) и введите LetterNames. В поле Формула (Refers to) введите следующую формулу и, закончив, щелкните на кнопке Добавить (Add) и на кнопке ОК:

=OFFSET(INDIRECT(ADDRESS(MATCH(Letter,Names,0)+1,1)),0,0,COUNTIF(Names,Letter),1)

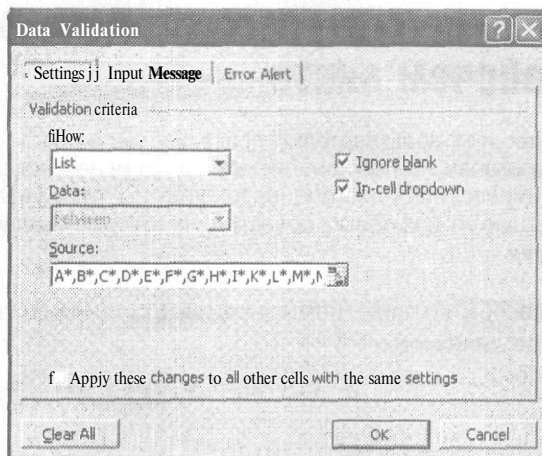


Рис. 3.9. Список проверки, состоящий из букв и символа шаблона

В русской версии Excel

=СМЕЩ(ДВССЫЛ(АДРЕС(ПОИСКПОЗ(Letter,Names,0)+1,1)),0,0, СЧЁТСЛИ(Names,Letter).1)

Результат будет выглядеть, как на рис. 3.10.

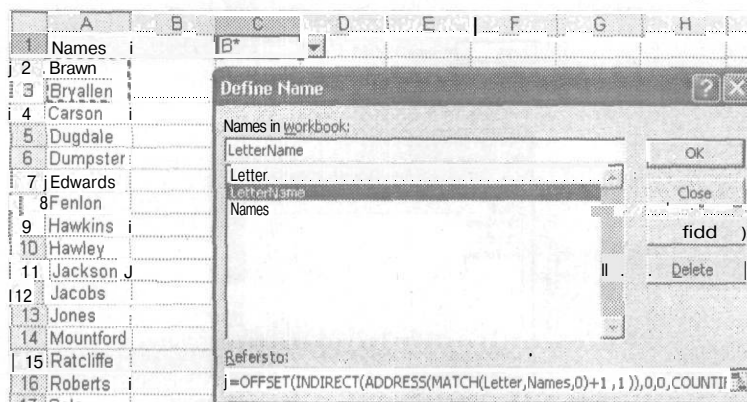


Рис. 3.10. Динамический именованный диапазон, управляемый содержимым другой ячейки

ПРИМЕЧАНИЕ

Формулы для динамических именованных диапазонов не нужно вводить заново с нуля. Вместо этого в диалоговом окне Присвоение имени (Define Name) щелкните существующий динамический именованный диапазон, измените имя в поле Имя (Names in workbook), перейдите в поле Формула (Refers to), измените формулу и щелкните на кнопке Добавить (Add). Исходный динамический именованный диапазон не будет заменен новым, вместо этого будет создан новый диапазон с другим именем, которое вы ему присвоите.

Для проверки введите любую букву в ячейку с именем Letter и вы увидите рамку выделения вокруг данных, начинающихся с этой буквы.

Т Р Ю К
№ 44

Поиск именованных диапазонов на рабочем листе

Excel позволяет пользователям присваивать понятные имена определенным диапазонам на рабочих листах. Со временем, когда количество различных именованных диапазонов на рабочем листе увеличивается, требуется инструмент для поиска областей, на которые ссылаются именованные диапазоны.

Вот два быстрых способа, которые помогут находить соответствующие диапазоны для любого имени диапазона.

Способ 1

Первый очень быстрый способ найти диапазоны — выбрать команду Вставка ► Имя ► Вставить (Insert ► Name ► Paste) или нажать клавишу F3. В диалоговом окне Вставка имени (Paste Name) щелкните на кнопке ОК (рис. 3.11)¹, и по строкам будут выведены все существующие имена (начиная с активной ячейки), причем в соседнем столбце появятся соответствующие диапазоны, на которые ссылаются имена.

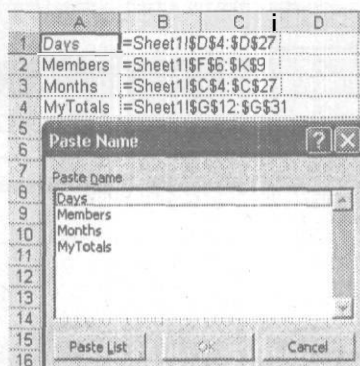


Рис. 3.11. Диалоговое окно вставки имени

Вы получите список всех имен в активной рабочей книге.

Хотя этот способ удобен для идентификации определенных диапазонов, он требует, чтобы вы вручную выбрали именованный диапазон или воспользовались диалоговым окном Перейти (Go To). Однако, получив список именованных диапазонов, можно удалить все адреса ячеек, на которые ссылаются имена, и заменить их простой функцией гиперссылки.

Можно создать список всех именованных диапазонов, щелкнув любой элемент списка в котором, вы перейдете к указанному диапазону. Например, предполо-

¹ Кнопка ОК в этой ситуации неактивна, и щелкнуть надо на кнопке Все имена (Paste List). — Примеч. ред.

жим, список имен находится в столбце А, начиная с ячейки А1. В ячейке В1 введите следующую формулу:

```
=HYPERLINK("[Book1.xls]"&A1,A1)
```

В русской версии Excel

```
=ГИПЕРССЫЛКА("[Book1.xls]"&A1,A1)
```

Скопируйте эту формулу вниз на необходимое количество строк и замените Book1.xls именем вашей книги.

Способ 2

Этот способ очень прост, но малоизвестен. Мы узнали о нем от одного из членов форума по Excel на сайте OzGrid в разделе «Hey! That is Cool!» (<http://www.ozgrid.com/forum/forumdisplay.php?fid=14>).

Все, что нужно сделать, — это уменьшить масштаб рабочего листа Excel. Выберите значение, меньшее 40 %, например, 39 % или меньше. Все именованные диапазоны появятся на листе (рис. 3.12).

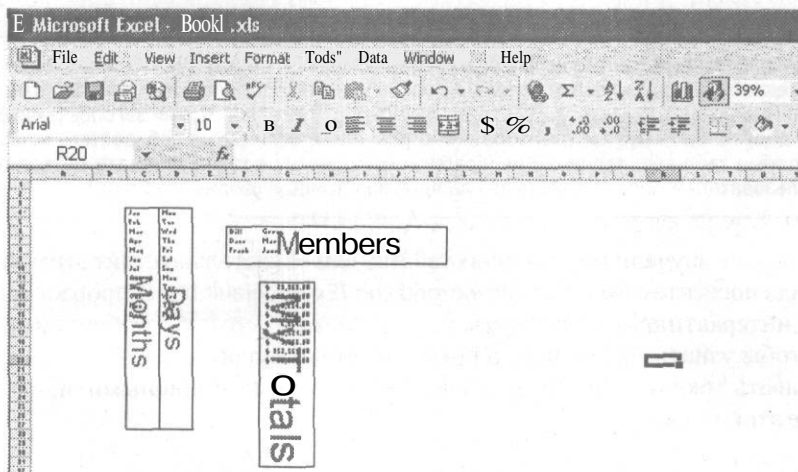


Рис. 3.12. Масштабирование именованных диапазонов для упрощения идентификации

Трюки со сводными таблицами

Трюки с 45 по 49

Сводные таблицы (PivotTables) — это одна из самых мощных и привлекательных возможностей Excel, хотя многие пользователи и не знают, что же они умеют. Сводные таблицы умеют отображать и выделять различную информацию из таблицы данных, находящейся в Microsoft Excel или другой базе данных совместимого типа. Часто при помощи сводных таблиц извлекают статистическую информацию из необработанных данных. Различные поля сводной таблицы можно передвигать, чтобы увидеть данные с разных точек зрения.

ПРИМЕЧАНИЕ

Необработанные данные для сводной таблицы должны быть организованы в формате классической таблицы. Первая строка должна содержать заголовки, а данные, относящиеся к ним, должны находиться в соответствующем столбце под своими заголовками. В данных не должно быть пустых столбцов или строк. Даже если вы не собираетесь использовать сводные таблицы, хранение необработанных данных в таком формате позволит другим пользователям проанализировать данные при помощи сводных таблиц.

Если вы еще не изучали мир сводных таблиц, вам обязательно стоит этим заняться. Для начала посетите сайт <http://www.ozgrid.com/Excel/default.htm> и проработайте бесплатные интерактивные справочные материалы от Microsoft по сводным таблицам Excel. Чтобы узнать еще больше о преимуществах сводных таблиц, а также о том, как создавать трюки, делающие их еще более гибкими и мощными, продолжайте изучение этой главы.

Т Р Ю К
№45

Сводные таблицы: сами по себе — трюк

Сводные таблицы — это одна из самых необузданных, но и наиболее мощных возможностей Excel, сама по себе являющаяся изобретательным трюком, для постижения которого требуется множество опытов и экспериментов.

Мы часто используем сводные таблицы при разработке электронных таблиц для наших клиентов. Когда клиенты видят сводную таблицу, они практически всегда спрашивают, а смогут ли они сами создавать их. Хотя любой пользователь может создать сводную таблицу, к сожалению, многие люди стараются избегать их, так как эти таблицы кажутся слишком сложными. Действительно, когда вы впервые

работаете со сводной таблицей, процесс может показаться немного запутанным. Определенно требуется некоторая настойчивость.

Вы обнаружите, что ваша настойчивость окупится, как только познакомитесь с лучшим свойством сводных таблиц — возможностью манипулировать ими методом проб и ошибок и немедленно видеть результаты этих манипуляций. Если вы получили неожиданный результат, то при помощи возможности Excel Отмена (Undo) всегда сможете отменить его и прийти к другому. Что бы вы ни делали, вы не измените структуру исходной таблицы, а потому не причините своей работе никакого вреда.

Почему они называются сводными таблицами?

Сводные таблицы позволяют сочетать данные различными способами, используя технику перетаскивания (drag-and-drop), и немедленно получать результаты. Это интерактивное средство: закончив создание таблицы, легко можно видеть, как повлияет на информацию перемещение данных. Все станет совершенно ясно, как только вы попытаетесь работать со сводными таблицами.

Даже для опытных разработчиков сводных таблиц в процессе получения желаемых результатов присутствует элемент проб и ошибок. Вы часто будете перетаскивать и разными способами крутить данные в таблице!

Для чего подходят сводные таблицы?

Сводные таблицы позволяют получать итоговую информацию из таблиц данных. Представьте таблицу с именами, адресами, возрастом, местом работы, телефонными номерами и почтовыми индексами. При помощи сводной таблицы легко можно узнать:

- у скольких людей одинаковые имена;
- сколько людей живут по адресу с одинаковым почтовым индексом;
- сколько людей работают на одном и том же месте.

Можно получить и такую информацию, как:

- список людей, работающих на одном и том же месте;
- список адресов с одинаковым почтовым индексом.

Если данные нужно разобрать, разделить и составить отчет, сводные таблицы становятся жизненно важной составляющей набора инструментов.

Зачем использовать сводные таблицы, если электронные таблицы предлагают так много возможностей для анализа?

Вероятно, самое большое преимущество сводных таблиц в том, что можно генерировать и извлекать полезную информацию из огромной таблицы буквально за

минуты, и не занимая слишком много компьютерной памяти. Во многих случаях такой же результат можно получить из таблицы данных при помощи встроенных функций Excel, но для этого понадобилось бы намного больше времени и памяти.

Еще одно преимущество сводных таблиц в том, что, если нужна какая-то новая информация, можно просто перетащить ее мышью. Кроме того, можно сделать так, чтобы информация обновлялась каждый раз при открытии рабочей книги или щелчке на кнопке Обновить данные (Refresh).

Сводные диаграммы расширяют возможности сводных таблиц

Microsoft впервые представила сводные диаграммы (PivotCharts) в Excel 2000. На основе таблицы, которую вы создаете при помощи мастера сводных таблиц, можно создать сводную диаграмму (или, точнее, отчет сводной диаграммы (Pivot-Chart Report) на основе сводной таблицы). Во время создания сводной таблицы безо всяких усилий с вашей стороны можно одновременно создать сводную диаграмму. Сводные диаграммы позволяют создавать интерактивные диаграммы, что раньше было невозможно без помощи VBA или элементов управления Excel.

Подробнее о мастере сводных таблиц (PivotTable Wizard) — далее в этой главе.

ПРИМЕЧАНИЕ

Сводные диаграммы недоступны в Excel для Macintosh.

Создание таблиц и списков для использования в сводных таблицах

При создании сводной таблицы необходимо правильно организовать набор данных в таблице или списке. Все данные сводных таблиц будут основаны на этой таблице или списке, поэтому жизненно важно, чтобы все таблицы или списки были настроены одинаковым образом.

В этом контексте таблица — это всего лишь список с заголовком, несколько столбцов данных и разные заголовки для каждого столбца. Часто в контексте таблицы упоминается список. Хорошая практика в создании списков существенно поможет, когда к данным настанет время применить сводную таблицу.

Когда вы извлекаете данные при помощи функций поиска или функций баз данных, к настройке таблицы или списка можно отнестись чуть менее серьезно. Вы всегда можете компенсировать это за счет функции и, вероятно, все так же получить нужный результат. Однако проще всего настраивать список или таблицу настолько точно и аккуратно, насколько это возможно. Встроенные возможности Excel базируются на множестве предположений о макете и настройке данных. Хотя они обеспечивают некоторую степень гибкости, чаще всего проще работать, если держаться следующих правил относительно настройки таблицы или списка:

- Заголовки необходимы, коль скоро сводная таблица использует их для имен полей. Заголовки всегда должны находиться в строке непосредственно над данными. Никогда не оставляйте пустую строку между данными и заголовками. Кроме того, заголовки должны быть отформатированы по-другому, например, жирным шрифтом.
- Над заголовками оставьте, по меньшей мере, три пустые строки. В них можно будет поместить формулы, важные данные и т. д. При желании эти строки можно скрыть.
- Если на одном рабочем листе несколько списков или таблиц, оставляйте, по меньшей мере, один пустой столбец между списками или таблицами. Так Excel будет проще понять, что это различные сущности. Однако если списки и таблицы связаны между собой, лучше объединить их в одну большую таблицу.
- Избегайте пустых ячеек в данных. Вместо того чтобы оставлять пустые ячейки для одинаковых данных в столбце, повторяйте эти данные необходимым количеством раз.
- Сортируйте списки данных, лучше по самому левому столбцу. Так данные будет проще читать и интерпретировать.

Если как можно строже соблюдать эти советы, работа со сводными таблицами станет относительно простой задачей.

На рис. 4.1 показаны хорошо организованная таблица данных и процесс создания сводной таблицы. Обратите внимание, что в столбце Date повторяется множество одинаковых дат. На переднем плане вы видите этап создания макета (layout) данных, демонстрирующий дополнительные поля Страница (Page), Строка (Row) и Column (Столбец) и обязательное поле Данные (Data).

Product	Date	Amount	Commission/Sale
Visual Basic Collection	Wed 3 Jul 02	\$ 13.43	Commission
Print Table and PrintChart Wizard - Layout	5 Jul 02	\$ 6.00	Commission
	1 Jul 02	\$ 11.99	Commission
	1 Jul 02	\$ 8.98	Commission
	5 Jul 02	\$ 8.98	Commission
	5 Jul 02	\$ 7.48	Commission
	9 Jul 02	\$ 7.48	Commission
	7 Jul 02	\$ 13.43	Commission
	9 Jul 02	\$ 8.70	Commission
	Aug 02	\$ 7.50	Commission
	Aug 02	\$ 7.48	Commission
	Aug 02	\$ 7.48	Commission
	Aug 02	\$ 5.98	Commission
	Aug 02	\$ 5.98	Commission
	Aug 02	\$ 3.00	Commission
	Aug 02	\$ 8.98	Commission
	Aug 02	\$ 8.70	Commission
	Sep 02	\$ 43.96	Sale
	Sep 02	\$ 3.00	Commission
	Sep 02	\$ 43.96	Sale
	Sep 02	\$ 20.99	Commission

Рис. 4.1. Сводная таблица, созданная на основе хорошо настроенной таблицы данных

Мастер сводных таблиц и сводных диаграмм

Как упоминалось ранее, чтобы помочь пользователям в создании сводных таблиц, Excel предлагает средство Мастер сводных таблиц и сводных диаграмм

(PivotTable and PivotChart Wizard). Этот мастер проводит вас через четыре шага процесса создания сводной таблицы, в ходе которого вы сообщаете Excel следующее:

- как организованы данные и нужно ли создавать связанную сводную диаграмму (если сводные диаграммы доступны в вашей версии Excel);
- где хранятся данные: диапазон в этой рабочей книге, база данных, другая рабочая книга и т. д.;
- какой столбец данных соответствует какому полю: дополнительные поля Страница (Page), Строка (Row) и Столбец (Column) и обязательное поле Данные (Data);
- куда следует поместить сводную таблицу (например, на новый рабочий лист или в существующий).

В ходе создания можно также делать множество дополнительных шагов по настройке сводной таблицы, однако большинству пользователей проще делать это уже после того, как они сообщат Excel, куда следует поместить сводную таблицу.

ПРИМЕЧАНИЕ

У Excel 2000 и более поздних версий есть большое преимущество перед Excel 97: они позволяют выбирать, как следует настроить данные, уже после того, как работа мастера закончится.

Теперь, когда вы больше узнали о сводных таблицах и их предназначении, настало время изучить некоторые удобные трюки, делающие эту возможность еще более мощной.

ТРЮК
№ 46

Совместное использование сводных таблиц, но не их данных

Создайте снимок экрана вашей сводной таблицы, которой более не требуются лежащие в ее основе структуры данных.

Иногда может понадобиться отправить сводные таблицы другим пользователям для просмотра, но по каким-либо причинам нельзя отправить лежащие в основе этих сводных таблиц данные. Возможно, вы хотите, чтобы другие увидели только определенные данные по причинам конфиденциальности. В таком случае можно создать статическую копию сводной таблицы и позволить получателю увидеть только то, что ему необходимо увидеть. Лучше всего, если размер файла, где хранится статическая копия, будет составлять лишь небольшую долю от размера исходного файла.

Предполагая, что в рабочей книге есть сводная таблица, все, что нужно, — это выделить всю сводную таблицу, скопировать ее и на чистом листе выбрать команду Правка ▶ Специальная вставка ▶ Значения (Edit > Paste Special ▶ Values). Теперь можно переместить этот рабочий лист в другую рабочую книгу или, возможно, использовать как есть.

Единственный недостаток этого способа в том, что вместе со значениями Excel не вставляет форматирование сводной таблицы. Это может усложнить чтение статической копии и, возможно, сделать ее менее впечатляющей. Если вы хотите включить и форматирование, можете сделать статическую картинку (в противоположность статической копии) вашей сводной таблицы и вставить ее на чистый рабочий лист. Так вы получите полноцветный форматированный снимок исходной сводной таблицы, к которому сможете применить любое желаемое форматирование, не беспокоясь о том, что оно будет утеряно при обновлении исходной сводной таблицы. Это происходит благодаря тому, что полноцветный отформатированный снимок никак не связан с исходной сводной таблицей.

Чтобы создать статическую картинку, отформатируйте сводную таблицу нужным образом и затем выделите любую ячейку в этой таблице. На панели инструментов Сводные таблицы (PivotTable) выберите команду Сводная таблица ► Выделить ► Таблицу целиком (PivotTable ► Select ► Entire Table). Когда таблица будет выделена, удерживая клавишу Shift, выберите команду Правка ► Копировать рисунок (Edit ► Copy Picture). В раскрывшемся диалоговом окне Копировать рисунок (Copy Picture) установите показанные на рис. 4.2 переключатели и щелкните на кнопке ОК.

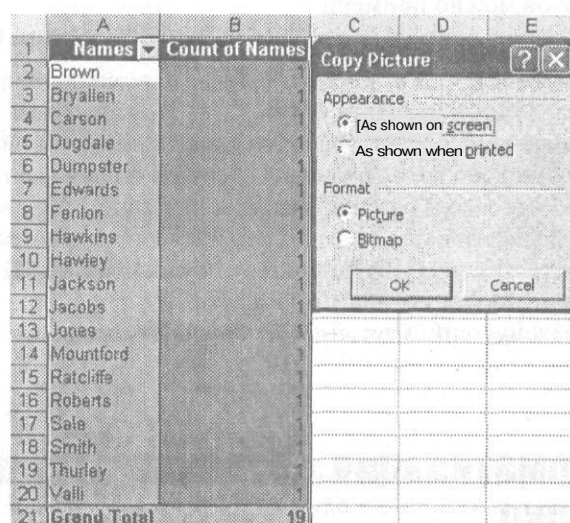


Рис. 4.2. Диалоговое окно копирования рисунка

Наконец, щелкните в любом месте за пределами сводной таблицы и выберите команду Правка ► Вставить (Edit ► Paste). Вы получите полноцветный форматированный снимок сводной таблицы (рис. 4.3), не потеряв при этом форматирование. Это может быть очень удобно, особенно если нужно отправлять сводные таблицы другим пользователям по электронной почте. Они получают всю необходимую информацию и соответствующее форматирование, но размер файла будет небольшим, и никто не сможет изменить ваши данные. **Вдобавок**, другие пользователи смогут увидеть только то, что вы разрешите им увидеть.

	A	B	C	D	E
1	Original Pivot Table		Picture Of tpivot Table		
2	Names	Count of Names	Names	Count of Names	
3	Brawn		Brown		
4	Bryallen		Bryallen		
5	Carson		Carson		
6	Dugdale		Dugdale		
7	Dumpster		Dumpster		
8	Edwards		Edwards		
9	Fenlon		Fenlon		
10	Hawkins		Hawkins		
11	Hawley		Hawley		
12	Lackson		Jackson		
13	Jacobs		Jacobs		
14	Jones		Jones		
15	Mountford		Mountford		
16	Ratcliffe		Ratcliffe		
17	Roberts		Roberts		
18	Sale		Sale		
19	Smith		Smith		
20	Thurley		Thurley		
21	Valli		Valli		
22	Grand Total	19	Grand Total		19

Рис. 4.3. Исходная сводная таблица и рисунок со сводной таблицей

Этот метод рисунков можно применить и к диапазону ячеек. Выполните предыдущие шаги или воспользуйтесь малоизвестным инструментом Камера (Camera).

Чтобы воспользоваться вторым методом, выберите команду Вид ► Панели инструментов ► Настройка (View ► Toolbars ► Customize). В диалоговом окне Настройка (Customize) щелкните вкладку Команды (Commands), в поле Категории (Categories) выберите Tools (Сервис), а в поле Команды (Commands) справа найдите пункт Камера (Camera). Щелкните левой кнопкой мыши и перетащите эту кнопку на панель инструментов. Выделите диапазон ячеек, щелкните кнопку Камера (Camera) и щелкните в любом месте таблицы. Вы получите рисунок диапазона, связанный с исходным диапазоном. Какие бы данные или форматирование вы ни применили к исходному диапазону, они будут автоматически отображены на рисунке этого диапазона.

ТРЮК
№47

Автоматизация создания сводных таблиц

Шаги, которые нужно выполнить для создания сводной таблицы, требуют приложения некоторых усилий, и эти усилия зачастую являются излишними. Написав немного кода VBA, вы сможете автоматически создавать простые сводные таблицы.

Сводные таблицы — это весьма разумная и эффективная возможность, которую можно применять к данным, хранящимся в списке или таблице. К сожалению, простой мысли о процессе создания сводной таблицы достаточно, чтобы пользователи отказались от идеи экспериментировать с ними. Хотя некоторые настрой-

ки сводных таблиц могут быть достаточно сложными, большую часть сводных таблиц можно создавать легко и просто. Два чаще всего задаваемых вопроса об Excel относятся к тому, как подсчитать количество всех элементов в списке и как создать список уникальных элементов на основе списка, содержащего много повторяющихся элементов. В этом разделе мы продемонстрируем, как быстро и просто создать сводную таблицу, которая будет выполнять эти задачи.

Предположим, в столбце A есть длинный список имен, причем в ячейке A1 находится заголовок. Вы хотите узнать, сколько элементов содержится в списке, и создать список уникальных элементов. Выделите ячейку A1 (заголовок) и выберите команду Данные ► Сводная таблица (Data ► PivotTable and PivotChart Report) (или Data ► PivotTable Report на Mac), чтобы запустить мастер сводных таблиц.

Удостоверьтесь, что установлен переключатель В списке или базе данных Microsoft Excel (Microsoft Excel List or Database) и что выделена одна ячейка данных. Это позволит Excel автоматически распознать данные, на основе которых будет далее строиться сводная таблица. Если вы работаете в Windows, то в группе Вид создаваемого отчета (What kind of report do you want to create?) выберите Сводная таблица (PivotTable) (на Macintosh этот вопрос не выводится). Щелкните кнопку Далее (Next). Мастер сводных таблиц должен автоматически выбрать правильный диапазон данных в столбце A и выделить его на листе. Если это так, щелкните кнопку Далее (Next). Иначе выделите диапазон мышью. Щелкните кнопку Макет (Layout) и перетащите в область Данные (Data) то, что будет вашим единственным полем, — над этой областью вы должны увидеть заголовок из ячейки A1. Еще раз перетащите поле с заголовком, на этот раз в область Строка (Row). Теперь окно должно выглядеть, как на рис. 4.4. Щелкните на кнопке ОК.

ПРИМЕЧАНИЕ

На данном этапе, если необходимо, можно дважды щелкнуть кнопку поля в области Данные (Data) (на рис. 4.4 она называется Count of Names) и изменить значение параметра Операция (Summarize by). Можно выбрать Сумма (Sum), Среднее (Average) и т. д. По умолчанию Excel выбирает функцию СЧЕТ (COUNT), если работа идет с текстом, и функцию СУММ (SUM), если это числа.

Теперь в качестве местоположения отчета сводной таблицы выберите Новый лист (New Worksheet) и щелкните кнопку Готово (Finish). На новом рабочем листе вы должны увидеть сводную таблицу, содержащую уникальные имена из исходного списка и указание, сколько раз каждый элемент (имя) содержится в списке.

А что если вы хотите, чтобы все эти шаги вместо вас выполнял макрос, создавая сводную таблицу на основе любого указанного столбца? Если вы просто запишете макрос, он будет работать, только если данные содержат тот же заголовок. Чтобы избежать этого, можно создать простой макрос, хранящийся в вашей рабочей книге или в личной книге макроккоманд (описанной в главе 7), который будет создавать сводную таблицу по любому списку элементов. Для этого нужно написать некоторый общий код VBA и ввести его в стандартный модуль в вашей личной книге макроккоманд или любой другой книге макроккоманд.

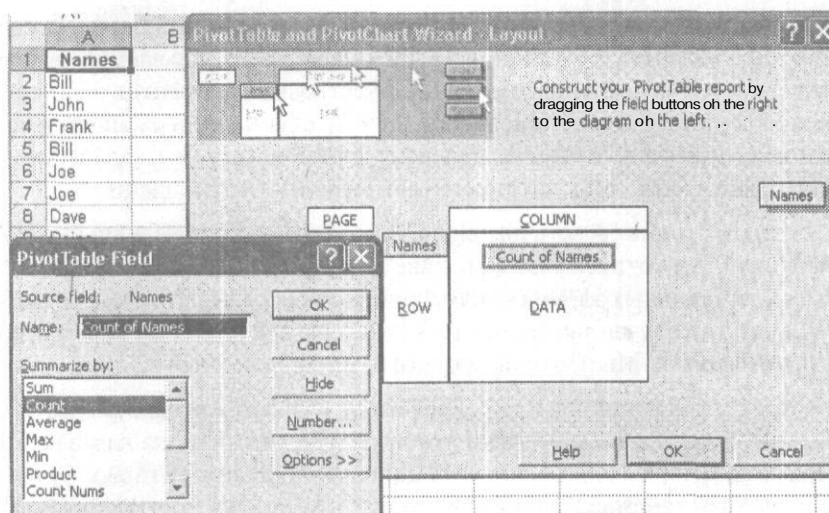


Рис. 4.4. Диалоговые окна Вычисление поля сводной таблицы (PivotTable Field) и Макет (PivotTable Layout)

Выберите команду Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor) (сочетание клавиш Alt/Option+F11) и затем выберите команду Insert ► Module. Введите код, приведенный в листинге 4.1.

Листинг 4.1

```
Sub GetCountC)
Dim Pt As PivotTable
Dim strField As String

strField = Selection.Cells(1,1).Text
Range(Selection. Selection.End(xlDown)).Name = "Items"

ActiveWorkbook.PivotCaches.Add(SourceType:=xlDatabase, _
SourceData:="=Items").CreatePivotTable TableDestination:="", _
TableName:="ItemList"

Set Pt = ActiveSheet.PivotTables("ItemList")
ActiveSheet.PivotTableWizard TableDestination:=Cells(3,1)
Pt.AddFields RowFields:=strField
Pt.PivotFields(strField).Orientation = xlDataField

End Sub
```

Чтобы вернуться в Excel, закройте окно Script или нажмите сочетание клавиш Alt/Apple+Q и сохраните рабочую книгу. Перед запуском этого кода выделите заголовок списка и удостоверьтесь, что список не содержит пустых ячеек.

ПРИМЕЧАНИЕ

Чтобы быстро удалить пустые ячейки, можно отсортировать список.

Код автоматически создаст именованный диапазон на основе вашего списка и присвоит ему имя Items. Затем он создаст на новом листе сводную таблицу, основываясь на данном именованном диапазоне.

В следующий раз, когда у вас появится длинный список данных, вы сможете просто выделить его заголовок и запустить этот макрос. Вся работа по настройке сводной таблицы будет выполнена в мгновение ока.

Т Р Ю К
№ 48

Перемещение общих итогов сводных таблиц

Одно из наиболее раздражающих **свойств** сводных таблиц состоит в том, что общий итог, где суммируются данные, всегда находится внизу таблицы, то есть приходится пролистывать всю таблицу, просто чтобы увидеть цифры. Давайте переместим общий итог наверх, где его проще найти.

Хотя сводные таблицы — это прекрасный инструмент для подведения итогов по данным и выделения значимой информации, у них нет встроенной функции перемещения общего итога наверх, где его легко обнаружить.

Перед тем как мы опишем очень общий способ перемещения общего итога наверх, сначала посмотрим, как это можно сделать при помощи функции **ПОЛУЧИТЬ.ДААННЫЕ.СВОДНОЙ.ТАБЛИЦЫ** (GETPIVOTDATA), которая предназначена специально для извлечения данных из сводной таблицы.

Эту функцию можно использовать так:

```
=GETPIVOTDATA("Sum of Amount", $B$5)
```

В русской версии Excel

```
=ПОЛУЧИТЬ.ДААННЫЕ.СВОДНОЙ.ТАБЛИЦЫ("Sum of Amount", $B$5)
```

или так:

```
=GETPIVOTDATA("Amount", $B$5)
```

В русской версии Excel

```
=ПОЛУЧИТЬ.ДААННЫЕ.СВОДНОЙ.ТАБЛИЦЫ("Amount", $B$5)
```

Обе функции **позволят выделить** нужные данные и будут отслеживать общий итог, если его переместить вверх, вниз, влево или вправо. Мы использовали адрес ячейки \$B\$5, но если вы укажете любую ячейку в пределах сводной таблицы, то всегда получите итог.

В первой функции используется поле Sum of Amount, а во второй — поле Amount. Если в области Данные (Data) сводной таблицы есть поле Amount, необходимо присвоить этому полю имя Amount. Если, однако, поле Amount используется в области Данные (Data) несколько раз, необходимо указать имя, которое вы присвоили ему, или же имя, принятое по умолчанию (рис. 4.5).

Чтобы изменить эти поля, нужно дважды щелкнуть их. Это может сбить с толку, если вы еще не до конца поняли сводные таблицы. К счастью, в Excel 2002

и более поздних версиях процесс стал намного проще, так как можно поместить в ячейку аргументы и применить правильный синтаксис функции с помощью мыши. В любой ячейке введите = (знак равенства) и щелкните ячейку, в которой находится общий итог. Excel автоматически вставит нужные аргументы.

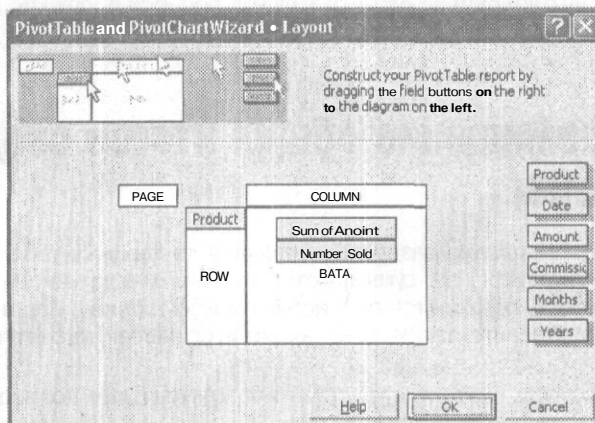


Рис. 4.5. Поле Amount используется два раза; в первом случае ему присвоено имя Sum of Amount, а во втором — Number Sold

ВНИМАНИЕ

К сожалению, если вы используете мастер функций (Function Wizard) или сначала введете =ПОЛУЧИТЬ_ДААННЫЕ_СВОДНОЙ_ТАБЛИЦЫ() (=GETPIVOTDATA()) и уже после этого щелкнете ячейку, в которой находится общий итог, Excel создаст неразбериху, попытавшись поместить в эту ячейку еще одну функцию ПОЛУЧИТЬ_ДААННЫЕ_СВОДНОЙ_ТАБЛИЦЫ (GETPIVOTDATA).

Вероятно, самый простой и менее запутанный способ получить общий итог — воспользоваться функцией

```
=MAX(PivGTCol)
```

В русской версии Excel

```
=МАКС(PivGTCol)
```

где столбец, в котором содержится общий итог, имеет имя PivGTCol.

Кроме того, для получения из сводной таблицы совокупности чисел в зависимости от того, насколько они большие, можно воспользоваться функциями НАИБОЛЬШИЙ (LARGE) и НАИМЕНЬШИЙ (SMALL). Например, следующая формула выделяет второе по величине число из сводной таблицы:

```
=LARGE(PivGTCol,2)
```

В русской версии Excel

```
=НАИБОЛЬШИЙ(PivGTCol,2)
```

Можно добавить несколько строк сразу же над сводной таблицей и поместить туда эти формулы, чтобы сразу же видеть информацию такого типа, не пролистывая сводную таблицу до конца.

Т Р Ю К
№ 49

Эффективная вставка данных из другой рабочей книги

Используйте **данные**, находящиеся в другой рабочей книге, как источник для вашей сводной таблицы.

При создании сводной таблицы в Excel можно выбирать разнообразные источники данных. Самый простой и мощный подход — использовать данные из этой же рабочей книги. К сожалению, по определенным причинам это не всегда возможно. Например, данные, находящиеся в другой рабочей книге, вводятся ежедневно, и пользователи, вводящие данные, не должны видеть сводную таблицу.

Применение динамических именованных диапазонов существенно сократит время обновления, необходимое для вашей сводной таблицы. Так как невозможно сослаться на динамический именованный диапазон из другой рабочей книги, это также означает, что ваша сводная таблица не будет ссылаться на тысячи пустых строк, существенно увеличивая размер файла. Таким образом, можно получить данные из другой рабочей книги и после этого основывать сводную таблицу на данных в этой же рабочей книге, а не использовать внешние ссылки.

В рабочей книге, где будет находиться ваша сводная таблица, вставьте новый рабочий лист и присвойте ему имя Data. Откройте рабочую книгу и рабочий лист с нужными данными. В любой пустой ячейке на этом рабочем листе введите формулу

```
=IF(A1="", "", .A1)
```

В русской версии Excel

```
=ЕСЛИ(A1="", "", .A1)
```

где A1 — первый заголовок таблицы данных.

Выделите ячейку A1. Затем вырежьте ее, активируйте исходную рабочую книгу и вставьте ячейку A1 в ячейку A1 на листе Data. Вы получите ссылку на другую рабочую книгу. Скопируйте эту ячейку в такое количество столбцов, сколько заголовков содержит источник данных. Затем выберите команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define) и в поле Имя (Names in workbook) введите PivotData. В поле Формула (Refers to) введите формулу

```
=OFFSET($A$1,0,0,COUNTA($A:$A),COUNTA($1:$1))
```

В русской версии Excel

```
=СМЕЩ($A$1,0,0,СЧЁТ3($A:$A),СЧЁТ3($1:$1))
```

Щелкните на кнопке Добавить (Add), затем на кнопке ОК. После этого нужно добавить код, который будет выполняться каждый раз при открытии рабочей книги. Правой кнопкой мыши щелкните значок Excel (расположенный в левом верхнем углу окна), в контекстном меню выберите команду Исходный текст (View Code) и введите код из листинга 4.2.

Листинг 4.2

```
Private Sub Workbook_Open()  
  With Worksheets("Data")  
    .Range("2:1000").Clear  
    .Range("1:1").AutoFill .Range("1:1000")  
    .Range("2:1000") = .Range("2:1000").Value  
  End With  
End Sub
```

ПРИМЕЧАНИЕ

Этот способ недоступен на Mac. Вам нужно будет открыть VBE, нажав сочетание клавиш Option+F11 или выбрав команду Tools ► Macro ► Visual Basic Editor. Затем в окне Projects, удерживая клавишу Ctrl, щелкните This Workbook.

Чтобы вернуться в Excel, закройте окно Script или нажмите сочетание клавиш Alt/Apple+Q.

Предыдущий код учитывает только 1000 строк данных. Это число должно быть всегда больше количества строк, которое, по вашему мнению, вам понадобится. Другими словами, если таблица в другой рабочей книге содержит 500 строк, добавьте еще несколько сотен, чтобы учесть возможный рост исходной таблицы.

ВНИМАНИЕ

Не указывайте слишком большие числа (например, 10 000, если только у вас в действительности нет такого количества данных), так как это сильно повлияет на скорость выполнения кода и обновления данных.

Сохраните рабочую книгу, закройте ее и откройте заново, удостоверившись, что включили макросы. Добавленный код выполнится автоматически и скопирует формулы в строке 1 на лист Data, а затем автоматически преобразует все строки, кроме первой, в значения. Вы получите копию источника данных, которая будет обновляться при каждом открытии рабочей книги.

Теперь, если необходимо, можно скрыть этот лист, выбрав команду **Формат ► Лист ► Скрыть** (Format ► Sheet ► Hide) или воспользовавшись способом, описанным в разделе «Трюк № 5. Как скрыть лист, чтобы его невозможно было отобразить».

Теперь, чтобы использовать в качестве основы сводной таблицы этот динамический именованный диапазон, выделите любую ячейку сводной таблицы и на панели инструментов Сводные таблицы (PivotTable) выберите команду Сводная таблица ► Мастер сводных таблиц (PivotTable ► Wizard). Щелкайте кнопку Назад (Back), пока не достигнете шага 1. Установите первый переключатель В списке или базе данных Microsoft Excel (Microsoft Excel List or Database), щелкните на кнопке Далее (Next) и на шаге 2 введите =PivotData (имя динамического именованного диапазона). Щелкните на кнопке Готово (Finish).

Вы не будете страдать от задержки в работе, которая случается, когда сводная таблица ссылается на внешний источник данных, так как сами данные хранятся в той же рабочей книге. Кроме того, коль скоро вы применили динамический именованный диапазон, сводная таблица стала динамической, она не ссылается на множество пустых строк, и размер файла остается допустимым.

Трюки с диаграммами

Трюки с 50 по 59

Диаграммы — одно из самых популярных средств Excel. Благодаря диаграммам электронные таблицы могут эффективно визуально воздействовать на пользователей, оставляя далеко позади простые вычисления. Хотя возможности создания диаграмм в Excel весьма впечатляют, иногда нужно выйти за пределы базовой функциональности, которую обеспечивает встроенный мастер диаграмм, и создать диаграммы, более чувствительные к изменению данных, или просто воспользоваться какими-либо параметрами, не входящими в стандартный набор, предлагаемый Excel. Трюки этой главы позволят делать все это и даже больше.

**ТРЮК
№ 50**

Выделение одного сектора из круговой диаграммы

Хотя круговые диаграммы (pie charts) — это отличное визуальное средство, иногда вам бывает необходимо выделить определенный сектор диаграммы. Отделив его от остальных фрагментов, вы привлечете к нему внимание пользователей.

По умолчанию на круговой диаграмме все секторы разделяются и отодвигаются на одно и то же расстояние. При помощи пары щелчков мыши вы сможете отделить только один сектор.

Для начала создайте базовую круговую диаграмму (рис. 5.1).

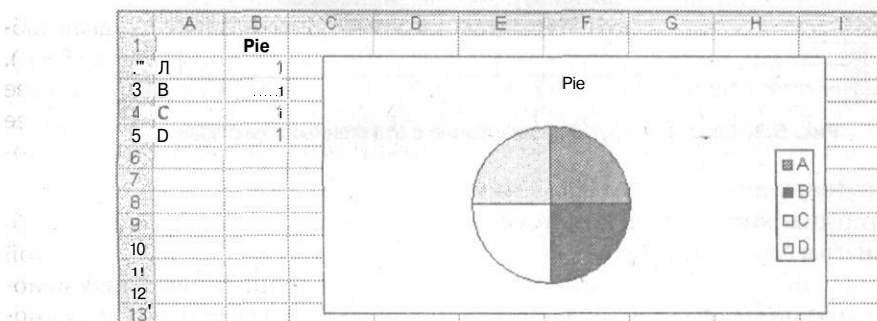


Рис. 5.1. Простая круговая диаграмма, построенная на основе данных рабочего листа

Затем щелкните круговую диаграмму и медленно дважды щелкните сектор, который хотите **отделить**¹. Перетащите выбранный сектор от центра диаграммы и вы увидите эффект разрезания, как на рис. 5.2.

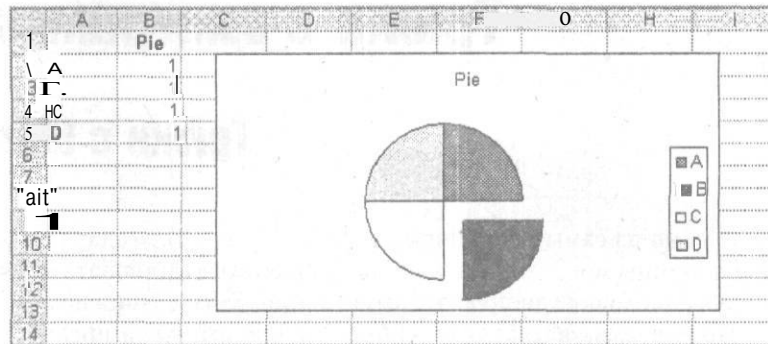


Рис. 5.2. Простая круговая диаграмма с выделенным сектором

Перетащив один сектор, вы оставите остальные секторы на месте. Можно повторить это для других секторов. Эта техника так же хорошо работает с объемной круговой диаграммой (3D pie chart). Чтобы создать объемную круговую диаграмму, щелкните **диаграмму**, затем щелкните ее правой кнопкой мыши, в контекстном меню выберите команду **Тип диаграммы (Chart Type)** и щелкните значок **Объемный вариант круговой диаграммы (3D Pie)**.

Если вы хотите одновременно отделить все секторы, щелчком выделите диаграмму и перетащите один из секторов от центра. Вы увидите эффект, как на рис. 5.3. Чем дальше вы перетащите секторы, тем меньше они станут.

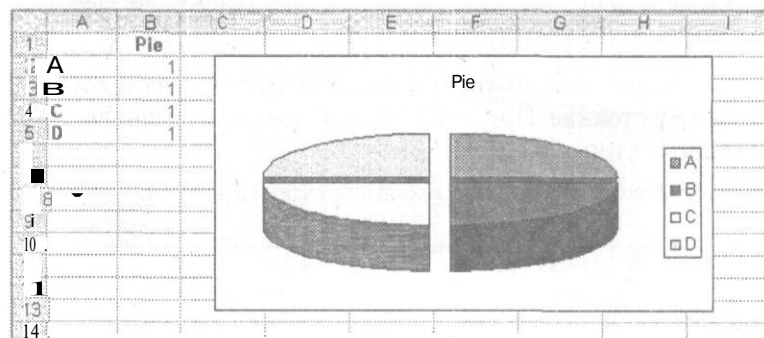


Рис. 5.3. Объемная круговая диаграмма с отделенными секторами

Собрать круговую диаграмму можно, выполнив противоположное действие. Щелкните сектор диаграммы и перетащите его в сторону центра, чтобы совместить все секторы диаграммы.

Энди Поуп

¹ Точнее, щелкните нужный сектор, а затем схватите его мышью и перетащите. — *Примеч. ред.*

Т Р Ю К
№ 51

Создание двух наборов секторов в одной круговой диаграмме

Большинство пользователей считают, что круговые диаграммы ограничены одним набором значений, но мы покажем способ, как создать круговую диаграмму на основе двух столбцов значений.

Немного неожиданно видеть два набора значений, выведенных по отдельным осям на одной диаграмме, но этот эффект стоит усилий. Чтобы узнать, как это работает, сначала создайте базовую круговую диаграмму. Поместите некоторые данные в диапазон B1:C5, выделите этот диапазон и щелкните кнопку Мастер диаграмм (Chart Wizard). На шаге 1 мастера в группе Тип (Chart Type) выберите первую круговую диаграмму. Теперь выполните остальные шаги мастера диаграмм, сделав все необходимые изменения. На шаге 4 поместите диаграмму на **текущий** рабочий лист.

Затем щелкните круговую диаграмму, после этого щелкните ее правой кнопкой мыши в контекстном меню выберите команду Исходные данные ► Ряд (Source Data ► Series). Чтобы добавить еще один ряд, щелкните кнопку **Добавить (Add)**. В поле Имя (Name) выберите ячейку **D1**, а в поле Значения (Values) выберите диапазон D2:D5. Щелкните на кнопке ОК. Вы получите диаграмму, показанную на рис. 5.4.

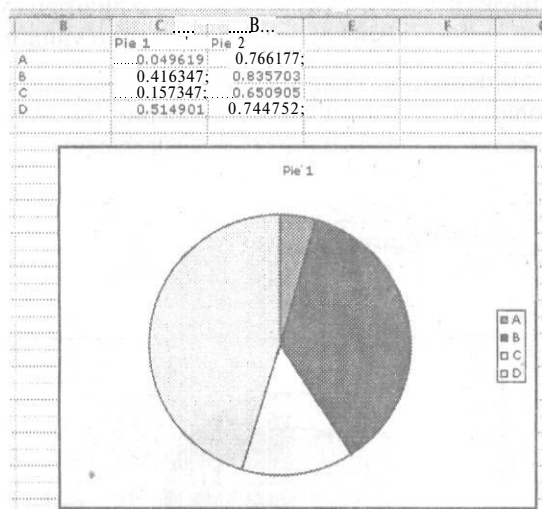


Рис. 5.4. Круговая диаграмма, построенная на основе данных на рабочем листе

Щелкните диаграмму, затем щелкните ее правой кнопкой мыши и в контекстном меню выберите команду **Формат рядов данных (Format Data Series)**. Перейдите на вкладку **Ось (Axis)** и выберите построение ряда по вспомогательной оси (Secondary axis). Внешне диаграмма останется такой же, но ее основа изменится.

Выделите диаграмму и, удерживая левую кнопку мыши, перетащите сектор от центра, затем отпустите кнопку. Вы создадите эффект отделения секторов (рис. 5.5).

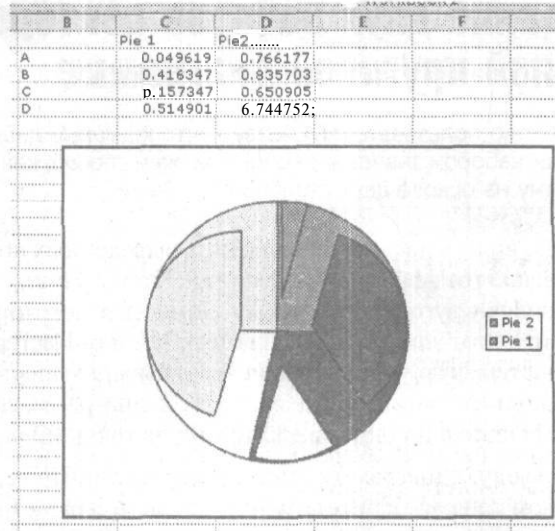


Рис. 5.5. Круговая диаграмма с отделенной вспомогательной осью

Отделив секторы круговой диаграммы, вы не только разделите две оси, открыв вторую круговую диаграмму, но и сожмете круговую диаграмму, построенную по вспомогательной оси, чтобы можно было одновременно увидеть обе диаграммы.

Теперь по очереди выделите все секторы круговой диаграммы и перетащите их обратно в центр, создав диаграмму, показанную на рис. 5.6. Помните, что два медленных щелчка позволят выделить один сектор диаграммы¹.

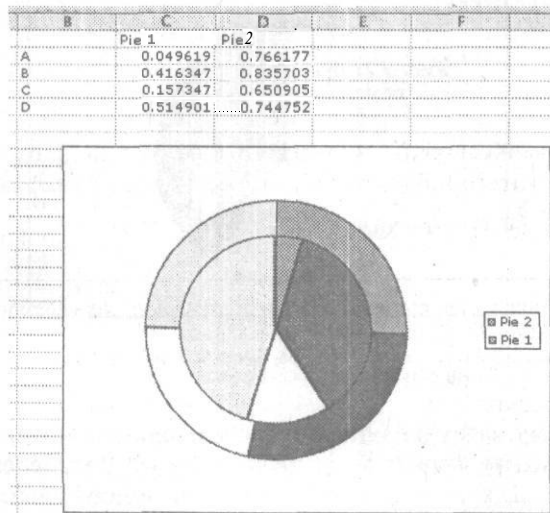


Рис. 5.6. Полная круговая диаграмма, содержащая две диаграммы

¹ Первый щелчок выделяет диаграмму, второй — нужный сектор. — Примеч. ред.

Снова соедините все сектора диаграммы и вы получите полностью функциональную круговую диаграмму с двумя рядами данных на отдельных осях. Теперь вы можете раскрасить и отформатировать диаграмму желаемым образом.

Энди Поуп

**Т Р Ю К
№52**

Создание диаграмм, изменяющихся вместе с данными

Новые данные могут автоматически добавляться и выводиться на диаграммах, как только вы поместите их в электронную таблицу.

Если вместо ссылок на диапазоны использовать динамические именованные диапазоны, новые данные на диаграммах будут выводиться при добавлении их на рабочий лист. Чтобы увидеть, как это работает, начните с чистого рабочего листа и добавьте какие-то данные (рис. 5.7).

	A	B	C
1	Dates	Temperature!	
2	1/10/2003	10	
3	2/10/2003	12	
4	3/10/2003	13	
	4/10/2003	13	
6	5/10/2003	12	
7	6/10/2003	14	
8	7/10/2003	15	
9	8/10/2003	13	
10	9/10/2003	12	
11	10/10/2003	11	

Рис. 5.7. Данные, которые будут помещены на диаграмму

Для создания динамической диаграммы понадобятся два именованных диапазона: один для меток категорий (Dates), а второй — для точек фактических данных (Temperature).

ПРИМЕЧАНИЕ

Если вы не помните, как создавать динамические именованные диапазоны, обратитесь к разделу «Трюк № 42. Создание увеличивающихся и уменьшающихся диапазонов».

Создайте динамический именованный диапазон с именем **TEMP_DATES** для дат в столбце A, выбрав команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define) и введя следующую формулу:

```
=OFFSET($A$1,1,0,COUNTA($A:$A)-1,1)
```

В русской версии Excel

```
=СМЕЩ($A$1,1,0,СЧЁТЗ($A:$A)-1,1)
```

Обратите внимание, что сразу же за аргументом СЧЁТЗ (COUNTA) стоит -1. Это гарантирует, что заголовок не будет включен в именованный диапазон для определенного ряда.

ПРИМЕЧАНИЕ

В этом примере в качестве аргумента функции COUNTA (СЧЁТЗ) указан весь столбец A (\$A:\$A). В ранних версиях Excel лучше ограничивать его намного меньшей группой ячеек, чтобы не перегружать вычисления. Другими словами, указывая диапазон полностью, вы заставляете Excel просматривать тысячи ненужных ячеек. Некоторые функции Excel достаточно умны, чтобы определить, какие ячейки содержат данные, некоторые сделать это не могут. В новых версиях Excel не обязательно строго ограничивать диапазон, так как обработка больших диапазонов в них улучшена.

Затем для температурных данных столбца B настройте еще один динамический диапазон с именем TEMP_READINGS, указав формулу

```
=OFFSET($B$2,0,0,COUNTA($B:$B)-1,1)
```

В русской версии Excel

```
=СМЕЩ($B$2,0,0,СЧЁТЗ($B:$B)-1,1)
```

Теперь можно создать диаграмму, используя созданные динамические именованные диапазоны вместо ссылок на ячейки.

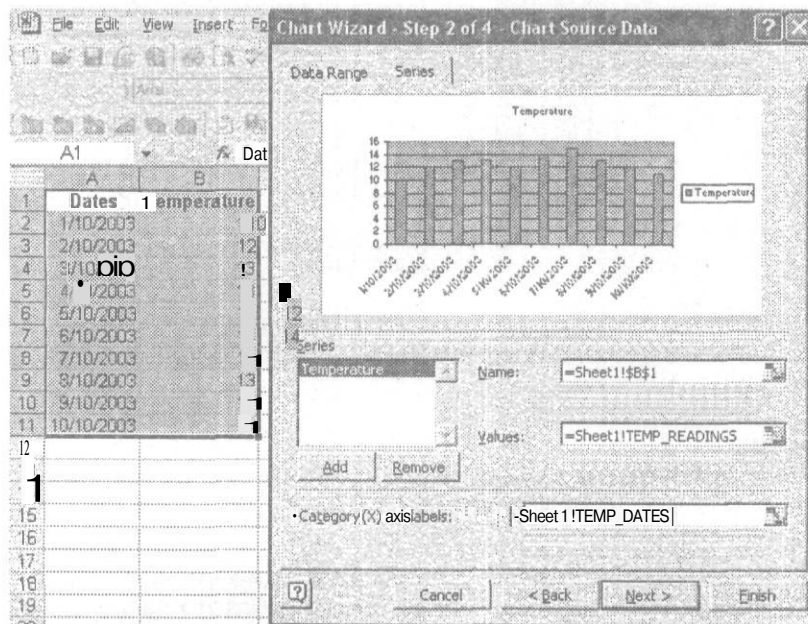


Рис. 5.8. Динамические именованные диапазоны вместо статических ссылок на диапазоны

Выделите данные (диапазон \$A\$1:\$B\$11), затем на стандартной панели инструментов щелкните кнопку Мастер диаграмм (Chart Wizard). На первом шаге мастера выберите тип диаграммы (в этом примере мы используем гистограмму) и щелк-

ните кнопку Далее (Next). На втором шаге вы увидите две вкладки: Диапазон данных (Data Range) и Ряд (Series). Вам нужна вкладка Ряд (Series). Удалите формулу, которая находится в поле Значения (Value), и введите следующую формулу:

```
=Sheet1!TEMP_READINGS
```

ВНИМАНИЕ

В ссылку в формуле важно включить имя листа рабочей книги. Не сделав этого, нельзя ввести в формуле именованный диапазон.

Наконец, удалите формулу в поле Подписи оси X (Category X Labels) и введите следующее:

```
=Sheet1!TEMP_DATES
```

Выполните остальные шаги мастера диаграмм, внося необходимые изменения. Результат должен выглядеть, как на рис. 5.8.

Новые записи в столбце A (Dates) или столбце B (Temperature) будут при добавлении автоматически появляться в настроенной диаграмме.

Построение диаграммы для нескольких последних данных

Еще один тип именованных диапазонов, который можно использовать с диаграммами, — это диапазоны, выбирающие только последние 10 значений (можно указать любое число) из ряда данных. Попробуйте выполнить следующее на тех же данных, которые использовались в первой части этого трюка.

Для данных в столбце A создайте динамический именованный диапазон с именем TEMP_DATES_10DAYS, который ссылается на следующие данные:

```
=OFFSET($A$1,COUNTA($A:$A)-10,0,10,1)
```

В русской версии Excel

```
=СМЕЩ($A$1,СЧЁТ3($A:$A)-10,0,10,1)
```

Для данных в столбце B создайте еще один динамический именованный диапазон с именем TEMP_READINGS_10DAYS и введите следующую формулу:

```
=OFFSET(Sheet1!$A$1,COUNTA(Sheet15!$A:$A)-10,1,10,1)
```

В русской версии Excel

```
=СМЕЩ(Sheet1!$A$1,СЧЁТ3(Sheet15!$A:$A)-10,1,10,1)
```

Если вы хотите изменить количество используемых значений, например, на 20, измените последнюю часть формулы, чтобы она выглядела так:

```
=OFFSET(Sheet1!$A$1,COUNTA(Sheet15!$A:$A)-20,1,20,1)
```

В русской версии Excel

```
=СМЕЩ(Sheet1!$A$1,СЧЁТ3(Sheet15!$A:$A)-20,1,20,1)
```

Использование динамических именованных диапазонов с диаграммами обеспечивает исключительную гибкость и экономит огромное количество времени

и усилий, которые вы потратили бы на настройку диаграмм после добавления еще одной записи к исходным данным.

Энди Поуп

Т Р Ю К
№ 53

Взаимодействие с диаграммами при помощи пользовательских элементов управления

Чтобы сделать диаграммы действительно интерактивными, можно использовать один или несколько динамических диапазонов, а также подключить полосу прокрутки или раскрывающийся список панели инструментов Формы (Forms), чтобы отображать те значения, которые ваши читатели хотели бы изучить более **внимательно**.

Как вы увидели в предыдущем трюке, чтобы добавить диаграммам гибкости, можно воспользоваться динамическими именованными диапазонами. Но также при помощи динамических именованных диапазонов можно создавать интерфейсы, управляющие данными, которые выводятся на диаграмму. Связав динамические именованные диапазоны с пользовательскими элементами управления, вы позволите пользователям изменять данные диаграмм при помощи элементов управления, которые одновременно будут обновлять данные на рабочем листе или наоборот.

Использование динамического именованного диапазона, связанного с полосой прокрутки

В этом примере при помощи полосы прокрутки вы будете отображать значения за определенный месяц периода, включающего 12 месяцев. Полоса прокрутки используется для изменения количества месяцев, для которых строится диаграмма. Также значение полосы прокрутки применяется в динамическом диапазоне, который, в свою очередь, используется как источник данных для диаграммы.

Для начала введите некоторые данные (рис. 5.9).

Выбрав команду Вставка ▶ Имя ▶ Присвоить (Insert ▶ Name ▶ Define), создайте динамический именованный диапазон и назовите его SALES_PERIOD. В поле Формула (Refers to) введите следующую формулу:

```
=OFFSET($B$5.0.0,$C$5.1)
```

В русской версии Excel

```
=СМЕЩ($B$5.0.0,$C$5.1)
```

Благодаря функции СМЕЩ (OFFSET) ячейка \$C\$5 заставляет диапазон, названный SALES_PERIOD, расширяться вверх и вниз, если число в ячейке \$C\$5 изменяется. Другими словами, если в ячейку \$C\$5 ввести число 5, диапазон будет включать ячейки B5:B10.

Month	Product XYZ Sales for 2003
JAN	505
FEB	943
MAR	500
APR	624
MAY	894
JUN	612
JUL	526
AUG	629
SEP	665
OCT	954
NOV	893
DEC	954

Рис. 5.9. Данные на рабочем листе для динамической диаграммы, связанной с полосой прокрутки

ПРИМЕЧАНИЕ

Если вы не хотите, чтобы пользователи видели ячейку C5, можно скрыть ее содержимое, щелкнув ячейку правой кнопкой мыши и в контекстном меню выбрав команду **Формат ячеек** > **Все форматы** (Format Cells > Custom). Введите формат ;;; и щелкните на кнопке OK. На рис. 5.9 содержимое ячейки C5 скрыто.

Создайте диаграмму (лучше всего подходит гистограмма или график). На шаге 2 мастера диаграмм перейдите на вкладку **Ряд** (Series) и в поле **Значения** (Values) введите =<Workbook.xls>ISALES_PERIOD. Так вы сделаете диаграмму динамической.

После того как диаграмма создана, нужно добавить полосу прокрутки с панели инструментов **Формы** (Forms). Для этого проще всего щелкнуть правой кнопкой мыши серую область вверху экрана (которая называется областью панелей инструментов) и выбрать команду **Формы** (Forms). На экране появится панель инструментов **Формы** (Forms) (рис. 5.10).

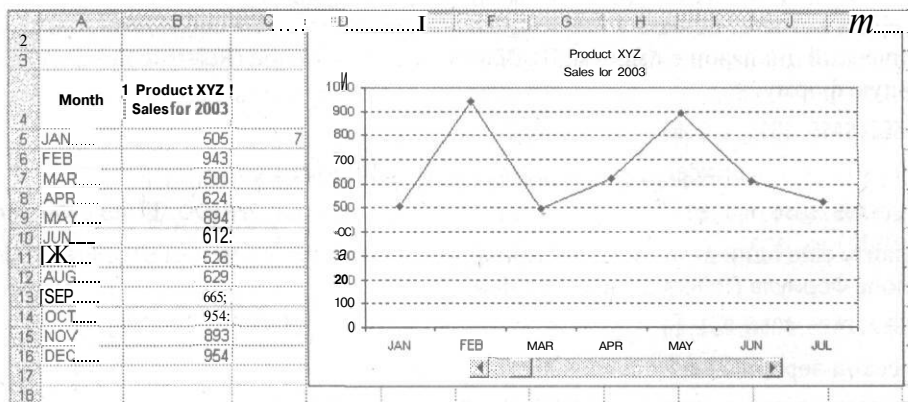


Рис. 5.10. Динамическая диаграмма, связанная с полосой прокрутки

Щелкните значок полосы прокрутки. Вставив полосу прокрутки на экран, выделите ее и переместите на диаграмму. Теперь правой кнопкой мыши щелкните

полосу прокрутки и в контекстном меню выберите команду **Формат объекта (Format Control)**. Укажите минимальное значение 1, максимальное значение 12 и связь с ячейкой \$C\$5. Получившаяся диаграмма будет выглядеть, как на рис. 5.10.

Динамический именованный диапазон, связанный с раскрывающимся списком

Динамическую диаграмму можно связать и с раскрывающимся списком. Начав с данных (см. рис. 5.9), добавьте динамический диапазон, который будет использоваться как источник данных для диаграммы. Динамический диапазон будет связан с раскрывающимся списком. Вы сможете просмотреть результаты теста любого студента из группы студентов. В раскрывающемся списке нужно выбрать имя студента, результаты которого вы хотите просмотреть.

В ячейку B12 вставьте формулу **=AVERAGE(B6:B11)** и скопируйте ее в этой же строке до ячейки F12 (рис. 5.11).

	A	B	C	D	E	F
1						
2						
3						
4	T					
5		Exam A	Exam B	Exam C	Exam D	Exam E
6	Andy	45	52	68	62	47
7	Bernard	64	89	70	55	67
8	Charlie	72	72	90	60	41
9	Dave	57	39	67	74	64
10	Edward	57	93	43	85	56
11	Frank	46	58	42	68	47
12	Average	57	67	64	67	54
13						

Рис. 5.11. Динамическая диаграмма, связанная с раскрывающимся списком

Выберите команду **Вставка ► Имя ► Присвоить (Insert ► Name ► Define)** и создайте динамический диапазон с именем **STUDENTS**. В поле **Формула (Refers to)** введите следующую формулу:

```
=OFFSET($A$5,$G$6,1,1,5)
```

В русской версии Excel

```
=СМЕЩ($A$5,$G$6,1,1,5)
```

Создайте еще один динамический диапазон, на этот раз с именем **STUDENT_NAME**, и в поле **Формула (Refers to)** введите формулу

```
=OFFSET($A$5,$G$6,0,1,1)
```

В русской версии Excel

```
=СМЕЩ($A$5,$G$6,0,1,1)
```

Благодаря ссылке на ячейку \$G\$6 в формуле **СМЕЩ (OFFSET)** вы заставляете диапазоны **STUDENTS** и **STUDENT_NAME** расширяться вверх и вниз, когда число в ячейке \$G\$6 изменяется.

Теперь создайте диаграмму в виде обычной гистограммы, указав для нее диапазон A11:F12. На втором шаге мастера диаграмм перейдите на вкладку Ряд (Series) и в поле Значения (Values) для первого ряда (Frank) введите =<Workbook.xls>!STUDENTS. В поле Имя (Name) введите <Workbook.xls>!STUDENT_NAME.

Теперь на лист нужно поместить поле со списком с панели инструментов Формы (Forms). Сделав это, выберите поле со списком (ComboBox), щелкните его правой кнопкой мыши, в качестве входного диапазона введите \$A\$6:\$A\$11 и укажите связь с ячейкой \$G\$6.

Наконец, в пустую ячейку, например, B4 вставьте функцию CONCATENATE (СЦЕПИТЬ), например, так:

```
=CONCATENATE("Test Result for ".INDEX(A6:A11,G6))
```

В русской версии Excel:

```
=СЦЕПИТЬ("Test Result for ".ИНДЕКС(A6:A11,G6))
```

Щелкните стрелку поля со списком (рис. 5.12), выберите имя студента и вы увидите его результаты теста.

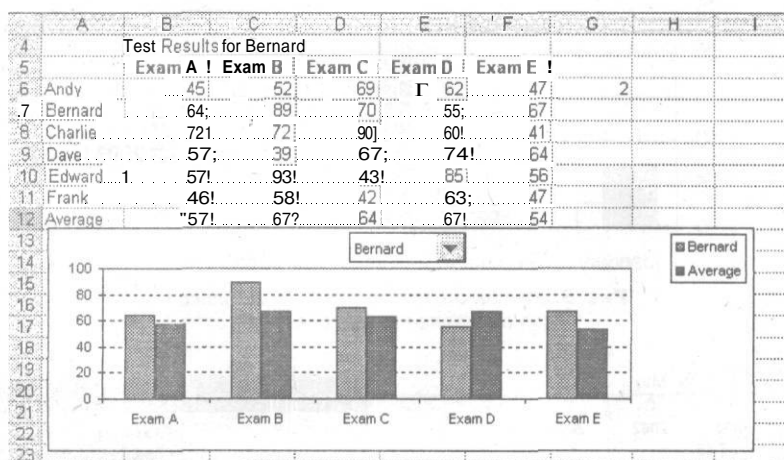


Рис. 5.12. Полная динамическая диаграмма, связанная с раскрывающимся списком

Энди Поуп

ТРЮК № 54 Три быстрых способа обновления диаграмм

Хотя создавать новые диаграммы очень легко, их также необходимо обновлять, чтобы они отражали новые обстоятельства, и для этого могут потребоваться определенные усилия. Сократить объем работы, необходимый для изменения данных, на основе которых построена диаграмма, можно несколькими способами.

Перетаскивание данных

Можно добавить данные к существующему ряду или создать абсолютно новый ряд данных, просто перетащив данные на диаграмму. Excel попытается решить, как следует обработать данные, но при этом он может добавить их к существующему ряду данных, тогда как вы хотели создать новый. Однако можно заставить Excel открыть диалоговое окно, в котором можно будет выбрать необходимое действие.

Попробуйте добавить на лист какие-то данные (рис. 5.13).

	A	B	C	D
1		2001	2002	2003
2	January	7.43	3	10
3	February	1	2	4
4	March	21.3	1	9
5	April	10	3	4
6	May			
7	T			

Рис. 5.13. Данные для обыкновенной гистограммы

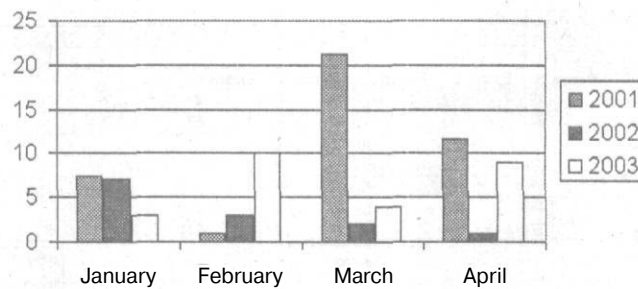


Рис. 5.14. Обыкновенная диаграмма только для определенного диапазона

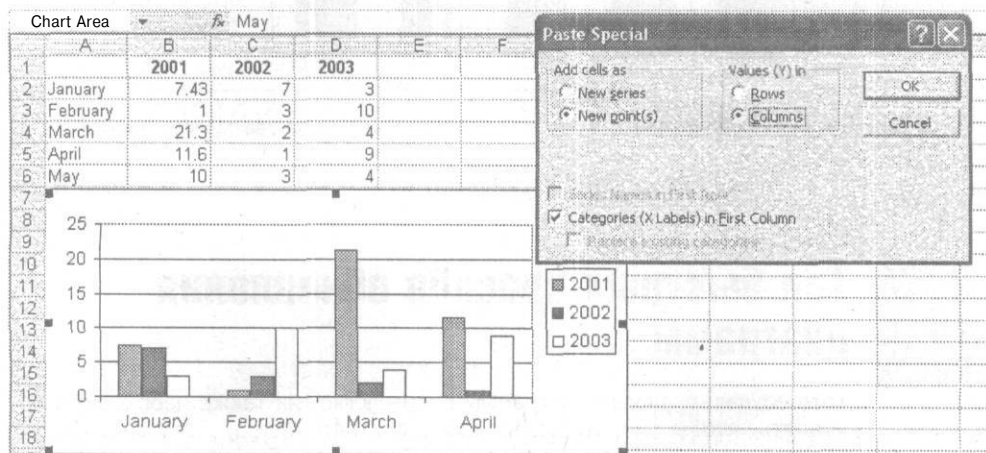


Рис. 5.15. Обыкновенная гистограмма и диалоговое окно специальной вставки

При помощи мастера диаграмм создайте обыкновенную гистограмму только для диапазона \$A\$1:\$D\$5 (рис. 5.14).

Выделите диапазон A6:D6, правой кнопкой мыши щелкните рамку выделения и, удерживая правую кнопку, перетащите данные на диаграмму. Когда вы отпустите кнопку, появится диалоговое окно Специальная вставка (Paste Special) (рис. 5.15).

Выберите параметр В столбцах (Columns) и щелкните на кнопке ОК. Ряд данных для мая (May) будет добавлен на диаграмму (рис. 5.16).

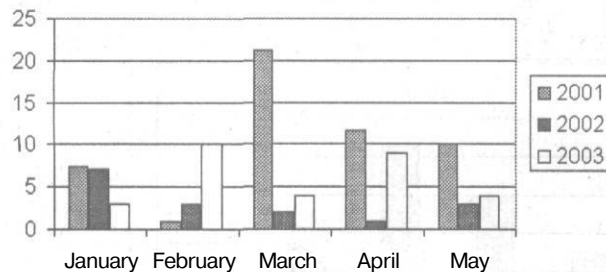


Рис. 5.16. Обыкновенная гистограмма с новым рядом данных

Диалоговое окно Специальная вставка (Paste Special) выполняет большинство действий, которые нужны для этого искусного трюка.

Диаграмма и строка формул

Диаграмму можно обновить и при помощи строки формул. Выделив диаграмму и щелкнув на ней ряд данных, посмотрите на строку формул: вы увидите формулу, которую Excel использует для ряда данных.

В этой формуле, которая называется функцией РЯД (SERIES), обычно указывается четыре аргумента, хотя для пузырьковой диаграммы требуется дополнительный пятый аргумент, обозначающий размер ([Size]).

Синтаксис (или порядок структуры) функции РЯД (SERIES) выглядит так:

=SERIES([Name],[X Values],[Y Values],[Plot Order])

В русской версии Excel

=РЯД([Имя],[Значения X],[Значения Y],[Номер графика])

Так, допустимая функция РЯД (SERIES) может выглядеть, как на рис. 5.17:

=SERIES(Sheet1!\$B\$1,Sheet1!\$A\$2:\$A\$5,Sheet1!\$B\$2:\$B\$5,1)

В русской версии Excel

=РЯД(Лист1!\$B\$1,Лист1!\$A\$2:\$A\$5,Лист1!\$B\$2:\$B\$5,1)

На рис. 5.17 первая часть ссылки, Sheet1!\$B\$1, относится к имени или заголовку диаграммы — 2004. Вторая часть ссылки, Sheet1!\$A\$2:\$A\$5, относится к значениям по оси X, в данном случае — к месяцам. Третья часть ссылки, Sheet1!\$B\$2:\$B\$5, относится к значениям по оси Y, то есть 7.43, 15, 21.3 и 11.6. Наконец, последняя

часть формулы, 1, относится к порядковому номеру графика, или к номеру ряда. В данном случае, когда у нас только один ряд, значение может быть равно только 1. Если бы рядов было несколько, у первого ряда был бы номер 1, у второго — номер 2 и т. д.

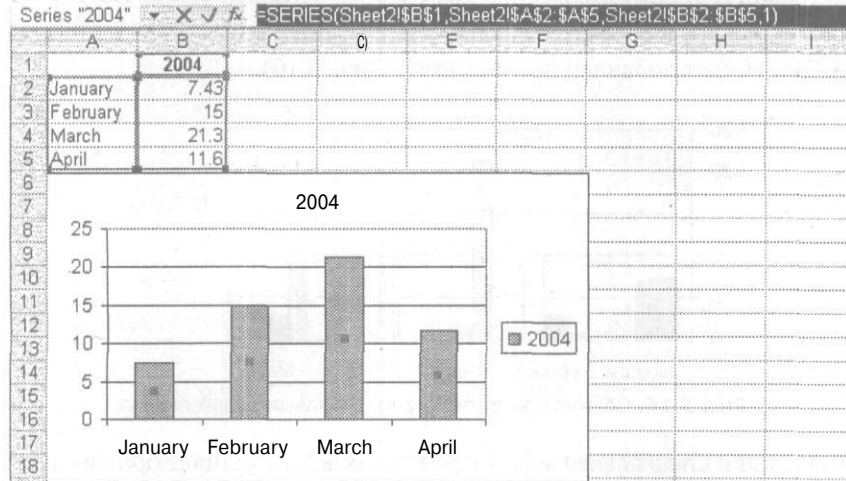


Рис. 5.17. Обыкновенная гистограмма с выделенной строкой формул

Чтобы изменить диаграмму, измените ссылки на ячейки в строке формул.

Помимо ссылок на ячейки, в диаграммы можно вводить и явные значения, известные как массивы констант (подробнее об этом в разделе «Константы в формулах массива» справки по Excel). Для этого добавьте {} (фигурные скобки) вокруг значений по осям X и Y, как показано в следующей формуле:

```
=SERIES("My Bar", {"A", "B", "C", "D"}, {1, 2, 3, 4}, 1)
```

В русской версии Excel

```
=РЯД("My Bar", {"A", "B", "C", "D"}, {1, 2, 3, 4}, 1)
```

В этой формуле РЯД (SERIES) A, B, C и D — это значения по оси X, а 1, 2, 3 и 4 — соответствующие им значения по оси Y.

Используя этот метод, можно создавать и обновлять диаграммы, не храня данные в ячейках.

Перетаскивание граничной области

Если диаграмма содержит ссылки на последовательные ячейки, можно легко увеличивать или уменьшать данные ряда, перетаскивая граничную область в желаемую точку. Медленно шелкните ряд данных, который хотите увеличить или уменьшить. После двух медленных щелчков по краям ряда появятся черные квадратик (маркеры). Все, что нужно, — шелкнуть квадратик и перетянуть границу в желаемом направлении (рис. 5.18).

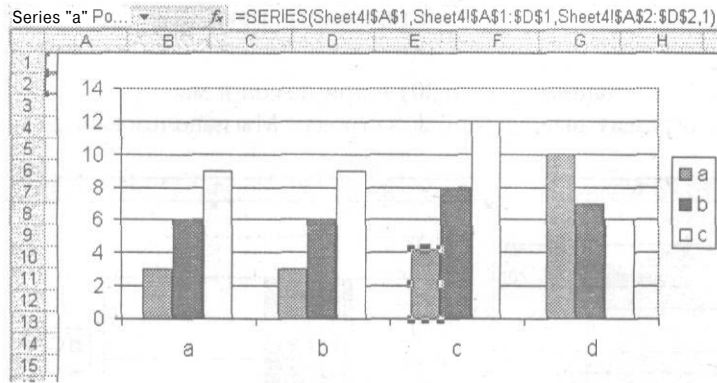


Рис. 5.18. Выделенная граничная область ряда диаграммы

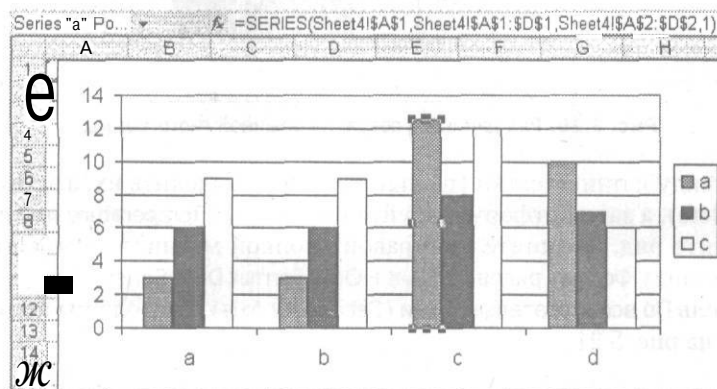


Рис. 5.19. Выделенная граничная область ряда диаграммы после увеличения

Если увеличивать или уменьшать ряд данных (рис. 5.19), исходный источник данных, а также подписи осей, если для них выбран параметр Авто (Auto), обновляются, отражая сделанные изменения.

Это прекрасная возможность проверки сценариев, если вы хотите узнать, какие получатся результаты для различных наборов данных.

Энди Поуп

**ТРЮК
№55**

Трюк с созданием простой диаграммы термометра

В Excel нет диаграммы для термометра. Если нужна именно такая диаграмма, придется ее создать.

Создав обычную гистограмму, сравнивающую значения разных категорий, и затем изменив различные элементы диаграммы, вы создадите визуальную работающую

диаграмму, представляющую собой термометр, приложив совсем немного усилий.

Задайте некоторые данные (рис. 5.20) и при помощи мастера диаграмм создайте обычную гистограмму, выводя данные в строках. Мы использовали диапазон В3:С4.

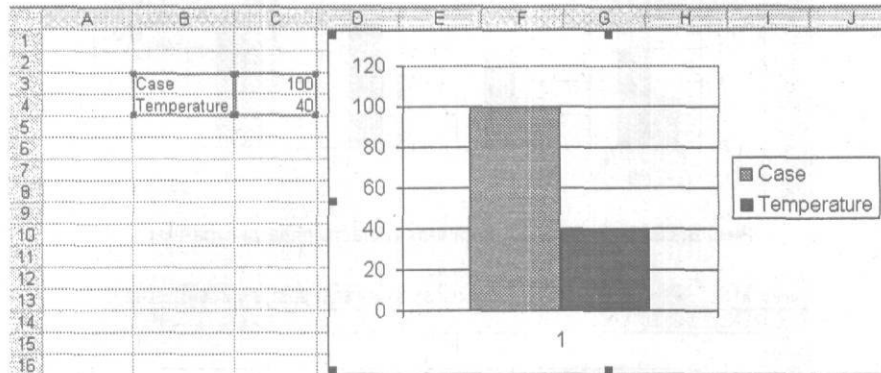


Рис. 5.20. Ряд данных и создание обычной гистограммы

Удалите легенду и линии сетки (щелкните, чтобы выделить их, а затем нажмите клавишу Delete), а затем отформатируйте ряд данных Temperature по оси Y2. Для этого выделите ряд, щелкните его правой кнопкой мыши, в контекстном меню выберите команду **Формат рядов данных** ▶ **Ось** (Format Data Series ▶ Axis) и выберите переключатель **По вспомогательной оси** (Secondary Axis). Вы должны получить диаграмму, как на рис. 5.21.

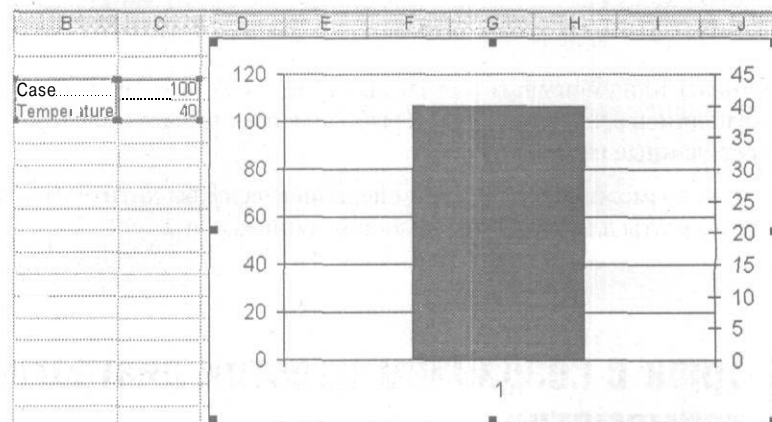


Рис. 5.21. Два ряда, выведенных по оси Y2

Отформатируйте оси Y1 (слева) и Y2 (справа), по очереди выделив их, щелкнув правой кнопкой мыши и выбрав в контекстном меню команду **Формат оси** ▶ **Шкала** (Format Axis ▶ Scale). Для параметра **Минимальное значение** (Minimum) выберите значение 0, для параметра **Максимальное значение** (Maximum) выберите значение 100,

для Цена основных делений (Major Unit) выберите 10, а для Цена промежуточных делений (Minor Unit) выберите 5. Теперь на диаграмме по оси Y1 — ряд Case, по оси Y2 — ряд Temperature, обе оси отформатированы одинаково (рис. 5.22).

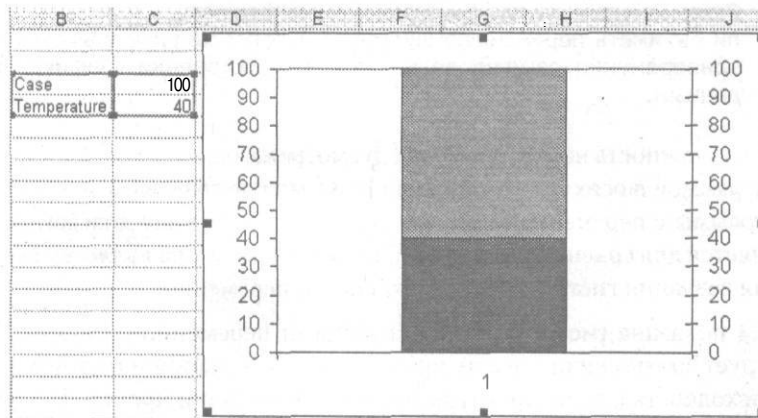


Рис. 5.22. Два ряда данных в двух одинаковых осях

Для ряда данных Case выберите белый цвет, для ряда данных Temperature — красный, для области построения диаграммы — белый. Теперь диаграмма уже должна стать похожей на термометр.

Уменьшите значение свойства Ширина зазора (Gap Width) для обоих столбцов, щелкнув их правой кнопкой мыши и выбрав команду Формат рядов данных ▶ Параметры (Format Data Series ▶ Options). Наконец, удалите ось X (щелкните эту ось и нажмите клавишу Delete), а затем измените размер диаграммы и поместите ее в подходящее место.

Немного поиграв с возможностями диаграмм Excel, можно быстро создать прекрасно выглядящий и работающий термометр (рис. 5.23).

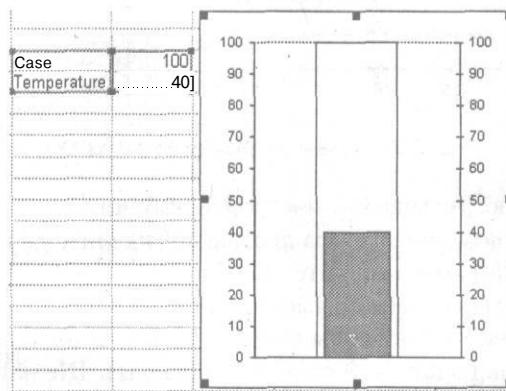


Рис. 5.23. Законченная диаграмма термометра

Энди Поуп

ТРЮК
№ 56

Создание гистограмм с переменной шириной и высотой

Разве не удобно было бы создавать гистограммы, столбцы которых могли бы иметь переменную ширину и высоту? Ширина и высота столбцов одновременно регулировались бы при построении графика по вашим данным.

В Excel эта возможность напрямую не предусмотрена, однако, применив трюк с точечной диаграммой в осях XY (XY scatter chart), можно создавать очень эффективные гистограммы с переменной шириной столбцов. Точечная диаграмма по осям XY применяется для сравнения значений; таким образом, она является прекрасной основой для создания гистограммы со столбцами переменной ширины.

На рис. 5.24 показана гистограмма со столбцами переменной ширины, которая демонстрирует соотношения между процентным содержанием и стоимостью следующих расходов: газ, электричество, вода, еда, поездки и прочие. На оси X (эта ось проходит внизу диаграммы) выводится процентное содержание (%), а на оси Y (ось слева на диаграмме) выводится стоимость (\$).

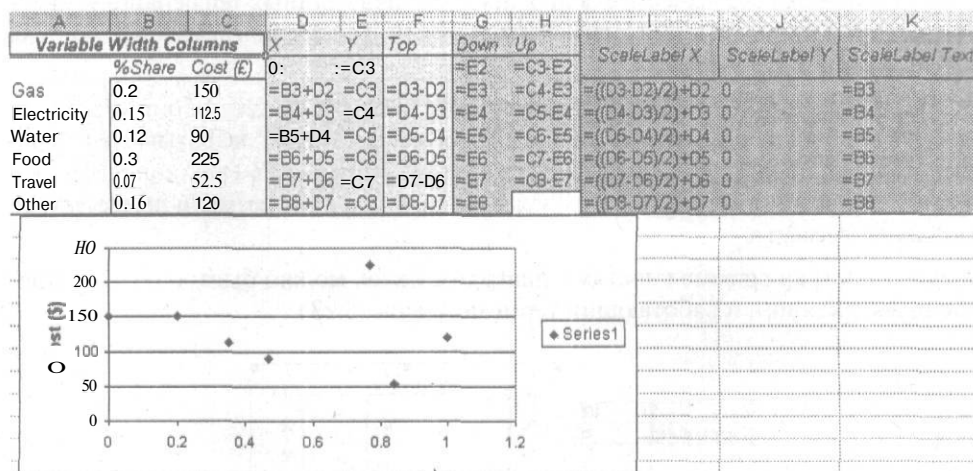


Рис. 5.24. Точечная диаграмма в осях XY

Чтобы создать такую диаграмму, настройте таблицу с некоторыми данными (см. рис. 5.24) и при помощи мастера диаграмм выделите диапазон D2:E8. Затем на шаге 1 мастера диаграмм выберите диаграмму Точечная (XY Scatter). Примите настройки точечной диаграммы по умолчанию (выводятся только точки) и щелкните на кнопке Далее (Next), чтобы перейти к шагу 2 мастера. Удостоверьтесь, что установлен переключатель В столбцах (Columns). Щелкните кнопку Далее (Next), чтобы перейти к третьему шагу мастера, и в поле Ось Y (значений) (Value (Y) Axis) введите Cost (\$). Щелкните кнопку Далее (Next) и удостоверьтесь, что

диаграмма будет создана как объект на текущем, а не на новом листе. Щелкните на кнопке Готово (Finish), чтобы увидеть точечную диаграмму.

ПРИМЕЧАНИЕ

Чтобы увидеть формулы в ячейках, можно нажать сочетание клавиш **Ctrl+~** (то же сочетание используется на Macintosh) или команду Сервис » Параметры » Вид (Tools » Options » View) и в группе Параметры окна (Window) установить флажок Формулы (Formulas).

Настало время повозиться с диаграммой и создать столбцы. Сначала удалите легенду и линии сетки (выделите их и нажмите клавишу Delete) и отформатируйте область построения диаграммы, убрав заливку. Для этого щелкните серый фон правой кнопкой мыши и в контекстном меню выберите команду Формат области построения (Format Plot Area). В группе Заливка (Area) выберите переключатель Прозрачная (None).

Дважды щелкните ось X, чтобы открыть диалоговое окно Формат оси (Format Axis). Щелкните вкладку Шкала (Scale). В области Шкала по оси X (категорий) (Value Axis Scale) для параметра Минимальное значение (Minimum) введите 0, а для параметра Максимальное значение (Maximum) введите 1. Перейдите на вкладку Вид (Pattern) и в группе Основные (Major Tick Mark type) выберите переключатель Нет (None). В группе Метки делений (Tick Mark Labels) также выберите переключатель Нет (None) и щелкните на кнопке ОК. Точечная диаграмма должна выглядеть, как на рис. 5.25.

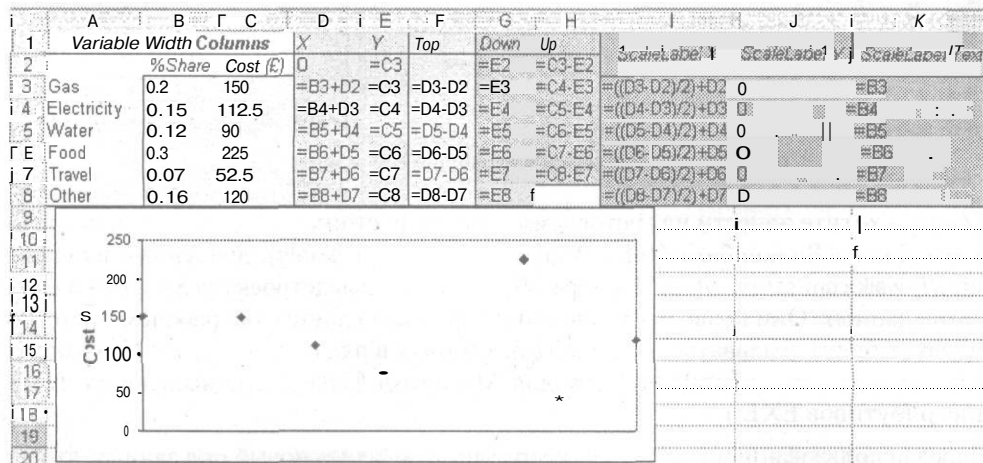


Рис. 5.25. Модифицированная точечная диаграмма

На следующем шаге нужно создать линии для столбцов, поэтому дважды щелкните точку данных, чтобы открыть диалоговое окно Формат ряда данных (Format Data Series). Перейдите на вкладку X-погрешности (X Error Bars). В группе Показать планки погрешностей по X (Display) выберите вариант Минус (Minus), затем перейдите в поле диапазона Пользовательская: - (Custom: - Range) и выделите диапазон F2:F8. Вы получите горизонтальную линию сверху столбца.

Теперь щелкните вкладку **Y-погрешности** (Y Error Bars) и в группе **Показать планки погрешностей по Y** (Display) выберите вариант **Обе** (Both). В поле **Пользовательская: +** (Custom + Range) выберите диапазон **H2:H7**, а в поле **Пользовательская: -** (Custom - Range) выберите диапазон **G2:G8**. Щелкните на кнопке **ОК**. Вы получите вертикальные линии по сторонам столбцов.

Теперь, когда вся трудная работа сделана, настало время немного убраться и добавить подписи. Сначала в диалоговом окне **Формат ряда данных** (Format Data Series) перейдите на вкладку **Patterns** (Вид) и в поле **Marker** (Маркер) выберите **None** (Отсутствует). Результат — гистограмма со столбцами переменной ширины, созданная на основе модифицированной точечной диаграммы по осям XY (рис. 5.26).

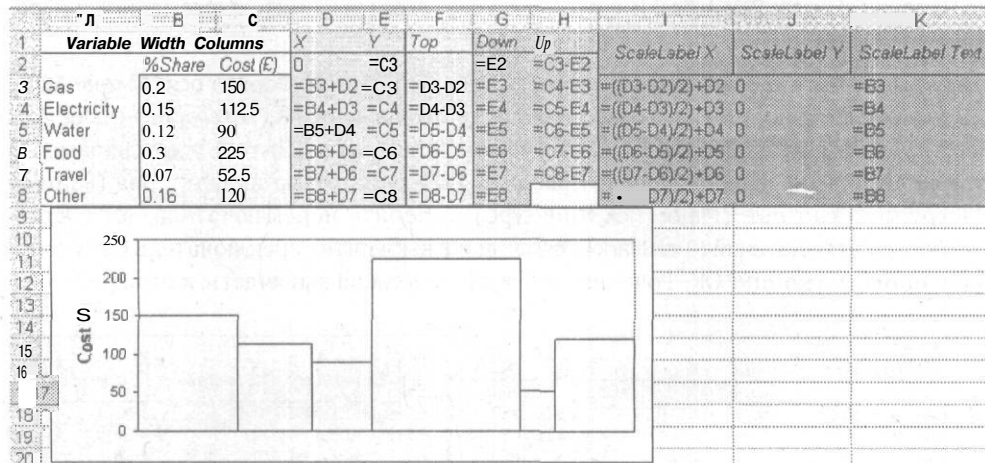


Рис. 5.26. Гистограмма со столбцами переменной ширины

Если вы хотите нанести на гистограмму подписи, стоит загрузить утилиты диаграмм Джона Волкенбаха (*John Walkenbach's Chart Tools*), доступные на сайте <http://j-walk.com/ss/excel/files/charttools.htm>. Часть этих надстроек предназначена для меток данных. Они позволяют показывать диапазон данных на рабочем листе, из которых будут создаваться подписи для данных в рядах диаграммы. (К сожалению, утилиты не работают в Excel для Macintosh, даже после выделения их из дистрибутивов EXE.)

Перед использованием *Chart Tools* необходимо добавить новый ряд данных, чтобы создать подписи по оси X для диаграммы. Выделите диаграмму, щелкните ее правой кнопкой мыши, в контекстном меню выберите команду **Исходные данные** (Source Data) и перейдите на вкладку **Ряд** (Series). Щелкните кнопку **Добавить** (Add), чтобы добавить новый ряд. В поле **Значения X** (X Values) выделите диапазон **I3:I8**, а в поле **Значения Y** (Y Values) — диапазон **J3:J8**. Отформатируйте новый ряд данных, убрав маркеры. Для этого нужно открыть диалоговое окно **Формат рядов данных** (Format Data Series), перейти на вкладку **Вид** (Patterns) и в группе **Маркер** (Marker) выбрать переключатель **Отсутствует** (None).

Настало время применить надстройку. Удостоверьтесь, что выбран ряд Series 2 (Ряд 2), и выберите команду Chart ► JWalk Chart Tools. В открывшемся диалоговом окне выберите диапазон меток данных K3:K8.

Чтобы добавить еще один ряд данных, из которого будут взяты подписи для столбцов диаграммы, выделите диаграмму, щелкните ее правой кнопкой мыши и в контекстном меню выберите команду Исходные данные (Source Data) и перейдите на вкладку Ряд (Series). Щелкните кнопку Добавить (Add), чтобы добавить новый ряд. В поле Значения X (X Values) выделите диапазон I3:I8, а в поле Значения Y (Y Values) выделите диапазон C3:C8. И снова отформатируйте новый ряд данных, убрав маркеры. Для этого откройте диалоговое окно Формат рядов данных (Format Data Series), перейдите на вкладку Вид (Patterns) и в группе Маркер (Marker) выберите Отсутствует (None).

Еще раз примените надстройку. В этот раз выделите Series 3 (Ряд 3) и свяжите подписи данных с диапазоном A3:A8. Результат должен выглядеть, как на рис. 5.27.

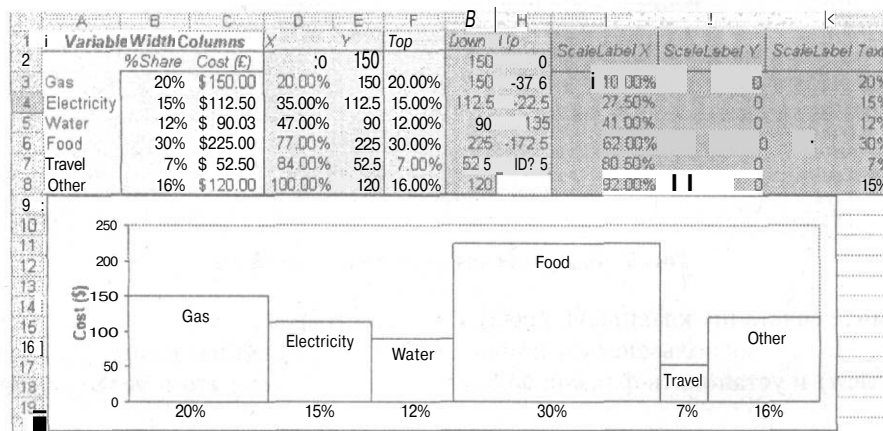


Рис. 5.27. Завершенная гистограмма со столбцами переменной ширины

Удивительное свойство этого типа диаграмм состоит в том, что полосы будут расширяться или сжиматься по оси X в сторону оси Y, когда значения в столбцах % Share или Cost (\$) будут изменяться, Отлично!

Энди Поуп



Создание диаграммы спидометра

Можно создать действительно впечатляющую работающую диаграмму спидометра с движущейся стрелкой, комбинируя кольцевую и круговую диаграммы. Кроме того, спидометром можно управлять с помощью полосы прокрутки.

Мастер диаграмм Excel предлагает множество различных типов диаграмм, кроме, к сожалению, диаграммы спидометра. Диаграмма спидометра — это весьма ловкий

способ представления данных. При помощи трюков этого раздела можно создать диаграмму спидометра и добавить полосу прокрутки панели инструментов Элементы управления (Control Toolbox), которая будет изменять диаграмму и одновременно данные на рабочем листе.

Сначала нужно настроить некоторые данные (рис. 5.28) и создать кольцевую диаграмму (doughnut chart). Кольцевые диаграммы работают схожим образом с круговыми, но они могут содержать несколько рядов, в отличие от круговых.

	A	B	E	F		
1	Bands	Dial	Dial Labels	Labels	Needle	Share
2	25	180	180	=\$F\$3 & "% Share"	200	
3	75	= (180/100)*A2	9	0	= ((180/100)*F3)-1	0
4		= (180/100)*(A3-A2)	18	10		
5		= 360-SUM(B2:B4)	18	20	= 360-SUM(E2:E4)	
6			18	30		
7			18	40		
8			18	50		
9			18	60		
10			18	70		
11	M		18	80		
12	I		18	90		
13			9	100		
14						
15				Low		
16				Normal		
17				High		

Рис. 5.28. Данные для диаграммы спидометра

Нажмите сочетание клавиш Alt/Apple+~, чтобы отобразить формулы на рабочем листе. Можно воспользоваться командой Сервис ▸ Параметры ▸ Вид (Tools ▸ Options ▸ View) и установить флажок Формулы (Formulas), хотя это и более длинный путь.

Теперь выделите диапазон B2:B5 и запустите мастер диаграмм. На первом шаге мастера перейдите на вкладку Стандартные (Standard Types) (хотя она должна раскрываться по умолчанию). Затем в группе Тип (Chart Type) выберите вариант Кольцевая (Doughnut). Щелкните кнопку Далее (Next), чтобы перейти ко второму шагу мастера, и удостоверьтесь, что данные выводятся на диаграмму по строкам (в столбцах). Щелкните кнопку Далее (Next), чтобы перейти к шагу 3. Если необходимо, можно что-либо изменить на этом шаге, но для этого трюка настройка этого шага не обязательна. Щелкните кнопку Далее (Next), чтобы перейти к четвертому шагу, и удостоверьтесь, что диаграмма будет создана как объект на текущем рабочем листе (и снова это параметр по умолчанию). Создание диаграммы как объекта упростит работу с ней при настройке спидометра (рис. 5.29).

Выделите кольцевую диаграмму, медленно дважды щелкните самый большой сектор, чтобы выделить его, а затем щелкните его правой кнопкой мыши, в контекстном меню выберите команду Формат точки данных (Format Data Point)¹ и перейдите

¹ У автора Формат рядов данных (Format Data Series), но тогда мы не попадем в нужное меню. — Примеч. перев.

на вкладку Параметры (Options). Выберите угол поворота для этого сектора равным 90 градусам. Щелкните вкладку Вид (Patterns), выберите невидимую границу и прозрачную заливку, затем щелкните кнопку ОК. По очереди каждый из оставшихся секторов дважды медленно щелкните, затем дважды щелкните, чтобы открыть диалоговое окно **Формат элемента данных (Format Data Series)**, и выберите нужный цвет. Кольцевая диаграмма должна выглядеть, как на рис. 5.30.

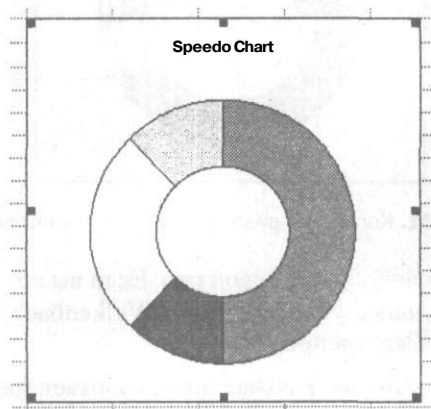


Рис. 5.29. Обычная кольцевая диаграмма

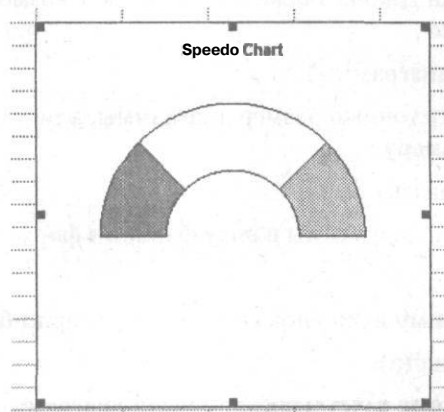


Рис. 5.30. Заготовка спидометра

Необходимо добавить еще один ряд (Series 2, Ряд 2) значений, чтобы создать циферблат. Снова выделите диаграмму, щелкните ее правой кнопкой мыши, в контекстном меню выберите команду **Исходные данные (Source Data)** и перейдите на вкладку Ряд (Series). Щелкните кнопку **Добавить (Add)**, чтобы создать новый ряд, и в поле **Значения (Values)** выберите диапазон C2:C13. Еще раз щелкните кнопку **Добавить (Add)**, чтобы добавить третий ряд (Series 3, Ряд 3), отвечающий за стрелку, и в поле **Значения (Values)** выберите диапазон E2:E5. Результат должен выглядеть, как на рис. 5.31.

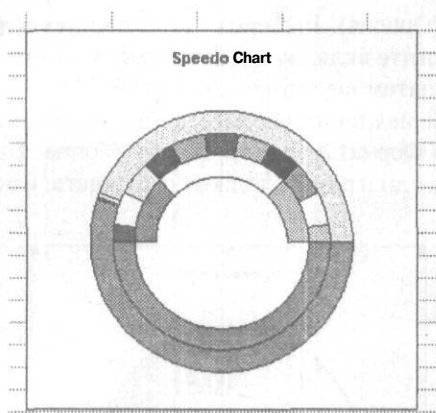


Рис. 5.31. Кольцевая диаграмма с несколькими рядами

Теперь спидометр начинает обретать свой вид. Если вы хотите добавить подписи, стоит загрузить специальные утилиты (John Walkenbach's Chart Tools) с сайта <http://j-walk.com/ss/excel/files/charttools.htm>.

Часть этой надстройки, которая, к **сожалению**, работает только в Windows, предназначена специально для меток данных. Она позволяет указывать диапазон на рабочем листе, на основе которого будут создаваться подписи данных для рядов диаграммы. Надстройка Джона также поддерживает возможности, перечисленные в следующем списке.

- Chart Size (Размер диаграммы).

Позволяет указывать точный размер диаграммы, а также приводить все диаграммы к одному размеру.

- Export (Экспортировать).

Позволяет сохранять диаграммы в виде файлов в формате .gif, .jpg, .tif и .png.

- Picture (Рисунок).

Преобразует диаграмму в рисунок (цветной или черно-белый).

- Text Size (Размер текста).

Фиксирует размер всех текстовых элементов диаграммы. Когда размер диаграммы **меняется**, текстовые элементы сохраняют свой размер.

- Chart Report (Отчет диаграммы).

Генерирует отчет для всех диаграмм или подробный отчет для одной диаграммы.

При помощи этой надстройки отформатируйте ряд Series 2 (Ряд 2), чтобы он отображал подписи данных, взятые из диапазона D2:D13. Не сбрасывая выделение Series 2 (Ряд 2), дважды щелкните его, чтобы открыть диалоговое окно Формат ряда данных (Format Data Series). Перейдите на вкладку Вид (Patterns) и выберите невидимую границу и прозрачную заливку. Диаграмма должна выглядеть, как на рис. 5.32.

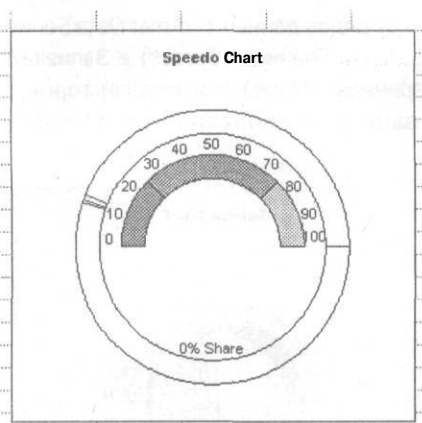


Рис. 5.32. Улучшенная диаграмма спидометра с добавленными подписями

Выделите ряд Series 3 (Ряд 3), затем щелкните его правой кнопкой мыши и в контекстном меню выберите команду Тип диаграммы (Chart Type). Выберите круговую диаграмму. Да, это выглядит немного странно (рис. 5.33). Но будьте уверены, если круговая диаграмма перекрывает кольцевую диаграмму, вы все сделали правильно.

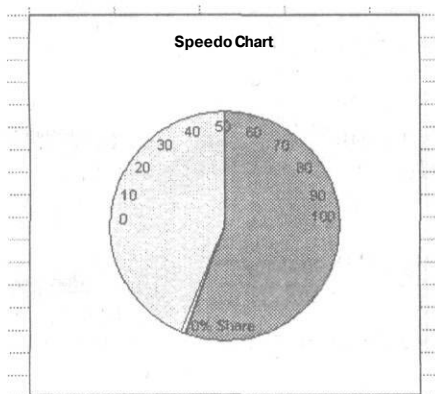


Рис. 5.33. Круговая диаграмма перекрывает диаграмму спидометра

Затем необходимо уменьшить размер круговой диаграммы, которую вы только что наложили на кольцевую диаграмму. Для этого разъедините ее и разберите на отдельные небольшие секторы. Выделите один из секторов круговой диаграммы (два медленных щелчка) и перетащите его наружу. Так вы разберете круговую диаграмму и уменьшите ее, как показано на рис. 5.34.

Теперь выделите всю круговую диаграмму, дважды щелкните ее, выберите команду Формат ряда данных (Format Data Series) и перейдите на вкладку Параметры (Options). Измените угол поворота первого сектора на 90 градусов. По очереди выделяйте все секторы круговой диаграммы, щелкайте их правой кнопкой мыши

и в диалоговом окне **Формат рядов данных (Format Data Series)** переходите на вкладку **Вид (Patterns)**. Для параметров **Граница (Border)** и **Заливка (Area)** выбирайте значение **Невидимая (или Прозрачная) (None)** для всех секторов, кроме третьего, который следует залить черным цветом. Вы получите диаграмму, показанную на рис. 5.35.

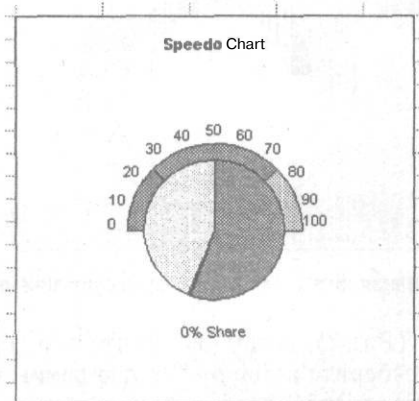


Рис. 5.34. Разобранная и уменьшенная круговая диаграмма

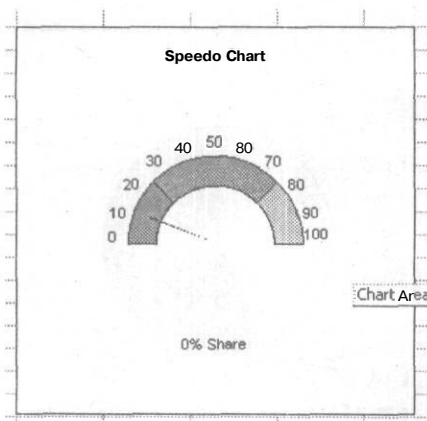


Рис. 5.35. Диаграмма спидометра, на которой окрашен только третий сектор круговой диаграммы

Чтобы добавить легенду, выделите диаграмму, щелкните ее правой кнопкой мыши, в контекстном меню выберите команду **Параметры диаграммы (Chart Options)** и перейдите на вкладку **Подписи данных (Data Labels)**. Установите флажок **Ключ легенды (Legend Key)**. Вы увидите спидометр (рис. 5.36). Теперь перемещайте диаграмму, изменяйте размер и редактируйте ее, как необходимо.

Теперь, когда диаграмма спидометра создана, нужно создать полосу прокрутки с панели инструментов **Элементы управления (Control Toolbox)** и связать полосу прокрутки и диаграмму.

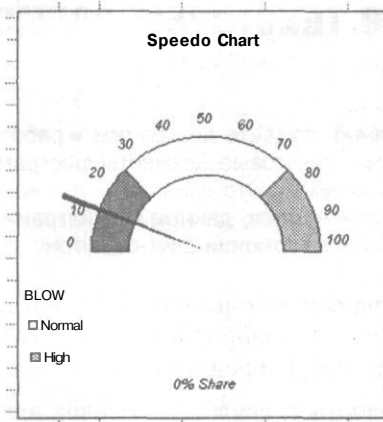


Рис. 5.36. Диаграмма спидометра с легендой

Для этого правой кнопкой мыши щелкните область панелей инструментов на экране — это верхняя область экрана, где расположены панели инструментов Стандартная (Standard) и Форматирование (Formatting), — и выберите команду Элементы управления (Control Toolbox). Выберите инструмент полосы прокрутки и перетащите его в нужное место на рабочем листе.

Выделите полосу прокрутки, щелкните ее правой кнопкой мыши и в контекстном меню выберите команду Свойства (Properties). Откроется диалоговое окно Свойства (Properties). В поле LinkedCell выберите ячейку F3, укажите максимальное значение 100 и минимальное значение 0. Закрыв это диалоговое окно и переместив полосу прокрутки на диаграмму, вы увидите приблизительно то же, что и на рис. 5.37.

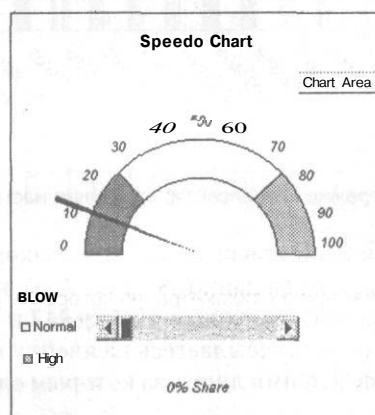


Рис. 5.37. Законченная диаграмма спидометра

Щелкая стрелки или перетаскивая ползунок на полосе прокрутки, вы будете изменять показания спидометра, но помните, что при этом также изменяются данные на связанном со спидометром рабочем листе.

Энди Поуп

Т Р Ю К
№ 58

Привязка текстовых элементов диаграмм к ячейкам

Если вы постоянно создаете диаграммы и работаете с ними, удобно привязать некоторые текстовые элементы диаграммы, например заголовки или подписи, к ячейкам. Это означает, что, когда соответствующие данные в ячейках изменяются, данные на диаграмме и ее текстовые элементы также остаются в гармонии друг с другом.

Текстовые элементы диаграммы, которые можно привязать к ячейке, — это заголовки диаграммы, заголовки основной и вспомогательной осей X, заголовки основной и вспомогательной осей Y и подписи рядов данных.

Чтобы увидеть, как это работает, свяжем заголовок диаграммы с ячейкой. Для начала задайте некоторые данные (рис. 5.38) и при помощи мастера диаграмм создайте обычную гистограмму. Теперь щелкните ячейку A17 и введите текст Age of Employees.

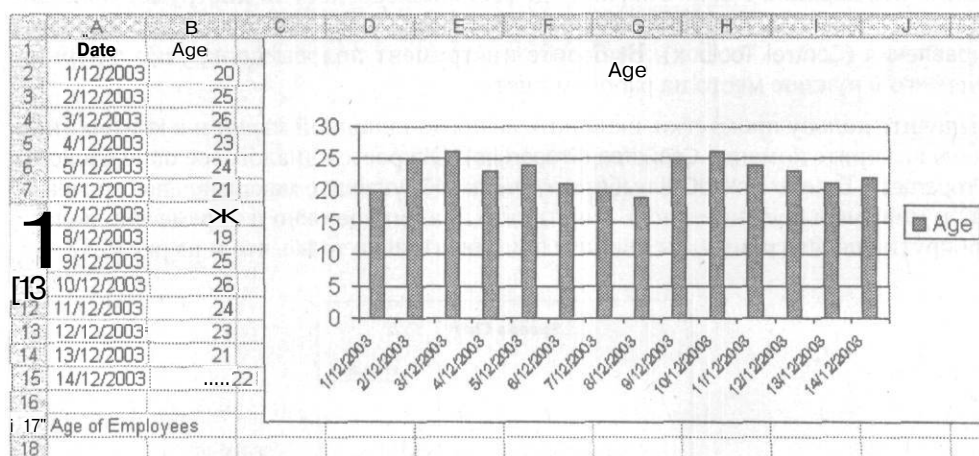


Рис. 5.38. Гистограмма с заголовком, созданная мастером диаграмм

Следующий шаг — установление связи между заголовком диаграммы и ячейкой. Выделите заголовок диаграммы (в данном случае Age), перейдите в строку формул, введите = (знак равенства), щелкните ячейку A17 и нажмите клавишу Enter. (Обратите внимание, что, если вы ссылаетесь на ячейку на другом листе, в ссылке на ячейку необходимо ввести имя листа, за которым следует восклицательный знак (!).)

Тот же способ работает для подписей данных, но необходимо выделять отдельные подписи данных перед тем, как связывать их с ячейками. Результат должен выглядеть, как на рис. 5.39.

Этот хитрый трюк позволит текстовым элементам и данным диаграммы оставаться в полной гармонии. Кроме того, он экономит время при создании диаграмм.

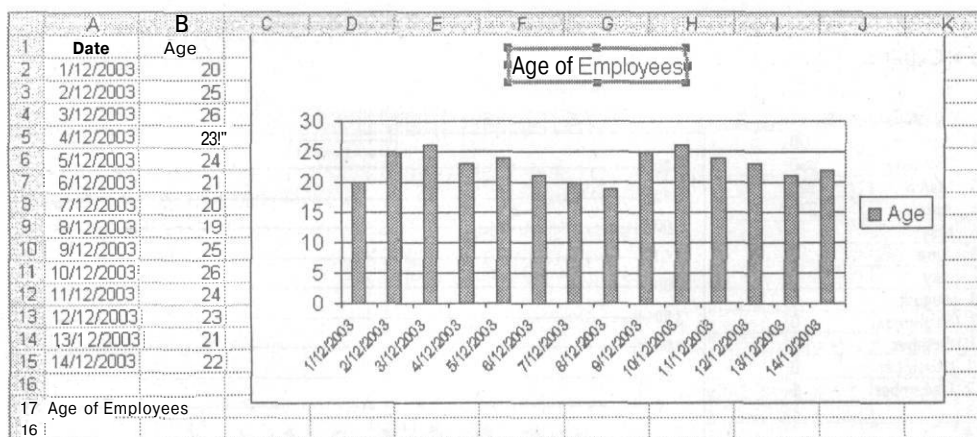


Рис. 5.39. Гистограмма, заголовок которой связан с ячейкой

Энди Поуп

Т Р Ю К
№ 59

Трюки с данными диаграммы, чтобы пустые ячейки не учитывались при построении графика

Excel считает, что пустые ячейки содержат значение 0, из-за чего могут получиться довольно уродливые диаграммы. График неожиданно уменьшается, и вы остаетесь с диаграммой, которая не может точно отразить картину, которую нужно описать. Благодаря трюкам этого раздела можно гарантировать, что пустые ячейки не будут учитываться при построении графика.

Можно предотвратить построение графика по пустым ячейкам двумя простыми способами: скрыв строки или столбцы или заставив ячейки возвращать значение #N/A.

Скрытые строки или столбцы

Введите в таблицу данные (рис. 5.40), создайте график и посмотрите, на что он похож, если при построении учитывались нулевые значения. Выделите диапазон A1:B12, запустите мастер диаграмм и создайте график как объект на текущем рабочем листе.

На диаграмме рис. 5.40 выводятся данные о зарплате по месяцам. Это означает, что, если сейчас апрель, на диаграмме будут отражены восемь нулевых значений, так как данные для месяцев с мая по декабрь пока что недоступны.

Чтобы избежать этого, скройте строки с 5 по 12 (May:Dec). Excel не учитывает скрытые строки при построении графика, поэтому вы получите результат, как на

рис. 5.41. Чтобы скрыть строки, выделите их и выберите команду **Формат** ▶ **Строка** ▶ **Скрыть** (Format ▶ Rows ▶ Hide).

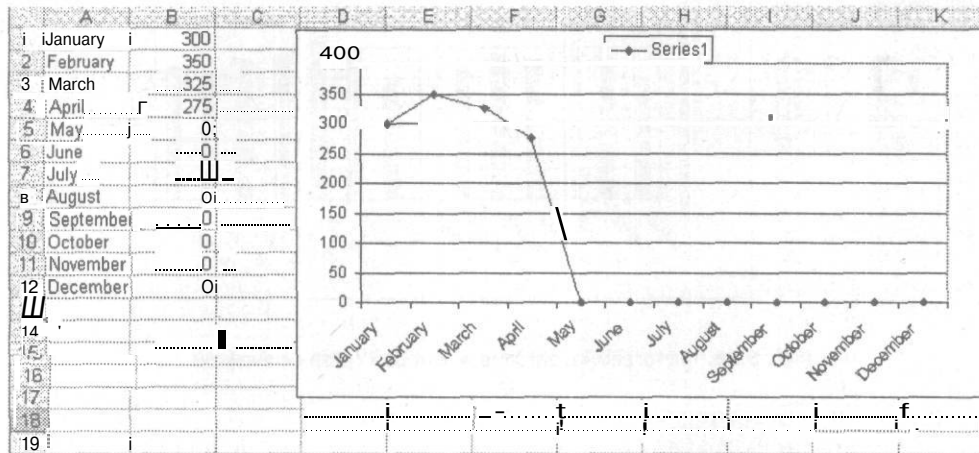


Рис. 5.40. Данные на графике с учетом нулевых значений

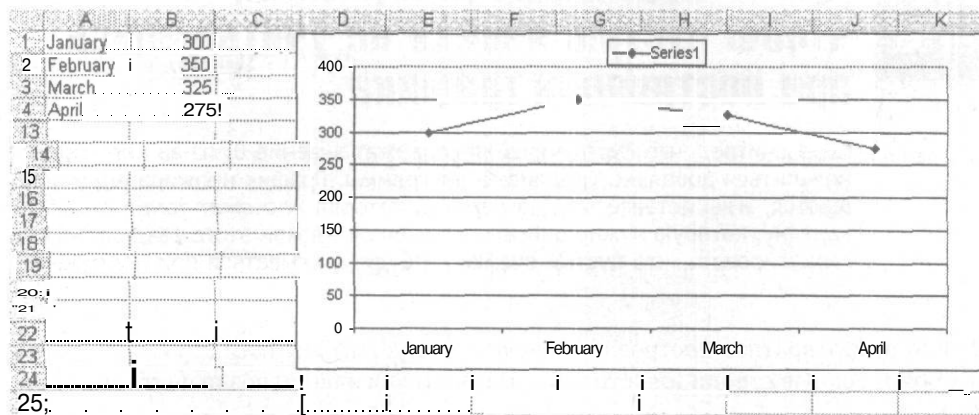


Рис. 5.41. Диаграмма без нулевых значений

Трюки с формулами и функциями

Трюки с 60 по 80

Формулы и функции — это логика, лежащая в основе мощи электронных таблиц. Управление программной логикой — это всегда вызов умениям пользователя, но отслеживать программную логику во множестве ячеек, листов и рабочих книг может быть сложно, особенно по мере роста и использования таблиц. Кроме того, возможности формул и функций, встроенных в Excel, не всегда точно отвечают требованиям, еще более усложняя ситуацию. К счастью, существует множество способов, помогающих содержать формулы и функции в порядке.



Добавление описаний к формулам

Именованные диапазоны Excel и примечания к ячейкам могут помочь в понимании формул, но иногда необходимо поместить описание в саму формулу. При помощи трюков этого раздела вы сможете быстро и просто добавить описания в формулы.

Даже если вы сами написали разнообразные формулы и функции, возвращаясь к ним через некоторое время, вы зачастую понимаете, что необходимо напряженно вчитываться в ссылки на ячейки, чтобы понять, что же эти формулы делают. Было бы здорово, если бы в конце формулы можно было просто добавить какой-то текст, который не влиял бы на результат, но дал бы необходимую информацию о формуле.

Проблема, конечно же, заключается в том, что, когда вы добавляете или внедряете текст в часть формулы, результат получается уже не числовой и его нельзя использовать в дальнейших вычислениях. Однако в Excel предусмотрена одна малоизвестная функция, при помощи которой можно добавлять текст в формулы и функции.

Скажем, в ячейке А1 находится следующая формула:

```
=SUM($A$1:$A$10)*$B$1
```

В русской версии Excel

```
=СУММ($A$1:$A$10)*$B$1
```

Предположим, в диапазоне \$A\$1:\$A\$10 содержатся различные числовые результаты, представляющие итоги для определенных месяцев, а в ячейке \$B\$1 — какое-то

процентное значение, представляющее налоговую ставку. Вы можете добавить текст с описанием в формулу, применив для этого функцию Excel Ч (N):

```
=SUM($A$1:$A$10,N("Values for April"))*$B$1+N("Tax Rate for April")
```

В русской версии Excel

```
=СУММ($A$1:$A$10,Ч("Values for April"))*$B$1+Ч("Tax Rate for April")
```

Теперь вы сможете понять, для чего используется формула, выделив эту ячейку и взглянув на строку формул. Функция Ч (N) для любого текста всегда возвращает значение 0, поэтому никоим образом не влияет на результат формулы.

Т Р Ю К
№ 61

Перемещение относительных формул без изменения ссылок

В Excel ссылка в формуле может быть либо относительной, либо абсолютной. Иногда, однако, возникает необходимость воспроизвести те же формулы в другом месте на рабочем листе или в рабочей книге, или же на другом листе.

Если формулу нужно сделать абсолютной, введите \$ (знак доллара) перед буквой столбца или номером строки в ссылке на ячейку, например, \$A\$1. Когда вы сделаете это, неважно, куда вы скопируете формулу, она все так же будет ссылаться на те же ячейки.

Иногда, однако, можно ввести много формул, содержащих не абсолютные, а относительные ссылки. Обычно это делается для того, чтобы, если вы скопируете исходную ячейку с формулой вниз или вбок, ссылка на строку или столбец изменилась соответствующим образом.

Кроме того, иногда формулы вводят, используя и относительные, и абсолютные ссылки, желая воспроизвести те же формулы в другом диапазоне на том же рабочем листе, на другом листе в той же рабочей книге или, возможно, на другом листе другой рабочей книги. Это можно сделать, не изменяя ссылки на диапазоны внутри формул.

Выделите диапазон ячеек, который хотите скопировать. Выберите команду Правка ▶ Заменить (Edit ▶ Replace) и в поле Что (Find What) введите = (знак равенства). В поле Заменить на (Replace With) введите @ (знак коммерческого at) или любой другой символ который, вы уверены, не используется ни в одной формуле. Щелкните на кнопке Заменить все (Replace All).

Во всех формулах на вашем рабочем листе вместо знака равенства будет стоять символ @.

Теперь вы просто можете скопировать этот диапазон, вставить его на нужное место, выделить и выбрать команду Правка ▶ Заменить (Edit ▶ Replace). На этот раз замените символ @ на = (знак равенства). Скопированные формулы будут ссылаться на те же ячейки, что и исходные.

ТРЮК
№ 62

Сравнение двух диапазонов Excel

Выявление различий между двумя большими таблицами может занять очень много времени. К счастью, есть, по меньшей мере, два способа автоматизировать задачу, которая в противном случае была бы скучной ручной работой.

Способы, которые мы будем применять в этом разделе, идентичны способам, с которыми мы уже встречались раньше, когда получали обновленную копию электронной таблицы и нам нужно было определить ячейки обновленной копии, отличающиеся от ячеек в оригинале. Оба способа экономят часы однообразной ручной проверки и, что еще важнее, исключают вероятность появления ошибок.

Для следующих примеров мы заранее скопировали новые данные на тот же лист, где находятся исходные. На рис. 6.1 показано, как данные выглядят в этих двух диапазонах. Обратите внимание, что для упрощения восприятия мы выделили ячейки таблицы 2, отличающиеся от соответствующих ячеек в таблице 1, жирным шрифтом.

	A	B	C
1	Name	Age	Area
2	Bill	22	1a
3	Joe	33	1a
4	Frank	55	2c
5	Brad	48	3d
6	Mary	29	3d
7	Anne	45	4a
8			
9	Name	Age	Area
10	Bill	22	2c
11	Joe	33	1a
12	Frank	55	2d
13	Brad	43	3d
14	Dave	29	3d
15	Anne	45	4a

Рис. 6.1. Диапазоны для сравнения

Способ 1. Использование значений Истина и Ложь

В первом способе нужно ввести простую формулу в другой диапазон того же размера и формы. Удобно то, что можно добавить формулу за один шаг, не копируя и не вставляя данные.

Чтобы сравнить диапазоны, показанные на рис. 6.1, выделите диапазон E1:G7, начиная с ячейки E1. Это гарантирует, что ячейка E1 будет активной ячейкой выделения. Выделив диапазон, шелкните строку формул и введите следующую формулу:

=A1=A9

ПРИМЕЧАНИЕ

Написав формулу, нажмите сочетание клавиш **Ctrl+Enter**. Так вы введете формулу с относительными ссылками в каждую ячейку выделенного диапазона. Это стандартный способ ввода формулы в массив ячеек с соответствующим изменением ссылок.

Диапазон E1:G7 будет заполнен значениями Истина (True) — такое же значение, и Ложь (False) — значение отличается.

Если два набора данных находятся на разных рабочих листах, для хранения значений Истина/Ложь (True/False) можно использовать третий рабочий лист, введя формулу способом для массивов ячеек. Например, предполагая, что вторая таблица данных находится на листе Sheet2 и начинается с ячейки A9, а исходная таблица данных хранится на листе Sheet1 и начинается с ячейки A1, на третьем рабочем листе введите следующую формулу массива:

```
=Sheet1!A1=Sheet2!A9
```

При работе с большими объемами данных бывает полезно уменьшить масштаб листа.

ПРИМЕЧАНИЕ

Чтобы удалить формулу, введенную способом для массивов, необходимо выделить и удалить весь диапазон. Часть его удалить невозможно.

Способ 2. Условное форматирование

Второй способ предпочтительнее, поскольку при его использовании проще внести изменения после того, как сравнение выполнено. Однако для применения этого способа оба набора данных должны находиться на одном рабочем листе, чтобы можно было просто копировать и вставлять данные.

Снова предполагая, что мы сравниваем два предыдущих диапазона, выделите диапазон A1:C7, начиная с ячейки A1. Это гарантирует, что ячейка A1 будет активной ячейкой выделения.

Выделив диапазон, выберите команду **Формат** ► **Условное форматирование** (Format ► Conditional Formatting). Выберите пункт **Формула** (Formula Is) и введите следующую формулу:

```
=NOT(A1=A9)
```

В русской версии Excel

```
=НЕ(A1=A9)
```

Щелкните кнопку **Формат** (Format) (рис. 6.2) и выберите форматирование, которым будут выделены отличающиеся данные.

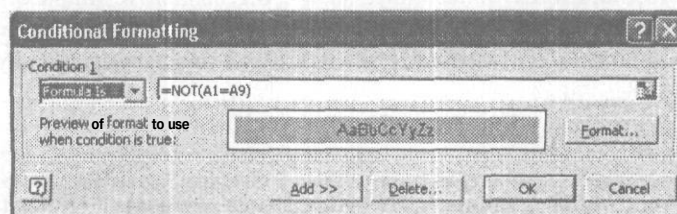


Рис. 6.2. Диалоговое окно условного форматирования

Щелкните на кнопке ОК, и все различия будут отформатированы согласно выбранному вами формату.

Если вы внесете какие-либо изменения в данные, то при одинаковом содержимом ячеек в обеих таблицах будет восстановлено обычное форматирование.

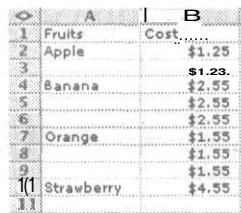
Т Р Ю К
№ 63

Заполнение всех пустых ячеек в списке

Часто пользователи оставляют ячейки пустыми, если данные в этих ячейках такие же, как и в ячейках выше. Это упрощает чтение списков, но ухудшает структуру данных. При помощи трюков этого раздела можно быстро и просто заполнить все пустые ячейки в списке.

Многие функции Excel предназначены для работы со списками. Чтобы обеспечить правильную работу большинства из них, списки не должны содержать пустые ячейки, а форматирование заголовков столбцов должно отличаться от форматирования данных в списке.

При настройке данных в Excel полезно убеждаться, что заняты все ячейки в списке. Однако часто списки задаются, как показано на рис. 6.3.



	A	B
1	Fruits	Cost
2	Apple	\$1.25
3		\$1.25
4	Banana	\$2.55
5		\$2.55
6		\$2.55
7	Orange	\$1.55
8		\$1.55
9		\$1.55
10	Strawberry	\$4.55
11		

Рис. 6.3. Список с пустыми ячейками на рабочем листе

Хотя в столбце Cost цены повторяются, виды фруктов в столбце Fruits — нет. Как упоминалось в начале главы 1, такая организация данных создаст много проблем при использовании таких средств, как промежуточные итоги (Subtotals) и сводные таблицы (PivotTables). В большинстве случаев Excel ожидает, что связанные данные помещены в непрерывный список или таблицу без пустых ячеек.

Есть два способа заполнения пустых ячеек в списке: при помощи формулы и при помощи макроса.

Способ 1. Заполнение пустых ячеек при помощи формулы

Предположим, есть список записей в столбце A, например, как в предыдущем примере, и в этом списке есть пустые ячейки. Вот как быстро и просто можно заполнить эти ячейки значениями из ячеек над ними.

Выделите все данные в столбце A, выберите команду Правка ▶ Перейти (Edit ▶ Go To) (сочетание клавиш Ctrl+G) и щелкните кнопку Выделить (Special). Установите

переключатель-флажок Пустые ячейки (Blanks) и щелкните кнопку ОК. Теперь в списке выделены только пустые ячейки. Нажмите клавишу = (клавишу знака равенства), а затем стрелку вверх. Удерживая клавишу Ctrl, нажмите клавишу Enter.

Можно быстро преобразовать формулы в значения, выделив столбец A целиком. Выберите команду Правка ▸ Копировать (Edit ▸ Copy) (сочетание клавиш Ctrl+C), затем выберите команду Правка ▸ Специальная вставка (Edit ▸ Paste Special), установите переключатель Значения (Values) и щелкните на кнопке ОК.

Способ 2. Заполнение пустых ячеек при помощи макроса

Если вам часто приходится заполнять пустые ячейки, лучше делать это при помощи макроса. Чтобы создать его, выберите команду Сервис ▸ Макрос ▸ Редактор Visual Basic (Tools ▸ Macro ▸ Visual Basic Editor) (сочетание клавиш Alt/Option+F11), затем выберите команду Insert ▸ Module и введите код из листинга 6.1.

Листинг 6.1

```
Sub FillBlanks()
Dim rRange1 As Range. rRange2 As Range
Dim lReply As Integer

If Selection.Cells.Count = 1 Then
MsgBox "нужно выделить лист и включить пустые ячейки", _
vbInformation, "OzGrid.com"
Exit Sub
ElseIf Selection.Columns.Count > 1 Then
MsgBox "нужно выделить только один столбец". _
vbInformation, "OzGrid.com"
Exit Sub
End If

Set rRange1 = Selection

On Error Resume Next
Set rRange2 = rRange1.SpecialCells(xlCellTypeBlanks)
On Error GoTo 0

If rRange2 Is Nothing Then
MsgBox "пустые ячейки не найдены", _
vbInformation, "OzGrid.com"
Exit Sub
End If

rRange2.FormulaR1C1 = "=R[-1]C"

lReply = MsgBox("преобразовать в значения", vbYesNo + vbQuestion, _
"OzGrid.com")

If lReply = vbYes Then rRange1 = rRange1.Value
End Sub
```

Закончив, закройте окно, чтобы вернуться в Excel, и сохраните рабочую книгу. Теперь выберите команду Сервис ► Макрос ► Макросы (Tools ► Macro ► Macros) (сочетание клавиш Alt/Option+F8), выберите Fill Blanks и щелкните кнопку Выполнить (Run), чтобы выполнить макрос, или кнопку Параметры (Options), чтобы назначить клавишное сочетание.

**ТРЮК
№ 64**

Как сделать, чтобы формула ссылалась на следующие строки при копировании по столбцам

Автоматическое приращение ссылок на ячейки в Excel в большинстве случаев работает прекрасно, но иногда нужно изменить принцип работы этого средства.

Если создать ссылку на одну ячейку, например, A1, а затем скопировать ее вправо по столбцам, то в результате эта ссылка будет меняться: =B1, =C1, =D1 и т. д., но на самом деле вы хотите получить другой результат. Нужно, чтобы приращение формулы шло по строкам, а не по столбцам, то есть =A1, =A2, =A3 и т. д.

К сожалению, в Excel такая возможность не предусмотрена. Но можно обойти это ограничение с помощью функции ДВССЫЛ (INDIRECT), в которую вложена функция АДРЕС (ADDRESS).

Наверное, лучший способ объяснить, как создать нужную функцию, — использовать пример с предсказуемыми результатами. В ячейках A1:A10 по порядку введите числа от 1 до 10. Выделите ячейку D1 и введите в ней следующую формулу: =INDIRECT(ADDRESS(COLUMN()-3,1))

В русской версии Excel

=ДВССЫЛ(АДРЕС(СТОЛБЕЦ()-3,1))

Когда вы сделаете это, в ячейке D1 должно появиться число 1. Это происходит потому, что формула ссылается на ячейку A1.

Если вы скопируете эту формулу в столбец справа, то в ячейке E1 появится число 2. Другими словами, хотя вы копируете формулу по столбцам, приращение ссылки в формуле идет по строкам (рис. 6.4).

ПРИМЕЧАНИЕ

Этот способ особенно полезен, если в электронной таблице по строкам одного столбца записаны заголовки, а вы хотите создать динамические ссылки на эти заголовки строк в других столбцах.

Если продолжить копировать эту формулу вправо по столбцам, в ячейке F1 появится число 3, в ячейке G1 — число 4 и т. д. Это достаточно понятный процесс, если вы ссылаетесь только на одну ячейку. Однако часто приходится ссылаться на диапазон ячеек, который используется как аргумент функции.

	A	a	C	D	E	F
1	1			1	2	
2	2					
3	3					
4	4					
X!	5					
6	6					
7	7					
8	8					
9	9					
10	10					
11						

Рис. 6.4. Результат копирования ячейки D1 в ячейку E1

Чтобы продемонстрировать, о чем идет речь, будем использовать необходимую всем функцию СУММ (SUM). Предположим, вы получили длинный список чисел, и необходимо просуммировать столбцы чисел в режиме промежуточной суммы:

=SUM(\$A\$1:\$A\$2), =SUM(\$A\$1:\$A\$3), =SUM(\$A\$1:\$A\$4)

В русской версии Excel

=СУММ(\$A\$1:\$A\$2), =СУММ(\$A\$1:\$A\$3), =СУММ(\$A\$1:\$A\$4)

Возникает проблема, так как результат должен быть динамическим и распространяться по 100 столбцам всего в одной строке, а не вниз по 100 строкам в одном столбце (как это часто бывает).

Конечно же, вы могли бы вручную ввести нужные функции в отдельные ячейки, но на это потребовалось бы огромное количество времени. Вместо этого можно применить тот же принцип, что и ранее, при ссылке на одну ячейку.

Заполните диапазон A1:A100 числами от 1 до 100 по порядку. В ячейку A1 введите 1, выделите ячейку A1 и, удерживая клавишу Ctrl, щелкните маркер заполнения и протащите его вниз на 100 строк.

Выделите ячейку D1 и введите следующую формулу:

=SUM(INDIRECT(ADDRESS(1,1)&" ":"&ADDRESS(COLUMN()-2,1)))

В русской версии Excel

=СУММ(ДВССЫЛ(АДРЕС(1,1)&" ":"&АДРЕС(СТОЛБЕЦ()-2,1)))

В результате вы получите 3 как сумму ячеек A1:A2. Скопируйте эту формулу в ячейку E1, в результате получите 6, то есть сумму ячеек A1:A3. Скопируйте ее в ячейку F1, получите 10 — сумму ячеек A1:A4 (рис. 6.5).

Переменная функция СТОЛБЕЦ (COLUMN) заставляет последнюю ссылку на ячейку увеличиваться на 1 каждый раз, когда вы копируете формулу в новый столбец. Функция СТОЛБЕЦ (COLUMN) всегда возвращает номер столбца (а не букву) в ту ячейку, в которой находится, если только вы не ссылаетесь на другую ячейку.

Иначе, можно было бы воспользоваться средством Excel Специальная вставка ► Транспонировать (Paste Special ► Transpose). Введите формулу =SUM(\$A\$1:\$A2) в ячейку B1 (обратите внимание на относительную ссылку на строку и абсолютную ссылку на столбец в ссылке \$A2) и скопируйте эту формулу вниз до ячейки

B100. Выделив диапазон B2:B100, скопируйте его, выделите ячейку D1 (или любую другую, справа от которой есть 100 или более столбцов) и выберите команду Правка ▶ Специальная вставка ▶ Транспонировать (Edit ▶ Paste Special ▶ Transpose). Если необходимо, можно удалить формулы в диапазоне B2:B100.

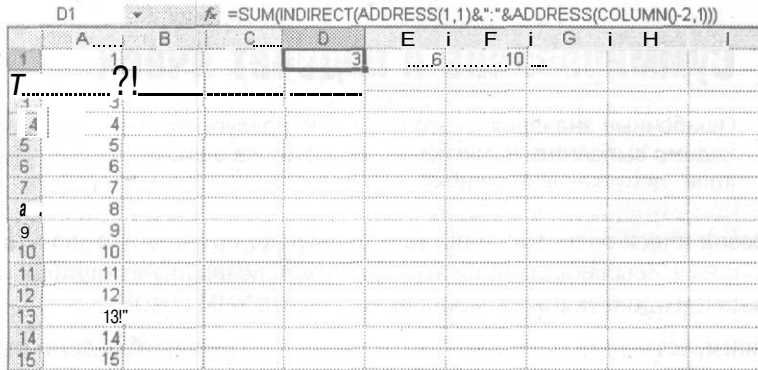


Рис. 6.5. Результат копирования ячейки D1 вправо до ячейки F1

ТРЮК
№ 65

Преобразование дат в отформатированные даты Excel

Даты, импортированные из других программ, часто вызывают проблемы в Excel. Многие вручную их перепечатывают, но есть и более простые способы решения проблемы.

Взглянем на некоторые форматы даты, с которыми можно встретиться, и узнаем, как преобразовать их в стандартные даты Excel.

Как демонстрирует рис. 6.6, формулы в столбце B преобразуют даты в столбце A в результаты в столбце C, причем в столбце C даты записаны в стандарте для США mm/dd/yy (мм/дд/гг).

	A	B	C
1	Old Date	Formula Used	Formula Result
2	112303	=VALUE(LEFT(A2,2)&"/"&MID(A2,3,2)&"/"&RIGHT(A2,2))	11/23/03
3	031123	=VALUE(MID(A3,3,2)&"/"&RIGHT(A3,2)&"/"&LEFT(A3,2))	11/23/03
4	231103	=VALUE(MID(A4,3,2)&"/"&LEFT(A4,2)&"/"&RIGHT(A4,2))	11/23/03

Рис. 6.6. Разные форматы даты, преобразованные в допустимые даты в формате для США

	A	B	C
1	Old Date	Formula Used	Formula Result
2	112303	=VALUE(MID(A2,3,2)&"/"&LEFT(A2,2)&"/"&RIGHT(A2,2))	23/11/2003
3	31123	=VALUE(RIGHT(A3,2)&"/"&MID(A3,3,2)&"/"&LEFT(A3,2))	23/11/2003
4	231103	=VALUE(LEFT(A4,2)&"/"&MID(A4,3,2)&"/"&RIGHT(A4,2))	23/11/2003

Рис. 6.7. Разные форматы даты, преобразованные в допустимые даты в формате для Европы

На рис. 6.7 продемонстрирован тот же подход, но ячейки в столбце С отформатированы согласно европейскому формату даты dd/mm/yy (дд/мм/гг).

Надеемся, эти ловкие формулы помогут упростить работу с импортированными датами.

**ТРЮК
№ 66**

Суммирование и подсчет ячеек

Ошибочные значения — это полезные предупреждения, но иногда необходимо выполнить вычисления, несмотря на ошибки. В этом помогут функции, допускающие ошибки.

Если диапазон ячеек содержит одно или несколько ошибочных значений, большинство формул, ссылающихся на этот диапазон, возвращают ошибку. Эту проблему можно преодолеть при помощи функции БДСУММ (DSUM).

Предположим, есть длинный список чисел, для которых необходимо вычислить общую сумму. Но одна из ячеек по какой-то причине возвращает ошибку #N/A.

Введите некоторые данные (рис. 6.8).

	A	B
1	Numbers1	Numbers2
2	#N/A	#N/A
3	1	1
4	2	2
5	3	3
6	4	4
7	5	5
8	6	6
9	7	7
10	8	8
11	9	9
12		

Рис. 6.8. Данные, которые будут выдавать сообщение об ошибке #N/A

Чтобы генерировать ошибку #N/A, введите формулу =#N/A() в ячейки A2 и B2. В ячейке A12 находится стандартная функция СУММ (SUM), суммирующая ячейки A2:A11, но, так как ячейка A2 возвращает ошибку #N/A, функция СУММ (SUM) также возвращает ошибку #N/A. Диапазону D1:D2 присвоено имя Criteria и он используется как последний аргумент функции БДСУММ (DSUM) в ячейке B12.

Синтаксис функции БДСУММ (DSUM) (и всех остальных функций баз данных)
=DSUM(database,field,criteria)

В русской версии Excel

=БДСУММ(база_данных,поле,критерий)

Аргумент база_данных (database) — это диапазон ячеек, формирующий список или базу данных. В диапазоне базы данных строки связанной информации считаются

записями, а столбцы данных считаются полями. Первая строка содержит подписи для всех столбцов.

Аргумент поле (field) определяет столбец, используемый функцией. Столбец может быть задан по имени (по подписям, которые находятся наверху столбцов) или по позиции. Позиция первого столбца равна 1, четвертого — 4 и т. д.

Аргумент критерий (criteria) определяет диапазон ячеек, содержащих условия. Диапазон, используемый для критерия, должен включать, по меньшей мере, одно название столбца и, по меньшей мере, одну ячейку под названием столбца, в которой указано условие для столбца.

Если данные, которые нужно суммировать, могут содержать ошибки различного типа, может потребоваться функция БДСУММ (DSUM) с набором критериев для всех возможных ошибок. Однако всегда лучше устранить источник ошибки, чем пытаться обойти ее.

Чтобы обойти ошибку, необходимо снова применить функцию БДСУММ (DSUM), но на этот раз указать критерий, охватывающий четыре столбца, расширив имеющийся диапазон для критерия на ячейки D1:G2 при помощи команды Вставка ▶ Имя ▶ Присвоить (Insert ▶ Name ▶ Define) (рис. 6.9).

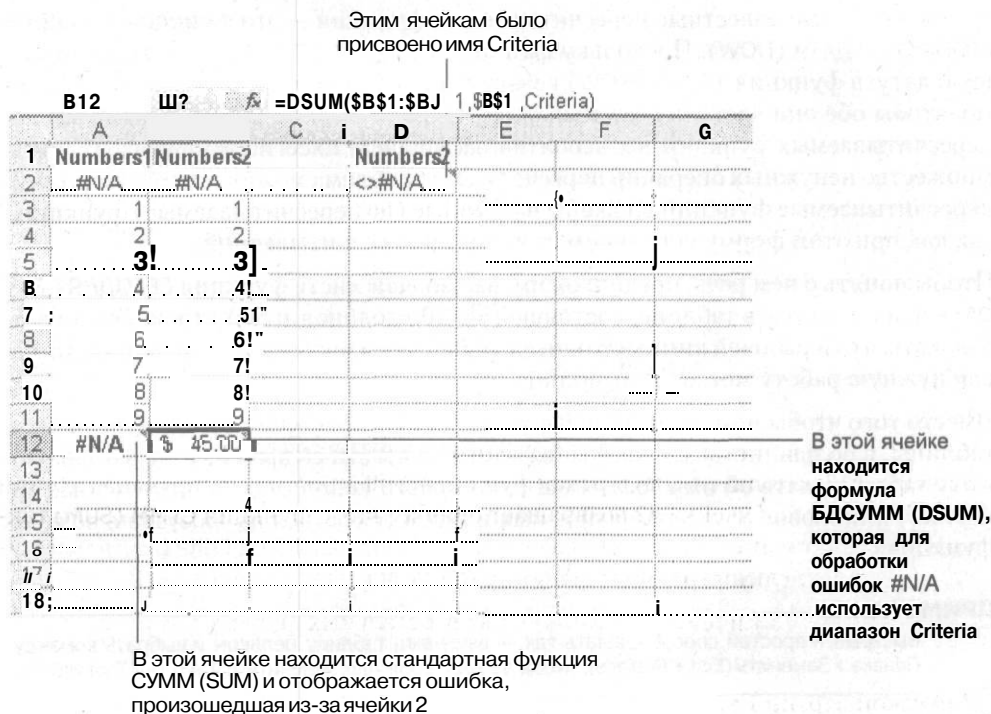


Рис. 6.9. Функция БДСУММ (DSUM) применяется для обработки различных ошибок

В Excel предусмотрен широкий набор функций баз данных, и подобным образом можно применять любую. Попробуйте использовать тот же метод для функций СЧЁТ

(DCOUNT), БСЧЁТА (DCOUNTA), ДМАКС (DMAX), ДМИН (DMIN), БДПРОИЗВЕД (DPRODUCT) и т. д.



Как уменьшить влияние переменных функций на скорость пересчета

Пересчитываемые функции, которые должны пересчитываться практически после каждого действия пользователя в Excel, могут потребовать огромного количества времени. Хотя пересчитываемые функции слишком полезны, чтобы полностью отказываться от них, всегда есть способ сократить вызываемые ими задержки.

Пересчитываемая функция — это просто функция, которая пересчитывается каждый раз, когда в Excel производится какое-либо действие, например, ввод данных, изменение ширины столбцов и т. д. Одно из немногих действий, которое не вызывает пересчета переменной функции, — это изменение форматирования ячеек, если только вы не делаете это при помощи средства Специальная вставка ▶ Форматы (Paste Special ▶ Formats).

Вероятно, самые известные пересчитываемые функции — это функции СЕГОДНЯ (TODAY) и ТДАТА (NOW). Поскольку функция СЕГОДНЯ (TODAY) возвращает текущую дату, а функция ТДАТА (NOW) возвращает текущие дату и время, необходимо, чтобы обе они часто пересчитывались. Однако если на рабочем листе много пересчитываемых функций, вы, вероятно, заставляете Excel постоянно выполнять множество ненужных операций пересчета. Эта проблема может усугубиться, если пересчитываемые функции вложены в обычные (не пересчитываемые) функции, так как при этом формула целиком становится пересчитываемой.

Чтобы понять о чем речь, предположим, на рабочем листе функция СЕГОДНЯ (TODAY) используется в таблице, состоящей из 20 столбцов и 500 строк. Это будет означать, что в рабочей книге находится 10 000 пересчитываемых функций, тогда как нужную работу могла бы выполнить только одна.

Вместо того чтобы помещать 10 000 функций СЕГОДНЯ (TODAY) в каждую ячейку таблицы, в большинстве случаев можно просто ввести ее в ячейку, не входящую в эту таблицу, дать ей имя TodayDate (или просто использовать идентификатор ячейки) или любое другое понятное имя и затем сослаться на TodayDate во всех функциях.

ПРИМЕЧАНИЕ

Быстрый и простой способ сделать так — выделить таблицу целиком и выбрать команду Правка ▶ Заменить (Edit ▶ Replace), чтобы во всех формулах заменить TODAY() на TodayDate.

Теперь у вас будет всего лишь одна функция СЕГОДНЯ (TODAY) вместо 10 000.

В качестве другого примера предположим, что первые 500 строк столбца В заполнены относительной формулой, например, =СЕГОДНЯ()-A1 (=TODAY()-A1), а в пер-

вых 500 строках столбца А находится дата, относящаяся к прошедшему периоду. Каждый раз, когда вы производите любое действие в Excel, вы заставляете Excel пересчитывать пересчитываемую функцию СЕГОДНЯ (TODAY) в 499 раз больше, чем необходимо! Поместив функцию СЕГОДНЯ (TODAY) в любую ячейку и присвоив ячейке имя TodayDate (или любое другое, удобное вам), вы можете просто использовать формулу =TodayDate-A1. Теперь Excel придется пересчитать только одну функцию СЕГОДНЯ (TODAY), что существенно улучшит производительность.

Т Р Ю К
№68

Подсчет только по одному вхождению каждой записи в списке

Если у вас есть большой список элементов, вы, возможно, захотите подсчитать элементы без учета их повторных вхождений в список. При помощи этого трюка вы сможете сосчитать каждую запись только один раз.

Предположим, есть список (рис. 6.10), отсортированный, чтобы было проще заметить повторяющиеся записи.

	... A
1	Names
2	Bill W
3	Bill W
4	Bob G
5	Dave H.....
6	Dave H
7	Fran T.....
8	Fran T.....
9	Frank W
10	Frank W
11	George H
12	Harry O.....
13	Mark W.....
14	Mary O
15	Mary O
16	Peter G.....
17	Raina H.....

Рис. 6.10. Диапазон отсортированных имен

Если подсчитать элементы этого списка обычным образом, используя функцию СЧЁТЗ (COUNTA), получится по несколько вхождений имен Bill W, Dave H, Frank T, Frank W и Mary O. Функция БСЧЁТА (DCOUNTA) предлагает альтернативный вариант, очень эффективный и легко настраиваемый.

Синтаксис функции БСЧЁТА (DCOUNTA)

DCOUNTA(database,field,criteria)

В русской версии Excel

БСЧЁТА(база_данных,поле,критерий)

ПРИМЕЧАНИЕ

У этой функции те же аргументы, что и у функции **БДСУММ (DSUM)**, описанной в разделе «Трюк № 65. Преобразование дат в отформатированные даты Excel».

Создав предыдущий список, в ячейке D1 введите слово Criteria (или любой другой заголовок, не совпадающий с заголовком поля или столбца). Под ним, в ячейке D2, введите формулу

=COUNTIF(\$A\$2:A2,A2)=1

В русской версии Excel

=СЧЁТЕСЛИ(\$A\$2:A2,A2)=1

Обратите внимание на комбинацию относительных (A2) и абсолютных (\$A\$2) ссылок! Это необходимо для того, чтобы критерий правильно работал.

Теперь в ячейке, где должен оказаться результат, введите следующую функцию:

=DCOUNTA(\$A\$1:\$A\$100,1,\$D\$1:\$D\$2)

В русской версии Excel

=БСЧЁТА(\$A\$1:\$A\$100,1,\$D\$1:\$D\$2)

Она будет применять критерий для исключения дубликатов и выдаст вам необходимый результат, равный 11, так как в списке 11 уникальных имен.

**Т Р Ю К
№ 69**

Суммирование каждой второй, третьей или n-й строки или ячейки

Часто бывает, что нужно суммировать каждую вторую, третью, четвертую и т. д. ячейку в электронной таблице. Теперь, благодаря следующему трюку, это можно сделать.

В Excel не предусмотрено стандартной функции, которая может суммировать каждую n-ю ячейку или строку. Однако можно выполнить эту задачу несколькими различными способами. Все эти подходы основаны на функциях **СТРОКА (ROW)** и **ОСТАТ (MOD)**.

Функция **СТРОКА (ROW)** возвращает номер строки для заданной ссылки на ячейку:

ROW(reference)

В русской версии Excel

СТРОКА(ссылка)

Функция **ОСТАТ (MOD)** возвращает остаток от деления числа на делитель:

MOD(number,divisor)

В русской версии Excel

ОСТАТ(число, делитель)

Поместите функцию СТРОКА (ROW) в функцию ОСТАТ (MOD) (чтобы передать числовой аргумент), разделите на 2 (чтобы суммировать каждую вторую ячейку) и проверьте, не равен ли результат нулю. Если это так, ячейка суммируется.

Эти функции можно использовать разнообразнейшими способами — некоторые обеспечат лучший результат, чем другие. Например, формула массива для суммирования каждой второй ячейки в диапазоне \$A\$1:\$A\$100 могла бы выглядеть так:

```
=SUM(IF(MOD(ROW($A$1:$A$500),2)=0,$A$1:$A$500,0))
```

В русской версии Excel

```
=СУММ(ЕСЛИ(ОСТАТ(СТРОКА($A$1:$A$500),2)=0,$A$1:$A$500,0))
```

ПРИМЕЧАНИЕ

Поскольку это формула массива, необходимо ввести ее, нажав сочетание клавиш Ctrl+ Shift+Enter. Excel добавит фигурные скобки, чтобы она выглядела так:

```
{=SUM(IF(MOD(ROW($A$1:$A$500),2)=0,$A$1:$A$500,0))}
```

В русской версии Excel:

```
{=СУММ(ЕСЛИ(ОСТАТ(СТРОКА($A$1:$A$500),2)=0,$A$1:$A$500,0))}
```

Нужно, чтобы Excel сам добавил фигурные скобки; если вы добавите их самостоятельно, формула не будет работать.

Хотя цель достигнута, такой способ отрицательно влияет на дизайн электронной таблицы. Это ненужное применение формулы массива. Чтобы еще ухудшить дела, в эту длинную формулу вложена пересчитываемая функция СТРОКА (ROW), превращающая большую формулу также в пересчитываемую. Это означает, что она будет постоянно пересчитываться, что бы вы ни сделали в рабочей книге. Это очень плохой способ!

Вот другая формула, представляющая собой чуть лучший выбор:

```
=SUMPRODUCT((MOD(ROW($A$1:$A$500),2)=0)*($A$1:$A$500))
```

В русской версии Excel

```
=СУММПРОИЗВ((ОСТАТ(СТРОКА($A$1:$A$500),2)=0)*($A$1:$A$500))
```

Нужно, однако, помнить, что эта формула вернет ошибку #ЗНАЧ! (#VALUE!), если какие-либо ячейки диапазона содержат не числа, а текст. Эта формула, хотя и не является фактически формулой массива, также замедляет работу Excel, если использовать ее слишком много раз или если каждый раз она ссылается на большой диапазон.

К счастью, есть лучший способ, представляющий собой не только более эффективное, но и намного более гибкое решение. Он требует применения функции БДСУММ (DSUM). В этом примере мы использовали диапазон A1:A500 как диапазон, в котором необходимо суммировать каждую n-ю ячейку.

В ячейку E1 введите слово Criteria. В ячейку E2 введите следующую формулу:

```
=MOD(ROW(A2)-$C$2-1,$C$2)=0
```

В русской версии Excel

```
=ОСТАТ(СТРОКА(A2)-$C$2-1,$C$2)=0
```

Выделите ячейку C2 и выберите команду Данные ▶ Проверка (Data ▶ Validation). В поле Тип данных (Allow) выберите пункт Список (List), а в поле Источник (Source) введите 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. Удостоверьтесь, что установлен флажок Список допустимых значений (In-Cell), и щелкните на кнопке ОК. В ячейке C1 введите текст SUM every.... В любой ячейке, кроме строки 1, введите следующую формулу:

```
=DSUM($A:$A,1,$E$1:$E$2)
```

В русской версии Excel

```
=БДСУММ($A:$A,1,$E$1:$E$2)
```

В ячейке непосредственно над той, где вы ввели функцию БДСУММ (DSUM), введите текст

```
= "Summing Every " & $C$2 &  
CHOOSE($C$2,"st","nd","rd","th","th","th","th","th","th","th") & " Cell"
```

Теперь осталось только выбрать нужное число в ячейке C2, а остальное сделает функция БДСУММ (DSUM).

Как можно видеть на рис. 6.11, при помощи функции БДСУММ (DSUM) можно суммировать ячейки через указанный вами интервал. Функция БДСУММ (DSUM) намного эффективнее, чем формула массива или функция СУММПРОИЗВ (SUMPRODUCT). Хотя настройка занимает немного больше времени, это тот случай, когда тяжело в ученье, легко в бою.

	A	B	C	D	E	F	G
1	Numbers		SUM every		Criteria T	This cell cannot have the same heading as our numbers.	
2	\$ 5.00				FALSE		
3	\$ 7.50		Summing Every 3rd Cell				
4	\$ 23.45			35.46			
5	\$ 10.00						
6	\$ 9.96						

Рис. 6.11. Возможный результат с форматированием

ТРЮК
№ 70

Поиск n-го вхождения значения

Встроенные функции поиска Excel могут выполнять довольно сложные задачи, но, к сожалению, в Excel нет специальной функции, которая могла бы возвращать n-е вхождение указанных данных. К счастью, есть способы научить Excel это делать.

В таблице данных можно применить функции поиска и ссылочные функции Excel, чтобы выбрать детали, соответствующие указанному значению. Возможно, самой популярной функцией Excel такого типа является ВПР (VLOOKUP). Хотя ее очень удобно применять для поиска указанного значения в самом левом столбце таблицы, найти n-е значение в самом левом столбце она не сможет.

Можно, однако, применить очень простой способ поиска любого указанного вхождения при помощи функции ВПР (VLOOKUP) или любой другой функции поиска.

В этом примере мы предполагаем, что у вас есть таблица, состоящая из двух столбцов данных, где в столбце А размещены имена, а в столбце В — возраст людей (рис. 6.12).

	A	B
1	Names	Ages
2	Fred	20
3	Joe	25
4	Dave	18
5	James	22
6	Dave	13
7	Reece	15
8	Jill	19
9	Robyn	18
10	Marlene	17
11	Dave	25
12	Tania	23
13	Bob	24
14	Wendy	22
15	Aleisha	20
16	Kate	18
17	Dave	19
18	Jack	21
19	Bill	23

Рис. 6.12. Настройка данных для функции ВПР (VLOOKUP)

При помощи функции ВПР (VLOOKUP) можно выделить возраст человека в зависимости от его имени. К сожалению, некоторые имена встречаются несколько раз. Вы хотите, чтобы функция ВПР (VLOOKUP) смогла найти не только первое вхождение имени Dave, но и последующие. Вот, как это можно сделать (помните, что в этом примере данные находятся в столбцах А и В).

Сначала полностью выделите столбец А, щелкнув букву А в заголовке столбца, а затем выберите команду Вставка ► Столбцы (Insert ► Columns), чтобы добавить пустой столбец (который станет столбцом А). Щелкните ячейку А2 (пропустив А1, так как В1 — это заголовок) и введите следующую формулу:

```
=B2&COUNTIF($B$2:B2,B2)
```

В русской версии Excel

```
=B2&СЧЁТЕСЛИ($B$2:B2,B2)
```

Скопируйте ее вниз на столько строк, сколько данных в столбце В (еще раз щелкните ячейку А2 и дважды щелкните маркер заполнения). Вы получите имена Dave1, Dave2, Dave3 и т. д. (рис. 6.13). Обратите внимание на абсолютную ссылку \$B\$2 в функции СЧЁТЕСЛИ (COUNTIF) и на то, что остальные ссылки — относительные. Это необходимо, чтобы функция работала правильно.

Если вы еще не догадались: теперь в столбце А можно найти n-е вхождение любого имени.

Щелкните ячейку D2 и введите следующую формулу:

```
=VLOOKUP("Dave3", $A$1:$C$100, 3, FALSE)
```

	A	B	C	3
1		Names	Ages	!
2	Fred1	Fred	20	
3	Joe1	Joe	25	
4	Dave1	Dave	18	
5	James1	James	22	
6	Dave2	Dave	13	
7	Reece1	Reece	15	
8	Jill1	Jill	19	
9	Robyn1	Robyn	18	
10	Marlene1	Marlene	17	
11	Dave3	Dave	25	
12	Tania1	Tania	23	
13	Bob1	Bob	24	
14	Wendy1	Wendy	22	
15	Aleisha1	Aleisha	20	
16	Kate1	Kate	18	
17	Dave4	Dave	19	
18	Jack1	Jack	21	
19	Bill1	Bill	23	

Рис. 6.13. Данные с формулой ВПР (VLOOKUP), добавленной к столбцу А

В русской версии Excel

=ВПР("Dave3", \$A\$1:\$C\$100, 3, ЛОЖЬ)

Формула вернет возраст для третьего вхождения имени Dave (рис. 6.14).

	A	B	C	D
1		Names	Ages	!
2	Fred1	Fred	20	25
3	Joe1	Joe	25	
4	Dave1	Dave	18	
5	James1	James	22	
6	Dave2	Dave	13	
7	Reece1	Reece	15	
8	Jill1	Jill	19	
9	Robyn1	Robyn	18	
10	Marlene1	Marlene	17	
11	Dave3	Dave	25	
12	Tania1	Tania	23	
13	Bob1	Bob	24	
14	Wendy1	Wendy	22	
15	Aleisha1	Aleisha	20	
16	Kate1	Kate	18	
17	Dave4	Dave	19	
18	Jack1	Jack	21	
19	Bill1	Bill	23	

Рис. 6.14. Данные со второй формулой ВПР (VLOOKUP), добавленной к столбцу D

Конечно, можно скрыть столбец А, так как видеть его не нужно.

Кроме того, имена в столбце А можно использовать как исходный диапазон для списка в другой ячейке, выбрав команду Данные ▶ Проверка ▶ Список (Data ▶ Validation ▶ List) и введя ссылку на ячейку, содержащую этот список, в функции ВПР (VLOOKUP).

Т Р Ю К
№ 71

Превращение функции Excel ПРОМЕЖУТОЧНЫЕ.ИТОГИ (SUBTOTAL) в динамическую

Хотя функция ПРОМЕЖУТОЧНЫЕ.ИТОГИ (SUBTOTAL) является одной из самых удобных функций Excel, иногда вы хотите выбрать функцию, которая используется внутри этой функции, или применить ее к диапазону, который может увеличиться или уменьшиться.

Функция ПРОМЕЖУТОЧНЫЕ.ИТОГИ (SUBTOTAL) используется в Excel для выполнения определенной функции в диапазоне ячеек, к которому предварительно были применены автофильтры (AutoFilter). После применения автофильтров функция ПРОМЕЖУТОЧНЫЕ.ИТОГИ (SUBTOTAL) работает только на видимых ячейках, все скрытые строки игнорируются. Выполняемое действие зависит только от числа (от 1 до 11), которое вы подаете в качестве первого аргумента, номер_функции (Function_num). Например:

```
=SUBTOTAL(1,A1:A100)
```

В русской версии Excel

```
=ПРОМЕЖУТОЧНЫЕ.ИТОГИ(1,A1:A100)
```

вычислит среднее для всех видимых ячеек в диапазоне A1:A100, после того как был применен автофильтр. Если все строки диапазона A1:A100 видимы, она просто считает среднее для них и выдает в точности такой же результат, что и функция

```
=AVERAGE(A1:A100)
```

В русской версии Excel

```
=СРЗНАЧ(A1:A100)
```

Число, которое указывается в качестве первого аргумента функции ПРОМЕЖУТОЧНЫЕ.ИТОГИ (SUBTOTAL), номер_функции (Function_num) и соответствующие функции перечислены в табл. 6.1.

Таблица 6.1. Номера функций ПРОМЕЖУТОЧНЫЕ.ИТОГИ (SUBTOTAL) и соответствующие функции

Номер функции	Функция
1	СРЗНАЧ (AVERAGE)
2	СЧЁТ (COUNT)
3	СЧЁТЗ (COUNTA)
4	МАКС (MAX)
5	МИН (MIN)
6	ПРОИЗВЕД (PRODUCT)
7	СТАНДОТКЛОН (STDEV)
8	СТАНДОТКЛОНП (STDEVP)
9	СУММ (SUM)
10	ДИСП (VAR)
11	ДИСПР (VARP)

Нужно всего лишь указать число от 1 до 11, и функция **ПРОМЕЖУТОЧНЫЕ.ИТОГИ** (SUBTOTAL) сможет выполнять любую нужную функцию. Можно даже выбрать ее в раскрывающемся списке, расположенном в любой понравившейся ячейке. Вот как это делается.

Добавьте все названия функций в указанном в табл. 6.1 порядке в диапазон ячеек. Для этого примера мы будем использовать диапазон D1:D11. Выделив этот диапазон, щелкните поле имени и введите имя Subs. Нажмите клавишу Enter.

Полностью выделите столбец D и выберите команду **Формат** ▶ **Столбец** ▶ **Скрыть** (Format ▶ Column ▶ Hide). Теперь выберите команду **Вид** ▶ **Панели инструментов** ▶ **Формы** (View ▶ Toolbars ▶ Forms), щелкните элемент управления **Поле со списком** (ComboBox) и затем щелкните ячейку C2.

При помощи маркеров размера установите такой размер поля со списком, чтобы можно было отобразить самое длинное название — **СТАНДОТКЛОНП** (или **AVERAGE**, если вы работаете в английской версии Excel).

ПРИМЕЧАНИЕ

Чтобы поле со списком автоматически регулировалось в соответствии с размерами столбца и строки, где оно находится, при регулировке размера поля со списком удерживайте клавишу Alt.

Правой кнопкой мыши щелкните поле со списком и выберите команду контекстного меню **Формат объекта** (Format Control). Перейдите на вкладку **Элемент управления** (Control). В поле **Формировать список по диапазону** (Input range) введите Subs. В поле **Связь с ячейкой** (Cell-Link) введите \$C\$2. Теперь в поле **Количество строк списка** (Drop-down lines) введите 11. В ячейке C3 введите следующую формулу:

```
=IF($C$2="", "", "Result of "&INDEX(Subs,$C$2))
```

В русской версии Excel

```
=ЕСЛИ($C$2="", "", "Result of "&ИНДЕКС(Subs,$C$2))
```

В ячейке C4 введите формулу

```
=IF($C$2="", "", SUBTOTAL($C$2,$A$4:$A$100))
```

В русской версии Excel

```
=ЕСЛИ($C$2="", "", ПРОМЕЖУТОЧНЫЕ.ИТОГИ($C$2,$A$4:$A$100))
```

где \$A\$4:\$A\$100 — диапазон, в котором будут подсчитываться промежуточные итоги.

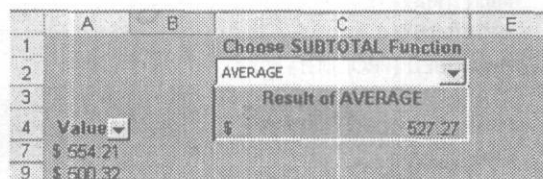


Рис. 6.15. Регулируемая функция **ПРОМЕЖУТОЧНЫЕ.ИТОГИ** (SUBTOTAL)

Теперь осталось выбрать нужную функцию **ПРОМЕЖУТОЧНЫЕ.ИТОГИ** (SUBTOTAL) в поле со списком и увидеть правильный результат (рис. 6.15).

Т Р Ю К
№ 72

Добавление расширений данных

В Excel предусмотрено множество различных форматов даты, которые можно использовать для ее отображения. Однако одного формата всегда не хватало (и он все так же не существует) — возможности отображать даты как 15th October 2003. Если это необходимо, можно научить Excel выводить дату в таком формате.

В Excel нет такого формата, чтобы после числа 15 выводились буквы «th». Что хуже всего, насколько мы знаем: невозможно определить пользовательский формат вывода даты в таком стиле. Хотя большинство людей просто мирятся с тем, что это невозможно, мы придумали способ, как этого добиться.

На чистом рабочем листе, начиная с ячейки A1, введите следующие данные: A1=st, A2=nd, A3=rd, A4:A20=th, A21=st, A22=nd, A23=rd, A24:A30=th, A31=st.

Присвойте этому диапазону имя Extensions, а затем выберите команду Вставка ▶ Имя ▶ Присвоить (Insert ▶ Name ▶ Define) и в поле Имя (Names in workbook) введите MyToday. В поле Формула (Refers to) введите следующую формулу:

```
=TEXT(TODAY(),"ddd d")&INDEX(Extensions.DAY(TODAY()),1) & TEXT(TODAY(), "mmm ууу")
```

В русской версии Excel

```
=ТЕКСТ(СЕГОДНЯ(),"ddd д")&ИНДЕКС(Extensions.ДЕНЬ(СЕГОДНЯ()),1) & ТЕКСТ(СЕГОДНЯ(),"mmm гgg")
```

Щелкните на кнопке Добавить (Add), затем щелкните на кнопке ОК.

Теперь в любой ячейке введите =MyToday, и текущая дата будет выведена в формате Thursday 16th October 2003.

Если вы не хотите хранить расширения даты, то есть th, st, rd и nd в ячейках на рабочем листе, можно использовать функцию **ВЫБОР** (CHOOSE). Для этого выберите команду Вставка ▶ Имя ▶ Присвоить (Insert ▶ Name ▶ Define) и в поле Имя (Names in workbook) введите Day.

В поле Формула (Refers to) введите следующую формулу:

```
=DAY(TODAY())
```

В русской версии Excel

```
=ДЕНЬ(СЕГОДНЯ())
```

Щелкните на кнопке Добавить (Add). Вернитесь в поле Имя (Names in workbook) и введите слово MyToday. В поле Формула (Refers to) введите следующую формулу и щелкните на кнопке Добавить (Add):

```
=TEXT(TODAY(),"ddd d")&IF(Day=31,"st",IF(Day=30,"th",CHOOSE(Day,"st","nd","rd","th","th","th",
```

```
"th"."th"."th"."th"."th"."th"."th"."th"."th"."th"."th"."th"."th"."th"."st"."nd".
"rd"."th"."th"."th"."th"."th"."th"."th"))& TEXT(TODAY(),"mmm yy")
```

В русской версии Excel

```
=ТЕКСТ(СЕГОДНЯ(),"ддд д")
&ЕСЛИ(Day=31,"st",ЕСЛИ(Day=30,"th",ВЫБОР(Day,"st","nd","rd","th",
"th"."th"."th"."th"."th"."th"."th"."th"."th"."th"."th"."th"."th"."th",
"th"."st"."nd"."rd"."th"."th"."th"."th"."th"."th"))&ТЕКСТ(СЕГОДНЯ(),
"мммм гггг")
```

После щелчка на кнопке ОК в любую ячейку можно будет ввести =MyToday, и вы увидите дату в этом формате.

ВНИМАНИЕ

Дата, которую вернет любая из этих функций, не будет настоящим числовым значением, это будет просто текстовая запись, на которую нельзя будет сослаться из формулы, ожидающей числовые данные.

Т Р Ю К
№73

Преобразование чисел со знаком минус справа

Вы когда-нибудь встречались с импортированными отрицательными числами, знак минус у которых стоял справа? SAP — это одна из программ, которые проделывают такое с отрицательными числами: например, 200- вместо -200. Изменение формата вручную — это весьма скучная работа, но не обязательно ее делать вручную.

Предположим, есть длинный список чисел, только что импортированных из другой программы, и некоторые из них являются так называемыми отрицательными числами. Ваша задача — преобразовать их в настоящие отрицательные числа, которые сможет распознать Excel. В этом упражнении мы будем использовать диапазон A1:A100. В ячейке B1 введите следующую формулу:

```
=SUBSTITUTE(IF(RIGHT(TRIM(A1))="-",RIGHT(TRIM(A1))&A1,A1),"-","")+0
```

В русской версии Excel

```
=ПОДСТАВИТЬ(ЕСЛИ(ПРАВСИМВ(СЖПРОБЕЛЫ(A1))="-",ПРАВСИМВ(СЖПРОБЕЛЫ(A1))&A1,A1),
"-","")+0
```

Введите ее в такое количество ячеек этого столбца, сколько данных в столбце A1, а затем скопируйте введенные формулы и выделите столбец A1. Выберите команду Правка ▶ Специальная вставка ▶ Значения (Edit ▶ Paste Special ▶ Values), чтобы удалить формулу и оставить только значения. На рис. 6.16 показано, как список выглядел до применения формулы (диапазон A1:A7) и после.

Чтобы понять, как работает эта формула, введите следующую формулу в любую ячейку. Ячейка A1 должна содержать значение 200-:

```
=RIGHT(TRIM(A1),1)&A1
```

	A	B
T	200--200
T	150	150
3	356--356
4	200	200
5	526526
B	301-	-301
7	258;	258

Рис. 6.16. До и после перемещения знака минус

В русской версии Excel

```
=ПРАВСИМВ(СЖПРОБЕЛЫ(A1),1)&A1
```

Вы получите результат -200-

Функция СЖПРОБЕЛЫ (TRIM) просто гарантирует, что в ячейке нет символов пробела. Получив значение -200-, необходимо удалить второе вхождение знака минус. Это делает функция ПОДСТАВИТЬ (SUBSTITUTE). Вы приказали ей заменить второе вхождение символа - на "" (пустой текст). В результате возвращается текстовое значение, поэтому вы просто прибавляете к нему 0, и Excel преобразует это значение в число.

Если вам необходимо часто выполнять эту работу, нужно записать специальный макрос, упрощающий ее. Вот как это сделать. Макрос был оптимизирован, чтобы обеспечить максимальную скорость выполнения.

Листинг 6.2

```
Sub ConvertMirrorNegatives()
Dim rCell As Range
Dim rRange As Range
Dim lCount As Long
Dim lLoop As Long

'Удостовериться, что данные выделены; если это не так, вывести _
'сообщение и закончить работу макроса.
If Selection.Cells.Count = 1 Then
MsgBox "Выберите диапазон для преобразования", vbInformation
Exit Sub
End If

'Определить переменную ТОЛЬКО для текстовых значений, например. 200-
On Error Resume Next
Set rRange = Selection.SpecialCells(xlCellTypeConstants, xlTextValues)

'Если переменная возвращает Nothing, неправильных отрицательных чисел нет: _
'вывести сообщение и закончить работу макроса
If rRange Is Nothing Then
MsgBox "Зеркальных отрицательных чисел нет", vbInformation
On Error GoTo 0
Exit Sub
End If
```

продолжение ↗

Листинг 6.2 (продолжение)

```
'Подсчитать, сколько ячеек содержат значения вида 200- и передать это число_
' в переменную для управления количеством проходов цикла.
ICount = WorksheetFunction.CountIf(Selection, "*-")
'Установить переменную на первую ячейку выделенного диапазона
Set rCell = Selection.Cells(1, 1)

'Выполнить цикл столько раз, сколько неправильных отрицательных чисел
For lLoop = 1 To ICount
'На каждом проходе цикла установить переменную на ячейку, содержащую *-
'Звездочка - это символ шаблона
Set rCell = rRange.Find(What:="*-", After:=rCell, _
    LookIn:=xlValues, LookAt:=xlPart, _
    SearchOrder:=xlByRows, SearchDirection:= _
    xlNext, MatchCase:=False)
'Использовать стандартное средство Replace для замены знака "минус" _
'пустым текстом. Другими словами, удалить его
rCell.Replace What:="-", Replacement:=""
'Умножить ячейку на -1, чтобы преобразовать в отрицательное число
rCell = rCell * -1
Next lLoop

On Error GoTo 0
End Sub
```

Чтобы применить этот макрос, выберите команду Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor) (сочетание клавиш **Alt/Option+F11**). Теперь выберите команду Insert ► Module и вставьте предыдущий код. Закройте окно, чтобы вернуться в Excel, и выберите команду Сервис ► Макрос ► Макросы (Tools ► Macro ► Macros), а затем выберите Convert Mirror Negatives. Щелкните кнопку Параметры (Options) и назначьте макросу клавиатурное сочетание. Теперь, когда нужно будет преобразовать импортированные отрицательные числа в настоящие отрицательные числа, которые сможет распознать Excel, нужно будет просто выделить их и использовать сочетание клавиш.

Т Р Ю К
№ 74

Отображение отрицательных значений времени

Иногда вам бывает нужно отобразить отрицательные значения времени, но Excel выводит только #####. Есть несколько способов решить эту проблему.

Если вы введете время 12:00:00 (полдень) в любую ячейку, а затем отформатируете ее, выбрав формат Общий (**General**), то увидите, что ее числовое значение равно 0.5 (в английской версии). Схожим образом введите время 24:00:00 в любую ячейку. Посмотрите на строку формул и вы увидите 1/1/1900 12:00:00 AM.

Для Excel даты и время — это не что иное, как числа. В случае даты он по умолчанию считает, что у 1 января 1900 года числовое значение равно 1, у 2 января 1900 года — 2 и т. д. Время считается как десятичные доли, кроме полуночи, числовое значение которой равно 1. (Подробнее об этом — в разделе «Трюк № 38. Трюки с возможностями даты и времени в Excel».) Из-за этого в Excel возникает настоящая проблема с отрицательными значениями времени.

Вот три способа заставить Excel отображать отрицательные значения времени.

Способ 1. Изменение системы дат Excel по умолчанию

Первый быстрый и простой способ научить Excel отображать отрицательные значения времени — выбрать команду пункт меню Сервис ▶ Параметры ▶ Вычисления (Tools ▶ Options ▶ Calculation) и установить флажок Система дат 1904 (1904 date system). Система дат 1904 называется также датой и временем Macintosh.

ВНИМАНИЕ

Помните, что при этом в Excel изменятся начальные даты, от которых отсчитывается значение всех ячеек. Вместо 1 января 1900 года это будет 2 января 1904 года. Изменение этого параметра влияет только на ту рабочую книгу, в которую вы вносите изменения.

Теперь можно вычитать значения времени и видеть на рабочем листе отрицательные значения времени.

Попробуйте. Выберите команду Сервис ▶ Параметры ▶ Вычисления (Tools ▶ Options ▶ Calculation) и установите флажок Система дат 1904 (1904 date system). В ячейку A2 введите 5:00:00 AM, в ячейку A3 — 6:00:00 AM, в ячейку A4 - =A2-A3.

Вы получите результат -1:00, но только если выберете систему дат 1904, как описано ранее.

Если изменение системы дат Excel по умолчанию на систему дат 1904 может вызвать проблемы в других вычислениях времени в рабочей книге, нужен другой способ, который изменяет вид ячеек, содержащих отрицательные значения.

Способ 2. Функция ТЕКСТ (TEXT)

Второй способ требует использования функции ТЕКСТ (TEXT). Для начала в ячейке A2 введите 5:00:00 AM, в ячейке A3 — 6:00:00 AM. В ячейке A4 введите следующую формулу:

```
=TEXT(MAX($A$2:$A$3)-MIN($A$2:$A$3), "-H:MM")
```

В русской версии Excel

```
=ТЕКСТ(МАКС($A$2:$A$3)-МИН($A$2:$A$3), "-Ч:MM")
```

Используя эту вложенную формулу, вы вычтете A3 из A2, чтобы получить положительное значение времени. Затем отформатируете ячейку при помощи функции ТЕКСТ (TEXT), чтобы она могла отображать отрицательные значения времени. Комбинация функций МАКС (MAX) и МИН (MIN) гарантирует, что более раннее время всегда будет вычитаться из более позднего.

ПРИМЕЧАНИЕ

Если вы захотите использовать полученное значение в другой формуле, нужно помнить, что возвращенный результат в действительности является текстовым значением, а не числовым.

Способ 3. Пользовательский формат

Еще один способ отображения отрицательных значений времени без перехода к системе дат 1904 и с гарантированным возвращением числового значения — это использование пользовательского формата `-ч:мм (-h:mm)` в диалоговом окне **Формат** ▶ **Ячейки** ▶ **Все форматы** (**Format** ▶ **Cells** ▶ **Custom**).

Этот способ работает только в том случае, если вы *всегда* хотите отображать только отрицательные значения времени. Он требует также, чтобы более раннее время вычиталось из более позднего. Это означает, что на самом деле возвращаться будут *положительные* значения времени, но перед ними будет выводиться знак минус.

Так, используя один из этих трех способов, вы сможете отображать отрицательные значения времени. Помните, что свой подвох заключен в каждом способе, поэтому применяйте их, не забывая о возможных подводных камнях.

Т Р Ю К
№ 75

Использование функции ВПР (VLOOKUP) в нескольких таблицах

Хотя сама по себе функция ВПР (VLOOKUP) очень удобна, она ограничена поиском в определенной таблице и возвращает результат, которого иногда может быть недостаточно. Этому ограничению можно избежать при помощи функции **ДВССЫЛ** (INDIRECT).

Иногда нужно использовать одну формулу ВПР (VLOOKUP) для поиска в различных таблицах существующих данных. Один из способов, как это можно сделать, — вложить несколько функций ВПР (VLOOKUP), приказав им проводить поиск в определенной таблице в зависимости от числа, которое вы вводите в определенной ячейке. Например:

```
=IF(A1=1,VLOOKUP(B1,Table1,2,FALSE),IF(A1=2,VLOOKUP(B1,Table2,2,FALSE),""))
```

В русской версии Excel

```
=ЕСЛИ(A1=1,ВПР(B1,Table1,2,ЛОЖЬ),ЕСЛИ(A1=2,ВПР(B1,Table2,2,ЛОЖЬ),""))
```

В этой формуле вы говорите функции ВПР (VLOOKUP) искать в именованном диапазоне **Table1**, если ячейка **A1** содержит число 1, и в именованном диапазоне **Table2**, если ячейка **A1** содержит число 2.

Как можно понять, если использовать больше двух вложенных функций **ЕСЛИ** (IF), эта формула станет очень большой и неудобной. В следующей формуле, например, используется только пять вложенных функций, но она уже устрашает!

```
=IF(A1=1,VLOOKUP(B1,Table1,2,FALSE),IF(A1=2,VLOOKUP(B1,Table2,2,FALSE),
IF(A1=3,VLOOKUP(B1,Table3,3,FALSE),IF(A1=4,VLOOKUP(B1,Table4,2,FALSE),
IF(A1=5,VLOOKUP(B1,Table5,2,FALSE),"")))))
```

Хотя эта формула вернет желаемый результат, можно сделать ее намного короче, добавив при этом больше пяти условий, и получить существенно более простую формулу.

Предположим, есть 12 разных таблиц в одной электронной таблице; каждая таблица представляет один из месяцев года. Каждая таблица включает два столбца и содержит имена пяти сотрудников и пять соответствующих сумм. Имя каждой таблицы соответствует представляемому ею месяцу, то есть данные для января помещены в именованный диапазон January, данные для февраля содержатся в именованном диапазоне February и т. д. (рис. 6.17).

January		February		March		April	
Name	Amount	Name	Amount	Name	Amount	Name	Amount
Fred	20	Fred	35	Fred	35	Fred	19
Bill	21	Bill	32	Bill	32	Bill	15
Jack	22	Jack	36	Jack	36	Jack	22
Joe	23	Joe	31	Joe	31	Joe	23
Phil	20	Phil	35	Phil	35	Phil	25
May		June		July		August	
Name	Amount	Name	Amount	Name	Amount	Name	Amount
Braden	25	Braden	28	Braden	15	Braden	56
Ashley	26	Ashley	13	Ashley	22	Ashley	84
Liam	28	Liam	22	Liam	33	Liam	52
Marlene	27	Marlene	45	Marlene	65	Marlene	31
Mark	31	Mark	85	Mark	98	Mark	31
September		October		November		December	
Name	Amount	Name	Amount	Name	Amount	Name	Amount
Braden	25	Braden	25	Braden	25	Braden	25
Ashley	26	Ashley	26	Ashley	26	Ashley	26
Liam	28	Liam	28	Liam	28	Liam	28
Marlene	27	Marlene	27	Marlene	27	Marlene	27
Mark	31	Mark	31	Mark	31	Mark	31

Рис. 6.17. Двенадцать таблиц, каждая из которых представляет один месяц года

Выделите ячейку A1. Затем выберите команду Данные ► Проверка (Data ► Validation) и на вкладке Параметры (Settings) в поле Тип данных (Allow) выберите Список (List). В поле Источник (Source) введите названия месяцев по-английски, отделив их друг от друга запятыми. Важно, чтобы название именованного диапазона для каждой таблицы соответствовало названиям месяцев в списке проверки. Щелкните на кнопке ОК.

Выделите ячейку B1 и настройте список проверки, как вы это сделали только что. На этот раз укажите имена всех сотрудников. Если имена сотрудников слишком большие, в качестве источника укажите ссылку на диапазон ячеек, содержащий эти имена. Щелкните на кнопке ОК.

В ячейке A2 введите следующую формулу

```
=VLOOKUP($B$1,INDIRECT($A$1),2,FALSE)
```

В русской версии Excel

```
=ВПР($B$1,ДВССЫЛ($A$1),2,ЛОЖЬ)
```

Выберите нужный месяц из списка в ячейке A1 и нужное имя сотрудника в списке в ячейке B1, и формула в ячейке A2 вернет соответствующую сумму для этого сотрудника для данного месяца.

ПРИМЕЧАНИЕ

У этого способа есть несколько преимуществ. Функция ДВССЫЛ (INDIRECT) используется для чтения содержимого ячейки как адреса диапазона, а не как текста. Так как вы определили 12 диапазонов, каждый из которых представляет месяц года, формула, использующая функцию ДВССЫЛ (INDIRECT), считывает слово January как ссылку на диапазон, а не как текстовую строку.

Еще одно преимущество использования формулы с функцией ДВССЫЛ (INDIRECT): можно обойти ограничение Excel, которое позволяет указывать только семь уровней вложенности функций.

**Т Р Ю К
№ 76**

Отображение общего количества времени в днях, часах и минутах

При суммировании часов в Excel результат можно вернуть как общее количество часов и минут, но, к сожалению, не как дни, часы и минуты. Отображение результата в таком формате потребует немного дополнительных усилий.

Например, общее время 75 часов 45 минут и 00 секунд обычно выводится как 75:45:00, если ячейка отформатирована с использованием пользовательского формата [ч]:мм:сс ([h]:mm:ss), который позволяет выводить количество часов, большее 24. Хотя это определенно правильный результат, он означает, что необходимо вручную сосчитать, сколько дней, часов и минут заключено в этом времени. Это может быть долгой и не защищенной от ошибок работой.

Предположим, список допустимых значений времени в ячейках A1:A10. В ячейке A11 находится обычная функция СУММ (SUM), которая суммирует общее количество часов, то есть =СУММ(\$A\$1:\$A\$10). Если общее количество часов превышает 24, результирующая ячейка, содержащая функцию СУММ (SUM), должна быть отформатирована как [ч]:мм ([h]:mm). Предположим, получен результат 306:26:00, который, конечно же, представляет 306 часов и 26 минут. Однако это не говорит о том, сколько дней, часов и минут содержится в итоговом значении времени.

Чтобы результат отображался в днях, часах и минутах, воспользуйтесь следующей формулой

```
=INT(A11)&" Days "&INT(MOD(A11,INT(A11))*24)&" Hours and "&MINUTE(A11)&" Minutes"
```

В русской версии Excel

```
=ЦЕЛОЕ(A11)&" дней "&ЦЕЛОЕ(ОСТАТ(A11,ЦЕЛОЕ(A11))*24)&" часов и "&МИНУТЫ(A11)&" минут"
```

Предполагая, что в ячейке А1 содержится значение 306:26:00, мы получим в результате 12 дней, 18 часов и 26 минут.

Посмотрим, как это работает. Если вы не знаете, как Excel хранит и использует даты и время, сначала нужно прочитать и понять раздел «Трюк № 38. Трюки с возможностями даты и времени в Excel».

Выделите ячейку с результатом выполнения формулы и щелкните кнопку Вставка функции (Install function) слева от строки формул (знак равенства в ранних версиях Excel для Windows и в версии Excel для Macintosh). Щелкните первое вхождение функции ЦЕЛОЕ (INT) в строке формул. Эта функция возвращает целую часть числа 12,76805556. Результат равен количеству дней.

Затем необходимо определить, сколько часов останется, если от исходного значения отнять 12 дней. Щелкните вторую функцию ЦЕЛОЕ (INT) в строке формул. Здесь функция `ОСТАТ(А11,ЦЕЛОЕ(А11)) (MOD(A11,INT(A11)))` возвращает остаток от деления 12,76805556 на 12, равный 0,76805556 (количество минут, представленное в виде десятичного числа). Теперь это число умножается на 24 (что дает 18,433333), возвращая количество минут. Так как нам нужна только целая часть (18), мы вложили формулу `MOD(А11,INT(А11))*24 (ОСТАТ(А11,ЦЕЛОЕ(А11))*24)` в функцию ЦЕЛОЕ (INT).

Щелкните функцию МИНУТЫ (MINUTE) в строке формул. Функция возвращает значение 26 на основе порядкового числа 12,76805556.

Результат, возвращаемый функцией МИНУТЫ (MINUTE), никогда не будет числовым значением, поэтому нужно сохранить исходную функцию СУММ (SUM), которая возвращает общее количество часов для значения в ячейке. Этот результат можно использовать в дальнейших вычислениях. Строку, содержащую общее количество часов, конечно же, можно скрыть.

ТРЮК № 77 Определение количества указанных дней в любом месяце

При создании приложений с возможностями календаря, в частности платежных ведомостей, иногда нужно знать, сколько раз данный день недели встречался в определенном месяце.

Хотя в Excel много функций даты и времени, на данный момент не существует функции, которая, например, сказала бы, сколько понедельников будет в январе 2005 года. Чтобы узнать это, можно использовать много функций даты и времени Excel с глубокой вложенностью, но, к сожалению, как можно представить, это будет весьма громоздкий и неуклюжий процесс.

VBA поможет упростить сложную задачу. Вместо того чтобы нащупывать нужную комбинацию сложных функций, можно написать пользовательскую функцию, которая будет выполнять ту же задачу, и все, что нужно сделать, — ввести день и дату, для которой нужно выполнить функцию.

Для определения количества дней в определенном месяце можно воспользоваться следующей функцией. Например:

```
=HowManyDaysInMonth("1/12/03", "wed")
```

вернет 5, так как в декабре 2003 года было пять сред. (Обратите внимание, что дату нужно указывать в местном формате, например, в США это 12/1/03. В примере формат даты показан для Австралии.)

Схожим образом следующая функция:

```
=HowManyDaysInMonth("1/12/03", "thu")
```

вернет 4, так как в декабре 2003 года было четыре четверга.

Чтобы использовать эту пользовательскую функцию в рабочей книге, сначала необходимо поместить следующий код в стандартный модуль. Откройте рабочую книгу, куда нужно поместить код, и выберите команду Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor) (сочетание клавиш Alt/Option+F11). Затем выберите команду Insert ► Module и вставьте код, приведенный в листинге 6.3.

Листинг 6.3

```
Function HowManyDaysInMonth(FullDate As String, sDay As String) As Integer
Dim i As Integer
Dim iDay As Integer, iMatchDay As Integer
Dim iDaysInMonth As Integer
Dim FullDateNew As Date

iMatchDay = Weekday(FullDate)
Select Case UCase(sDay)
Case "SUN"
iDay = 1
Case "MON"
iDay = 2
Case "TUE"
iDay = 3
Case "WED"
iDay = 4
Case "THU"
iDay = 5
Case "FRI"
iDay = 6
Case "SAT"
iDay = 7
End Select
iDaysInMonth = Day(DateAdd("d", -1, DateSerial
(Year(FullDate), Month(FullDate) + 1, 1)))
FullDateNew = DateSerial(Year(FullDate), Month(FullDate), iDaysInMonth)
For i = iDaysInMonth - 1 To 0 Step -1
If Weekday(FullDateNew - i) = iDay Then
HowManyDaysInMonth = HowManyDaysInMonth + 1
End If
Next i
End Function
```

Закройте окно, чтобы вернуться в Excel.

Теперь введите функцию в любую ячейку, как показано ранее, и Excel возвратит число, указывающее, сколько раз указанный день встречался в указанном месяце.



Построение мегаформул

Мегаформулы — формулы внутри формул внутри формулы — могут свести с ума даже бывалого ветерана Excel. Однако, имея немного предусмотрительности и шаг за шагом продвигаясь к желаемой формуле, вы сможете приручить эти сложные мегаформулы и обращаться с ними без страха.

Наполняет ли вас ужасом одна мысль о том, чтобы в одиночку разобраться и составить вложенную формулу? Иногда такие ячейки, заполненные сложной функциональной белибердой, вызывают дрожь в коленях и у нас. Однако, не забывая о предусмотрительности и пошаговом подходе, вы сможете без страха создавать мегаформулы. И, возможно, только возможно, когда-нибудь вы сможете снова прочитать и понять их.

Трюк заключается в том, чтобы строить формулы по кусочкам, используя стандартные функции Excel. Используя по одной функции в каждой ячейке, получая индивидуальные управляемые результаты, вы сможете вложить функции друг в друга и получить желаемый конечный результат. Вот пример такого процесса в действии.

Предположим, есть длинный список имен людей, каждое из которых состоит из имени, отчества и фамилии, то есть по одному полному имени на ячейку. Задача состоит в том, чтобы написать формулу в соседнем столбце, которая будет извлекать только фамилию человека.

Следовательно, надо найти начало фамилии, третьего слова в ячейке. Фактически, это означает поиск позиции второго пробела в ячейке. В Excel нет стандартной встроенной функции для автоматического поиска второго пробела в ячейке, но можно воспользоваться функцией НАЙТИ (FIND), применив ее так, чтобы получить желаемый результат.

В ячейку A1 введите имя David John Hawley (или любое другое имя, состоящее из трех слов). В ячейку C1 введите следующую функцию:

```
=FIND(" ".A1)
```

В русской версии Excel

```
=НАЙТИ(" ".A1)
```

Функция НАЙТИ (FIND) ищет одну текстовую строку **искомый_текст** (*find_text*) внутри другой текстовой строки **текст_для_поиска** (*within_text*) и возвращает число, обозначающее начальную позицию **искомый_текст** (*find_text*), если отчет начинается с первого символа **текст_для_поиска** (*within_text*).

Синтаксис функции

```
=FIND(find_text, within_text, start_num)
```

В русской версии Excel

```
=НАЙТИ(искомый_текст, текст_для_поиска, нач_позиция)
```

Эта функция вернет начальную позицию первого символа пробела в ячейке A1, так как приказали ей найти " " (пробел) в ячейке A1. Если в ячейке находится текст David John Hawley, то функция вернет значение 6. Но вы ищете второй пробел, а не первый. Теперь нужно использовать число, возвращенное формулой в ячейке C1, как начальную позицию для другой функции НАЙТИ (FIND), чтобы найти второй символ пробела. Так, в ячейке C2 введите формулу

```
=FIND(" ".A1,C1+1)
```

В русской версии Excel

```
=НАЙТИ(" ".A1,C1+1)
```

Обратите внимание, что в этот раз вы передали функции НАЙТИ (FIND) третий аргумент, начальную позицию, найденную C1 (в данном примере равную 6), к которой прибавлена единица; она будет служить начальной точкой для функции НАЙТИ (FIND), начиная с которой она будет искать пробел. Вторая функция возвратит значение, обозначающее позицию второго символа пробела.

Вооружившись этими функциями, вы хотите, чтобы следующая функция выбрала все символы до конца строки текста. Воспользуйтесь функцией ПСТР (MID), которая предназначена для выделения диапазона символов из строки. В ячейке C3 введите следующую формулу:

```
=MID(A1,C2+1,256)
```

В русской версии Excel

```
=ПСТР(A1,C2+1,256)
```

Функция ПСТР (MID) возвращает указанное количество символов текстовой строки, начиная с указанной позиции. Вот ее синтаксис:

```
MID(text, start_num, num_chars)
```

В русской версии Excel

```
ПСТР(текст, начальная_позиция, количество_символов)
```

Вы приказали функции ПСТР (MID) выбрать 256 символов из ячейки A1, начиная с первого символа после второго пробела в текстовой строке. Число 256 используется только чтобы гарантировать, что, независимо от длины фамилии (предполагая, что она все же короче 256 символов), вы получите фамилию человека полностью.

При наличии всех составляющих настало время построить вложенную формулу, которой вы так боялись всего несколько минут назад, целиком. Все, что нужно сделать, — это заменить все ссылки на ячейки (кроме A1) внутри функций на формулы в этих ячейках. Это делается путем копирования и вставки в строке формул.

Щелкните ячейку C2, в строке формул выделите функцию и скопируйте ее целиком, кроме знака =, вот так:

```
FIND(" ".A1,C1+1)
```

В русской версии Excel

```
НАЙТИ(" ",A1,C1+1)
```

Нажмите клавишу Enter, чтобы покинуть ячейку, и вы окажетесь в ячейке C3. Выделив ячейку C3, в строке формул выделите ссылку на ячейку C2 и вставьте (сочетание клавиш **Ctrl+V**) функцию НАЙТИ (FIND), которую только что скопировали. Нажмите клавишу Enter. Функция в ячейке C3 должна выглядеть так:

```
=MID(A1,FIND(" ",A1,C1+1)+1,256)
```

В русской версии Excel

```
=ПСТР(A1,НАЙТИ(" ",A1,C1+1)+1,256)
```

Теперь необходимо заменить ссылку на ячейку C1 функцией, которая находится в ячейке C1. Выделите ячейку C1, выделите в строке формул формулу, опустив символ =, щелкните на кнопке Копировать (Copy), а затем дважды нажмите клавишу Enter, чтобы попасть обратно в ячейку C3. Находясь в ячейке C3, выделите C1 в строке формул и вставьте только что скопированную функцию НАЙТИ (FIND). Нажмите клавишу Enter.

Теперь осталось вырезать ячейку C3 и вставить ее в ячейку B1, а затем удалить формулы, оставшиеся в ячейках C1 и C2. Вы должны получить итоговую формулу, которая выглядит так:

```
=MID(A1,FIND(" ",A1,FIND(" ",A1)+1)+1,256)
```

В русской версии Excel

```
=ПСТР(A1,НАЙТИ(" ",A1,НАЙТИ(" ",A1)+1)+1,256)
```

Следуя этой концепции, вы должны понять, как конструировать **мегаформулы**, используя разнообразные функции Excel. Сначала необходимо запланировать путь, по которому вы собираетесь идти к цели, а затем в индивидуальных ячейках получать **необходимые** результаты. Наконец, нужно заменить все ссылки на ячейки функциями, которые находятся в этих ячейках.

ПРИМЕЧАНИЕ

Если получилось больше семи уровней **вложенности**, понадобится функция **ДВССЫЛ** (INDIRECT), описанная в разделе «Трюк № 74. Отображение отрицательных значений времени».



Трюки с мегаформулами, которые ссылаются на другие рабочие книги

Формулы Excel становятся достаточно сложными, когда мегаформулы ссылаются на другую рабочую книгу. Нужно включать не только ссылки на ячейки, но и названия рабочих книг и листов, и, кроме того, полный путь, если книга, на которую ссылается формула, закрыта. Есть несколько способов упростить этот сложный процесс.

Написание таких формул с нуля может оказаться неожиданно быстрой решаемой задачей. В этом трюке мы покажем быстрый и простой способ, позволяющий конструировать сложные формулы без необходимости указывать названия рабочих книг и пути к файлам. Этот способ настолько прост, что о нем часто даже не догадываются.

Сначала убедимся, что вы правильно создаете ссылки на ячейки и рабочие листы. При написании формул лучше всегда избегать ввода ссылок на ячейки, имен листов и рабочих книг вручную, так как при этом можно случайно использовать неверный синтаксис или сделать опечатку. Большинству пользователей среднего уровня следует создавать ссылки на ячейки, листы и рабочие книги только при помощи мыши. Это определенно помогает предотвратить синтаксические ошибки и опечатки, однако если вы когда-либо пробовали делать это с вложенной функцией, вы знаете, что она быстро становится громоздкой и непонятной.

Например, взгляните на такую формулу:

```
=INT(SUM('C:\Ozgrid Likom\Finance\SoftwareSales\[Regnow.xls]Product  
Sales'!C2:C2924))
```

В русской версии Excel

```
=ЦЕЛОЕ(СУММ('C:\Ozgrid Likom\Finance\SoftwareSales\[Regnow.xls]Product  
Sales'!C2:C2924))
```

Это достаточно простая функция СУММ (SUM), вложенная в функцию ЦЕЛОЕ (INT). Она ссылается на ячейки в закрытой рабочей книге, поэтому указан весь путь вместе с ссылками на ячейки и названиями рабочего листа и рабочей книги. Однако если понадобится вложить в эту функцию еще несколько дополнительных функций, это скоро станет весьма сложной задачей.

Мы покажем быстрый способ написания **мегафункций**, которые ссылаются на внешние рабочие книги. Трюк состоит в том, чтобы написать функцию в той рабочей книге, на которую хотите сослаться, в любой пустой ячейке. Если сослаться только на один рабочий лист в этой рабочей книге, лучше использовать ячейку на этом листе.

Сначала, используя метод простого написания вложенных функций, описанный в разделе «Трюк № 77. Определение количества указанных дней в любом месяце», сконструируйте формулу в любой пустой ячейке в той рабочей книге, на которую она будет ссылаться. Получив желаемый результат, вырежьте формулу из ячейки, перейдите в ту рабочую книгу, где должен появиться результат, выберите подходящую ячейку и вставьте.

Excel выполнит всю черную работу за вас, вставив названия рабочих книг и листов. Если понадобится добавить или модифицировать формулу, просто откройте рабочую книгу, на которую она ссылается, вырежьте формулу из исходной рабочей книги и вставьте в эту. Затем внесите необходимые изменения и вставьте модифицированную формулу в книгу, где хотите видеть конечный результат.

Т Р Ю К
№ 80

Трюк с одной функцией баз данных Excel, чтобы она работала вместо нескольких функций

Функции баз данных Excel, например, БДСУММ (DSUM), БСЧЁТ (DCOUNT) и другие могут работать вместо, возможно, тысяч функций, сокращая, таким образом, и время пересчета, и объем рабочей книги.

При работе с функциями баз данных Excel можно указать до 256 различных критериев. Может понадобиться, например, суммировать значения в столбце А, только если соответствующие значения в столбце В больше 100, а соответствующий возраст в столбце С меньше 40. Если, однако, вы хотите суммировать те значения, для которых соответствующие значения в столбце В меньше 50, понадобится использовать другую функцию и другой диапазон критерия. Было бы намного проще, если бы была только одна функция, для которой можно быстро и просто изменить критерий! Если вы раньше никогда не работали с функциями баз данных Excel, настоятельно рекомендуем познакомиться с ними, так как они прекрасно подходят для выделения статистической информации из базы данных или таблицы Excel.

Чтобы увидеть, как это работает, задайте данные (рис. 6.18). Укажите те же заголовки столбцов, но в самих столбцах могут быть любые фиктивные данные. Присвойте этой таблице данных, включая все заголовки столбцов, имя AllData. Назовите лист Data.

	A	B	C	D	E	F
1	Name	Dates	Full Cost	Amount Paid	Percent Paid	Cost Remaining
2	Bill J	2/15/01	\$ 25.00	\$ 10.00	40%	\$ 15.00
3	John J	2/16/01	\$ 35.00	\$ 12.00	34%	\$ 23.00
4	Fred B	2/19/01	\$ 25.00	\$ 20.00	80%	\$ 5.00
5	Joe H	2/20/01	\$ 65.00	\$ 65.00	100%	\$ -
6	Mary K	2/21/01	\$ 88.00	\$ 80.00	91%	\$ 8.00
7	Lisa G	2/22/01	\$ 45.00	\$ 25.00	56%	\$ 20.00
8	Dave H	2/23/01	\$ 60.00	\$ 55.00	92%	\$ 5.00
9	Edward F	2/26/01	\$ 21.00	\$ 21.00	100%	\$ -
10	Keith B	2/27/01	\$ 33.00	\$ 10.00	30%	\$ 23.00
11	Aleisha H	2/28/01	\$ 22.00	\$ 22.00	100%	\$ -
12	Kylie M	3/1/01	\$ 25.00	\$ 20.00	80%	\$ 5.00
13	Jonn D	3/2/01	\$ 44.00	\$ 15.00	34%	\$ 29.00
14	Bill W	3/5/01	\$ 88.00	\$ 45.00	51%	\$ 43.00
15	Harry B	3/6/01	\$ 77.00	\$ 28.00	36%	\$ 49.00

Рис. 6.18. Данные для работы

Добавьте еще один рабочий лист и назовите его Results. В ячейке A2 введите следующую формулу:

```
=Data!A1
```

Скопируйте ее до ячейки F2, чтобы получить зеркальное отображение заголовков вашей таблицы. В ячейке A3 введите любое имя, присутствующее в таблице на листе с данными, например, John D. Затем в ячейке B3 введите формулу

```
=DGET(AllData,B2,$A$2:$A$3)
```

В русской версии Excel
 =БИЗВЛЕЧЬ(A11Data.B2.\$A\$2:\$A\$3)

Скопируйте эту формулу до ячейки F3 и отформатируйте ячейки C3:F3 нужным образом.

ПРИМЕЧАНИЕ

Чтобы быстро скопировать ячейки, подобные этим, без форматирования, выделите ячейку, правой кнопкой мыши щелкните маркер заполнения и, удерживая правую кнопку мыши, перетащите маркер на нужную позицию. Затем в контекстном меню выберите команду Заполнить значения (Fill Without Formatting).

Соответствующие данные для имени, введенного в ячейке A3, должны быть извлечены из таблицы с данными. Это простой пример того, как можно извлекать связанную информацию при помощи функции БИЗВЛЕЧЬ (DGET).

ВНИМАНИЕ

Если вы получите ошибку #ЧИСЛ! (#NUM!), это означает, что в столбце Name есть два или более одинаковых имени.

Сейчас большинство пользователей последовали бы той же концепции для всех имен, для которых необходимо получить информацию из таблицы. Однако эти усилия не нужны.

Так как вы всегда ссылаетесь на ячейку A3, получая оттуда имя, было бы намного удобнее, чтобы в этой ячейке находился раскрывающийся список, содержащий все имена из таблицы данных. Для создания такого списка можно воспользоваться стандартной возможностью проверки Excel. Однако, так как исходный список имен находится на другом рабочем листе, нельзя сослаться на него как на список на том же листе, то есть указать стандартную ссылку на диапазон. Это ограничение можно легко обойти, присвоив имя столбцу Name исходной таблицы, а затем указав это имя как источник для списка проверки.

ПРИМЕЧАНИЕ

Так как большинство таблиц являются статическими, другими словами, данные обычно постоянно добавляются и удаляются, лучше для столбца Names использовать динамический именованный диапазон. Подробнее об этом — в разделе «Трюк № 42. Создание увеличивающихся и уменьшающихся диапазонов».

Снова щелкните лист Data и, выделив любую ячейку, выберите команду Вставка ► Имя ► Присвоить (Insert ► Name ► Define). В поле Имя (Names in workbook) введите Names. В поле Формула (Refers to) введите следующую формулу и щелкните на кнопке Добавить (Add):

=OFFSET(\$A\$2,0,0,COUNTA(\$A\$2:\$A\$1000),1)

В русской версии Excel

=СМЕЩ(\$A\$2.0.0.СЧЁТЗ(\$A\$2:\$A\$1000),1)

Щелкните рабочий лист Results, выделите ячейку A3 и выберите команду Данные ▶ Проверка (Data ▶ Validation). В поле Тип данных (Allow) выберите Список (List), а в поле Источник (Source) введите формулу

```
=Name
```

Удостоверьтесь, что установлен флажок Список допустимых значений (In-Cell), и щелкните на кнопке ОК. Теперь в списке в ячейке A3 можно выбрать любое имя, и справа будут автоматически выведены соответствующие ему данные.

Можно продвинуться еще дальше и использовать функцию **БСЧЁТ (DCOUNT)**, чтобы узнать количество человек, для которых значение Full Cost больше одного указанного числа, а значение Percent Paid меньше другого.

Для этого сначала необходимо создать динамический именованный диапазон для столбцов Full Cost и Percent Paid. В поле Имя (Names in workbook) введите FullCost. В поле Формула (Refers to) введите следующую формулу и щелкните на кнопке Добавить (Add):

```
=OFFSET($C$2,0,0,COUNTA($C$2:$C$1000),1)
```

В русской версии Excel

```
=СМЕЩ($C$2,0,0,СЧЁТ3($C$2:$C$1000),1)
```

После этого в поле Имя (Names in workbook) введите PercentPaid. В поле Формула (Refers to) введите следующую формулу и щелкните на кнопке Добавить (Add):

```
=OFFSET($E$2,0,0,COUNTA($E$2:$E$1000),1)
```

В русской версии Excel

```
=СМЕЩ($E$2,0,0,СЧЁТ3($E$2:$E$1000),1)
```

Перейдите на лист Results, выделите ячейку AI и выберите команду Данные ▶ Проверка (Data ▶ Validation). В поле Тип данных (Allow) выберите Список (List), а в поле Источник (Source) введите =Full_Cost. Щелкните на кнопке ОК.

Выделите ячейку AI и выберите команду Данные ▶ Проверка (Data ▶ Validation). В поле Тип данных (Allow) выберите Список (List), а в поле Источник (Source) введите =Percent_Paid. Щелкните на кнопке ОК.

В ячейке A12 введите формулу

```
=Data!C1
```

Выделите ячейку B12 и введите следующую формулу:

```
=Data!E1
```

Выделите ячейку A13 и введите следующую формулу:

```
= ">"&A11
```

Выделите ячейку B13 и введите следующую формулу:

```
= "<"&TEXT(B11, "0%")
```

В русской версии Excel

```
= "<"&ТЕКСТ(B11, "0%")
```

В ячейке A15 введите следующее:

```
=DCOUNT(A11:Data,$A$12,$A$12:$B$13)
```

В русской версии Excel

```
=БСЧЁТ(A11:Data,$A$12,$A$12:$B$13)
```

Выберите любое значение Full Cost в ячейке AI и любое значение Percent Paid в ячейке B11, и функция БСЧЁТ (DCOUNT) сообщит количество людей, отвечающих этим критериям. Например, если вы выберете 65 и 100 %, то узнаете количество людей, для которых значение Full Cost больше 65, а значение Percent Paid меньше 100.

Как видно, при помощи функции БСЧЁТ (DCOUNT) можно извлекать любую комбинацию критериев для столбцов Full Cost и Percent Paid. Приложив еще немного усилий, можно перевести это на очередной уровень и сделать операторы, которые используются в критерии, взаимозаменяемыми.

Сначала необходимо создать список операторов сравнения, которые можно использовать в списке проверки. Перейдите к любому неиспользуемому столбцу на листе Results и в любой строке этого столбца введите заголовок Operators. Под заголовком введите операторы =, >=, >, < и <= (рис. 6.19).



Рис. 6.19. Операторы сравнения

Чтобы присвоить диапазону имя, выделите заголовок и все операторы ниже его и выберите команду Вставка ▶ Имя ▶ Создать (Insert ▶ Name ▶ Create). Удостоверьтесь, что выбран переключатель В строке выше (Top Row Only), и щелкните на кнопке ОК. Excel автоматически присвоит этому диапазону имя, основываясь на его заголовке, — в данном случае, Operators.

Выделите ячейку G7 и введите заголовок Select a Criteria.

Выделив ячейки G7 и H7, выровняйте их по центру, выбрав команду Формат ▶ Ячейки ▶ Выравнивание (Format ▶ Cells ▶ Alignment) и в поле По горизонтали (Horizontal TextAlignment) выбрав вариант По центру выделения (Center Across Selection).

Выделите ячейки G8 и H8, выберите команду Данные ▶ Проверка (Data ▶ Validation) и в списке Тип данных (Allow) выберите Список (List). В поле Источник (Source) введите =Operators. Удостоверьтесь, что установлен флажок Список допустимых значений (In-Cell), и щелкните на кнопке ОК.

Перейдите обратно на лист Data и создайте динамический именованный диапазон для столбца Dates. Выберите команду Вставка ▶ Имя ▶ Присвоить (Insert ▶ Name ▶ Define) и в поле Имя (Names in workbook) введите Dates. В поле Формула (Refers to) введите следующую формулу и щелкните на кнопке Добавить (Add):

```
=OFFSET($B$2,0,0,COUNTA($B$2:$B$1000),1)
```




Рис. 6.21. Рабочий лист с итоговым интерфейсом

Применяя этот принцип, можно легко заставить одну или несколько функций баз данных выполнять работу, для которой обычно требуются сотни функций.

Рабочий пример этого упражнения, а также схожие примеры можно загрузить с сайта <http://www.ozgrid.com/download/default.htm>. Открыв страницу, загрузите файл `DFunctionsWithValidation.zip`.

Трюки с макросами

Трюки с 81 по 94

Макросы чудесным образом упрощают автоматизацию повторяющихся задач в Excel, но способ их создания и применения иногда вызывает проблемы. К счастью, Excel — это достаточно гибкое приложение, позволяющее справиться с этими проблемами и создать новые возможности с минимальными усилиями.

Т Р Ю К
№ 81

Ускорение выполнения кода и исключение мерцания экрана

Когда вы записываете макрос в Excel, создаваемый код часто заставляет экран **мерцать**, не только замедляя работу макроса, но и делая его неприятным. К счастью, можно исключить мерцание экрана, одновременно ускорив выполнение макроса.

Один недостаток записанных макросов в Excel заключается в том, что созданный код зачастую является очень неэффективным. Это значит, что макрос, который должен выполняться за секунды, часто работает слишком долго и выглядит непривлекательно. Кроме того, когда вы создаете макрос в режиме записи, записываются все нажатые клавиши, нужны они были или нет. Это означает, что, если вы сделаете ошибку, а потом исправите ее, нажатые для этого **клавиши** также будут записаны в коде макроса.

Если вы уже работали с макросами и интересовались кодом **VBA**, то, вероятно, слышали о свойстве `Application.ScreenUpdating`. Присвоив ему значение Ложь (`False`) в начале выполнения макроса, вы не только прекратите постоянное мерцание экрана, происходящее из-за того, что макрос был записан, а не запрограммирован в **VBE**, но и ускорите выполнение макроса. Причина, почему этот способ ускоряет работу макроса, заключается в том, что Excel более не нужно перерисовывать экран при выполнении таких команд, как `Select`, `Activate`, `LargeScroll`, `SmallScroll` и многих других.

Чтобы включить строку `Application.ScreenUpdating = False` в начало существующего макроса, выберите команду Сервис ▶ Макрос ▶ Макросы (Tools ▶ Macro ▶ Macros), выберите макрос, щелкните кнопку Изменить (Edit) и введите следующий код, показанный в листинге 7.1.

Листинг 7.1

```
'a Macro
'Macro recorded 1/12/2003 by OzGrid.com

Application.ScreenUpdating = False
'ВАШ КОД
Application.ScreenUpdating = True
End Sub
```

Обратите внимание, что в конце кода вы заново включили обновление, присвоив свойству `Application.ScreenUpdating` значение Истина (`True`). Хотя Excel делает это, когда фокус управления будет передан обратно Excel (другими словами, когда работа макроса завершится), лучше все же позаботиться о безопасности программы и включить этот код в конце макроса.

Иногда можно обнаружить, что свойству `Application.ScreenUpdating` присваивается значение Истина (`True`) еще до завершения записанного макроса. Это может произойти с записанными макросами, в которых часто используется команда `Select`. Если это произошло, нужно просто повторить строку `Application.ScreenUpdating = False` в других фрагментах кода вашего макроса.



Запуск макроса в определенное время

Часто вы, наверняка, думаете, как было бы удобно запускать макрос в предопределенное время или через указанные интервалы. К счастью, в Excel предусмотрен метод VBA, который делает это возможным.

При помощи метода `Application.OnTime` можно запускать макрос автоматически, но сначала необходимо провести небольшую настройку. Предположим, есть макрос, который вы хотите запускать каждый день в 15:00 (3:00 p.m.). Сначала нужно определить, как заставить выполняться метод `OnTime`. Это можно сделать с помощью события `Workbook_Open` в частном модуле для объекта `Workbook`.

В Windows самый быстрый способ попасть в частный модуль объекта `Workbook` (`ThisWorkbook`, ЭтаКнига) — правой кнопкой мыши щелкнуть значок Excel рядом с меню Файл (File) и в контекстном меню выбрать команду Исходный текст (View Code). (На Macintosh откройте VBE и затем откройте модуль объекта `Workbook` в окне Project.) Введите код из листинга 7.2.

Листинг 7.2

```
Private Sub Workbook_Open()
    Application.OnTime TimeValue("15:00:00"), "MyMacro"
End Sub
```

Вместо `MyMacro` следует подставить имя макроса, который вы хотите выполнить. Он должен находиться в стандартном модуле и содержать метод `OnTime`, как показано в листинге 7.3.

Листинг 7.3

```
Sub MyMacro()  
    Application.OnTime TimeValue("15:00:00"), "MyMacro"  
    ' ВАШ КОД  
  
End Sub
```

Процедура MyMacro будет запускаться каждый день в 15:00, пока Excel остается открытым.

Теперь предположим, вы хотите выполнять MyMacro каждые 15 минут после открытия книги. И снова вы будете запускать процедуру во время открытия книги, поэтому правой кнопкой мыши щелкните значок Excel рядом с меню Файл (File), выберите команду Исходный текст (View Code) и введите код из листинга 7.4.

Листинг 7.4

```
Private Sub Workbook_BeforeClose(Cancel As Boolean)  
    Application.OnTime dTime, "MyMacro". . False  
End Sub  
  
Private Sub Workbook_Open()  
    Application.OnTime Now + TimeValue("00:15:00"), "MyMacro"  
End Sub
```

В любом стандартном модуле (который можно открыть командой Insert ► Module) введите код, приведенный в листинге 7.5.

Листинг 7.5

```
Public dTime As Date  
Sub MyMacro()  
    dTime = Now + TimeValue("00:15:00")  
    Application.OnTime dTime, "MyMacro"  
  
    ' ВАШ КОД  
End Sub
```

Обратите внимание, что вы передаете значение времени, равное 15 минутам, общей (public) переменной dTime. Это нужно, чтобы вы могли отменить метод OnTime в событии Workbook_BeforeClose, присвоив дополнительному аргументу Schedule значение Ложь (False). По умолчанию его значение равно Истина (True), поэтому, изменив значение на Ложь (False), вы приказываете Excel отменить метод OnTime, настроенный на выполнение в указанное время.

Если не передать время в переменную, Excel не будет знать, какой метод OnTime нужно отменить, так как значение Now + TimeValue(«00:15:00») не является статическим, но становится статическим, когда передается в переменную. Если вы не присвоите дополнительному аргументу Schedule значение Ложь (False), рабочая книга будет автоматически открываться каждые 15 минут, после того как вы закроете ее, и выполнять макрос MyMacro.

Использование кодовых имен для ссылок на листы в рабочих книгах Excel

Иногда бывает нужно создать макрос, который будет работать, даже если имена листов, на которые он ссылается, изменятся.

Если в Excel записан макрос, который ссылается на определенный лист в рабочей книге, известно, что код будет работать, только если имя листа (или листов) останется тем же. Например, если имя рабочего листа — Budget, а код макроса содержит ссылку `Sheets(«Budget»).Select`, при изменении имени рабочего листа макрос перестанет работать. Причина в том, что утилита записи макроса генерирует код, основываясь на имени ярлычка листа или на имени, которое вы видите при работе в Excel.

Для преодоления ограничения есть два варианта. Первый подразумевает использование порядковых номеров. Порядковый номер листа определяется его позицией в рабочей книге. У самого левого листа порядковый номер всегда равен 1, у следующего листа справа — 2 и т. д. Excel VBA позволяет указывать любой лист при помощи его порядкового номера, но, к сожалению, Excel не использует этот метод при записи макроса.

Кроме того, хотя использование порядкового номера листа, например, `Sheets(3).Select` — это гораздо лучший вариант, чем `Sheets(«Budget»).Select`, позиция листа в рабочей книге может измениться, если вы добавите, удалите или переместите листы.

Вместо того чтобы использовать порядковые номера, находчивые программисты на VBA используют кодовые имена (CodeNames). Каждому листу рабочей книги присваивается уникальное кодовое имя, которое не изменяется даже при перемещении или переименовании листа или добавлении новых листов. Кодовое имя листа можно увидеть только в VBE — выберите команду Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor) или нажмите сочетание клавиш `Alt/Option+F11`, — открыв окно Project: выберите команду View ► Project Explorer или нажмите сочетание клавиш `Ctrl+R`.

На рис. 7.1 кодовое имя листа с именем вкладки Budget — это `Sheet3`. Кодовое имя листа — это всегда имя, которое выводится в круглых скобках в окне Project Explorer. В рабочей книге на этот лист можно ссылаться при помощи кода VBA, указав команду `Sheet3.Select`, в противоположность `Sheets(«Budget»).Select` или `Sheets(3).Select`.

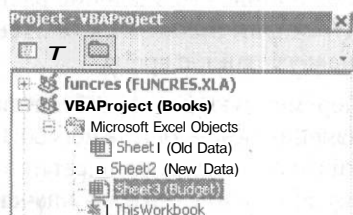


Рис. 7.1. Кодовые имена (CodeNames) и имена листов в окне VBE Project

Если в рабочей книге уже много кода VBA, записанного или написанного в VBE, и в нем не используются кодовые имена, код можно изменить на уровне проекта (то есть весь код во всех модулях рабочей книги), выбрав в VBE команду Edit ▶ Replace.

ВНИМАНИЕ

Единственная ситуация, когда невозможно использовать кодовые имена, — при ссылке на листы, находящиеся в других рабочих книгах, а не в книге, где находится сам код.



Простое подключение кнопок к макросам

Вместо того чтобы назначать каждой кнопке свой макрос, иногда удобнее создать один макрос, управляющий всеми кнопками.

Обычно пользователи предпочитают выполнять макросы, либо нажав сочетание клавиш, либо щелкнув простую кнопку, а не искать нужный макрос в меню и диалоговых окнах. Самый известный способ добраться до кнопки — это панель инструментов Формы (Forms), которую можно открыть, выбрав команду Вид ▶ Панели инструментов ▶ Формы (View ▶ Toolbars ▶ Forms). Эти кнопки, по нашему мнению, представляют собой лучший выбор среди вариантов запуска макросов, особенно записанных макросов, так как такие макросы часто требуют, чтобы в момент выполнения активен был определенный рабочий лист. Проще говоря, записанные макросы всегда используют `ActiveSheet`, если вы записывали макрос, не переключаясь между листами. Это означает, что если пользователь находится не на требуемом листе (другими словами, не на том же листе, на котором вы записывали макрос), то записанный макрос чаще всего выдает ошибку и останавливается или меняет данные на неправильном листе. Благодаря кнопке на рабочем листе, можно заставить пользователя перейти к нужной кнопке на нужном рабочем листе, чтобы настроить подходящие условия для выполнения макроса перед его запуском.

ПРИМЕЧАНИЕ

Почему кнопка с панели инструментов Формы (Forms), а не с панели инструментов Элементы управления (Control Toolbox)? Кнопки практически всегда применяются для распознавания щелчка мыши и затем запуска определенного макроса. Командную кнопку с панели инструментов Элементы управления (Control Toolbox) необходимо использовать, только если требуется распознать другие события, например, двойной щелчок, щелчок правой кнопкой мыши и т. д. Элементы управления на этой панели известны как элементы управления ActiveX, и использование их только для запуска макроса добавляет Excel совершенно ненужную нагрузку, особенно если кнопок много. Это сродни забиванию гвоздей кувалдой.

Если в рабочей книге много кнопок и каждая кнопка используется для запуска определенного макроса, можно присоединить макросы к кнопкам, правой кнопкой мыши щелкнув границу кнопки и в контекстном меню выбрав команду Назначить макрос (Assign Macro). Затем в диалоговом окне Назначить макрос объекту (Assign Macro) выберите подходящий макрос (рис. 7.2).

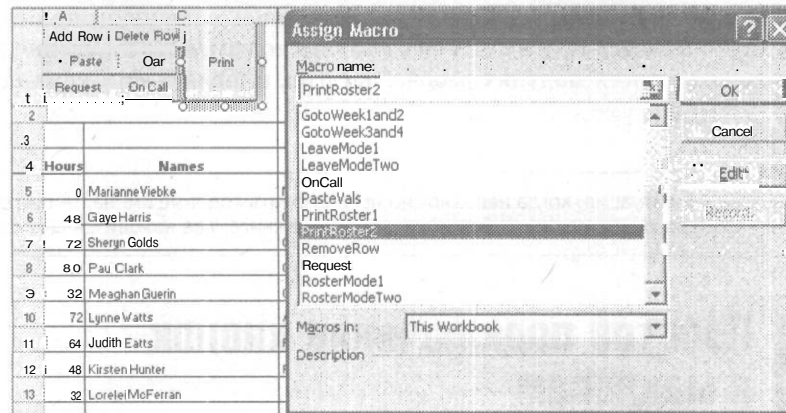


Рис. 7.2. Выделенная кнопка Print и диалоговое окно назначения макроса объекту с выбранным макросом

Каждая кнопка обычно запускает свой макрос, поэтому часто приходится прокручивать целый список макросов, чтобы найти нужный. Есть простой способ назначить все кнопки одному макросу, но при этом сделать так, чтобы каждая кнопка запускала свой макрос.

Поместите следующий код в любой стандартный модуль. Выберите команду Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor) (сочетание клавиш **Alt/Option+F11**) и создайте новый модуль командой Insert ► Module. Затем введите код из листинга 7.6.

Листинг 7.6

```
Sub WhichButton()
    Run Application.Caller
End Sub
```

Теперь необходимо присвоить каждой кнопке то же имя, что и у макроса, который она должна запускать. Чтобы дать имя кнопке с панели инструментов Формы (Forms), щелкните ее и замените имя в поле имени названием макроса, который будет запускать эта кнопка. Сделайте это для всех кнопок. Теперь, если вы назовете кнопку `Macro1` и назначите ее макросу `WhichButton`, щелчок на этой кнопке запустит макрос `Macro1`.

Т Р Ю К
№85

Создание всплывающего экрана для рабочей книги

Всплывающие экраны придают приложению дополнительную каплю изысканности, не говоря уж о том, что они развлекают вас, пока приложение загружается. Почему бы рабочему листу не делать то же самое?

Для создания всплывающего экрана для любой книги можно воспользоваться возможностями VBA. Вы увидите, что процесс намного проще, чем вы могли подумать.

Чтобы создать всплывающий экран, который отображается 5–10 секунд, пока рабочая книга открывается, а затем автоматически закрывается, нажмите сочетание клавиш **Alt/Option+F11** или выберите команду Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor), чтобы открыть VBE. Затем выберите команду Insert ► UserForm. Если панель инструментов Control не отображается, выберите команду View ► Toolbox, чтобы ее увидеть.

На панели инструментов щелкните элемент управления Label. (Чтобы увидеть название каждого элемента управления, проведите над ними указателем мыши и задержите над каждым непродолжительное время.) Щелкните в любом месте объекта UserForm, чтобы вставить метку. Перемещая маркеры размера, выберите такой размер метки, чтобы в нее можно было ввести небольшой текст. Не снимая выделения метки, щелкните ее. Если метка не выделена, дважды медленно щелкните ее. Теперь она должна войти в режим редактирования, в котором можно выделить надпись по умолчанию Label1.

На этой метке введите текст My Splash Screen. Чтобы изменить другие свойства метки (размер шрифта, цвет и т. п.), удостоверьтесь, что метка выделена, и нажмите клавишу F4 или выберите команду View ► Properties Window. Затем в окне Label Controls Property измените нужные свойства метки. После этого дважды щелкните объект UserForm (но не метку) и в окне Procedure в верхнем правом углу экрана выберите Initialize (рис. 7.3).

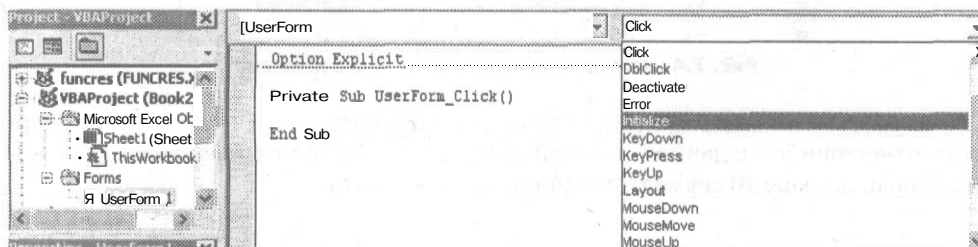


Рис. 7.3. Раскрывающийся список Procedure с различными событиями объекта UserForm

В этой процедуре введите следующий код:

```
Application.OnTime Now + TimeValue("00:00:05"), "KillForm"
```

Код для объекта UserForm должен выглядеть так, как показано в листинге 7.7.

Листинг 7.7

```
Private Sub UserForm_Initialize()  
    Application.OnTime Now + TimeValue("00:00:05"), "KillForm"  
End Sub
```

Теперь выберите команду Insert ► Module и введите код из листинга 7.8.

Листинг 7.8

```
Sub KillForm()  
    Unload UserForm1  
End Sub
```

Теперь осталось добавить немного кода в частный модуль объекта Workbook (ThisWorkbook, ЭтаКнига). В окне Project Explorer вы должны видеть имя рабочей книги. Разворачивайте папки, начинающиеся ниже рабочей книги, пока не увидите ThisWorkbook (ЭтаКнига) в папке Microsoft Excel Objects. Дважды щелкните пункт ThisWorkbook (ЭтаКнига), чтобы раскрыть ее частный модуль.

В частном модуле объекта ThisWorkbook (ЭтаКнига) введите код из листинга 7.9.

Листинг 7.9

```
Private Sub Workbook_Open()
    UserForm1.Show
End Sub
```

Закройте окно, чтобы вернуться в Excel. Сохраните и закройте рабочую книгу, а затем откройте ее заново, чтобы увидеть всплывающий экран в действии. На рис. 7.4 показан пример всплывающего экрана.

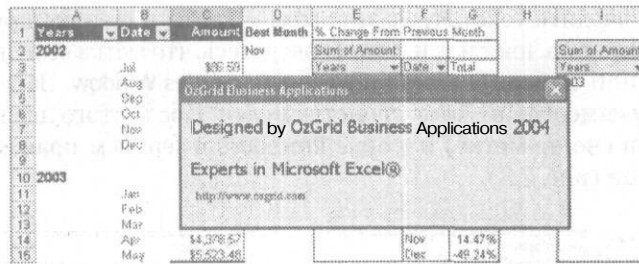


Рис. 7.4. Пример всплывающего экрана в действии

Помните, что всплывающий экран должен отображаться совсем недолго и содержать относящийся к данной рабочей книге текст. Если пользователи будут видеть экран дольше 10 секунд, он начнет их раздражать.

Т Р Ю К
№ 86

Отображение сообщения «Please Wait»

Вы встречались с макросами, которые, казалось, никогда не закончатся? Если такая проблема возникает с вашим макросом, можно сделать так, чтобы во время работы макроса выводилось сообщение «Please Wait».

Большинство пользователей ожидает, что код будет выполняться и завершаться практически мгновенно. К сожалению, так бывает не всегда. Записанные макросы особенно часто выполняются намного дольше, чем правильно написанный код VBA. Хуже того, код VBA в Excel в целом выполняется медленнее, чем большинство других видов кода.

К счастью, можно написать еще немного кода и создать сообщение «Please Wait», чтобы пользователи знали, что код выполняется и Excel не завис во время работы макроса. К сожалению, один из наиболее популярных способов сообщить пользователям, что код выполняется, — использовать индикатор хода выполнения Excel.

В этом случае возникают две проблемы. Во-первых, индикатор хода выполнения может еще более замедлить ваш код, ухудшая ситуацию. Во-вторых, возможно, медленная работа кода вызвана циклом, а использовать утилиту записи макроса для создания цикла невозможно.

Мы предпочитаем код VBA, например, следующий макрос `DoIt`, который использует прямоугольник с панели инструментов Рисование (Drawing). Код его приведен в листинге 7.10.

Листинг 7.10

```
Sub DoIt()  
Application.ScreenUpdating = True  
With Sheet1.Shapes("Rectangle1")  
    .Visible = msoTrue = (Not Sheet1.Shapes("Rectangle1").Visible)  
End With  
'Переключение листов заставляет отображаться Rectangle1 во время выполнения  
'кода  
Sheet2.Select  
Sheet1.Select  
End Sub
```

Чтобы воспользоваться этим кодом, вставьте прямоугольник с панели инструментов Рисование (Drawing) на любой лист нужной рабочей книги. Не снимая выделения прямоугольника, щелкните поле Имя (Name) и введите имя прямоугольника — `Rectangle1` (если ему еще не присвоено такое имя).

Введите текст, который должен отображаться во время выполнения кода, и отформатируйте, переместите и измените размер прямоугольника. Введите предыдущий макрос `DoIt` в стандартный модуль вашей рабочей книги. Если нужно, замените `Sheet1` в коде на кодовое имя листа, на который вы поместили `Rectangle1`. (Подробнее о кодовых именах — в разделе «Трюк № 83. Использование кодовых имен для ссылки на листы в рабочих книгах Excel».) Теперь выберите команду Сервис ► Макрос ► Макросы (Tools ► Macro ► Macros) и запустите макрос `DoIt`. Он полностью скроет `Rectangle1`.

В самом начале медленно выполняющегося кода поместите код из листинга 7.11.

Листинг 7.11

```
Run "DoIt"  
Application.ScreenUpdating = False
```

Благодаря строке `Application.ScreenUpdating = False` экран перестанет мерцать, и скорость выполнения макроса увеличится. В конце этого кода поместите строку `Run "DoIt"`. Затем запустите макрос, как обычно.

Т Р Ю К
№ 87

Как помечать и сбрасывать пометку ячеек при выделении

Иногда бывает трудно делать выбор при помощи флажков. К счастью, этот процесс можно упростить, написав немного кода.

При помощи рабочих книг Excel можно собирать данные для опросов. Обычно это делается путем вывода некоторого количества ответов, рядом с которыми стоят флажки. Пользователь выбирает нужные ответы, устанавливая соответствующие флажки. Проблема с применением этого метода заключается в том, что скоро в рабочей книге появятся сотни флажков.

Вместо этого можно написать немного очень простого кода VBA, чтобы помечать любую ячейку в указанном диапазоне одновременно с ее выделением. Если ячейка в указанном диапазоне уже помечена, код снимет пометку. Трюк этого кода заключается в использовании буквы «а» в ячейке, для которой установлен шрифт Marlett. Когда настанет время суммировать результаты, нужно будет просто подсчитать буквы «а» при помощи функции СЧЁТЕСЛИ (COUNTIF), например, так:

```
=COUNTIF($A$1:A$100,"a")
```

В русской версии Excel:

```
=СЧЁТЕСЛИ($A$1:A$100,"a")
```

Следующие примеры кода работают только в диапазоне A1:A100, но их можно легко модифицировать для любого диапазона. Чтобы использовать этот код, перейдите на рабочий лист, на котором должны появиться пометки, правой кнопкой мыши щелкните ярлычок с именем листа и в контекстном меню выберите команду Исходный текст (View Code). Вставьте либо КОД 1 (если хотите, чтобы ячейка помечалась в момент, когда ее выделяют), либо КОД 2 (если хотите, чтобы ячейка помечалась двойным щелчком), приведенный в листинге 7.12.

Листинг 7.12

'КОД1 - установить флажок одновременно с выделением

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If Target.Cells.Count > 1 Then Exit Sub
    If Not Intersect(Target, Range("A1:A100")) Is Nothing Then
        Target.Font.Name = "Marlett"
        If Target = vbNullString Then
            Target = "a"
        Else
            Target = vbNullString
        End If
    End If
End Sub
```

'КОД2 - установить флажок двойным щелчком

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
    If Not Intersect(Target, Range("A1:A100")) Is Nothing Then
        Cancel = True 'Предотвращает переход в режим редактирования
        Target.Font.Name = "Marlett"
        If Target = vbNullString Then
            Target = "a"
        Else
            Target = vbNullString
        End If
    End If
End Sub
```

```
End If
End If
End Sub
```

Поместив нужный код на место, закройте окно, чтобы вернуться в Excel, и сохраните рабочую книгу. Если хотите проверить, помечена ли ячейка, проверьте ее содержимое.

Т Р Ю К
№88

Подсчет или суммирование ячеек с определенным цветом заливки

При помощи небольшого фрагмента кода можно легко применять функции СУММ (SUM) и СЧЁТ (COUNT) к ячейкам, цвет заливки которых был указан вручную.

Очень часто бывает удобно суммировать или подсчитывать ячейки с определенным цветом заливки, который вы или другой пользователь выбрали вручную, так как пользователям зачастую удобнее работать с цветовыми заливками, чем с именованными диапазонами. Чтобы сделать это, сначала откройте рабочую книгу, в которой хотите суммировать или подсчитать ячейки в зависимости от цвета заливки. Перейдите в VBE, выбрав команду Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor) (сочетание клавиш Alt/Option+F11), и выберите команду Insert ► Module, чтобы вставить стандартный модуль. В этом модуле введите код из листинга 7.13.

Листинг 7.13

```
Function ColorFunction(rColor As Range, rRange As Range, Optional SUM _
    As Boolean)
Dim rCell As Range
Dim lCol As Long
Dim vResult

lCol = rColor.Interior.ColorIndex

If SUM = True Then
    For Each rCell In rRange
        If rCell.Interior.ColorIndex = lCol Then
            vResult = WorksheetFunction.SUM(rCell) + vResult
        End If
    Next rCell
Else
    For Each rCell In rRange
        If rCell.Interior.ColorIndex = lCol Then
            vResult = 1 + vResult
        End If
    Next rCell
End If

ColorFunction = vResult
End Function
```

Теперь при помощи пользовательской функции ColorFunction можно суммировать значения в ячейках диапазона \$A\$1:\$A\$12, имеющих такой же цвет заливки, что и ячейка \$C\$1:

```
=ColorFunction($C$1,$A$1:$A$12,TRUE)
```

В русской версии Excel

```
=ColorFunction($C$1,$A$1:$A$12,ИСТИНА)
```

Эта функция будет выполнять суммирование, так как в качестве последнего аргумента пользовательской функции указано ИСТИНА (TRUE).

Чтобы подсчитать ячейки с такой же цветовой заливкой, что и ячейка \$C\$1, можно воспользоваться функцией

```
=ColorFunction($C$1,$A$1:$A$12,FALSE)
```

В русской версии Excel

```
=ColorFunction($C$1,$A$1:$A$12,ЛОЖЬ)
```

или

```
=ColorFunction($C$1,$A$1:$A$12)
```

Так как последний аргумент не указан, функция по умолчанию использует значение ЛОЖЬ (FALSE) в качестве последнего аргумента. Теперь можно легко суммировать и подсчитывать ячейки с указанным цветом заливки (рис. 7.5).

	A	B	C	D	E
1	20				
2	22				
3	23				
4	25				
5	26				
6	27				
7	29				
8	30				
9	31				
10	32				
11	54				
12	21				
13	78	=ColorFunction(\$C\$1,\$A\$1:\$A\$12,TRUE)			
14	3	=ColorFunction(\$C\$1,\$A\$1:\$A\$12)			
15					

Рис. 7.5. Подсчет ячеек по цвету при помощи пользовательской функции ColorFunction

ТРЮК
№ 89

Добавление элемента управления Календарь в любую рабочую книгу Excel

Для гарантии правильности ввода дат пользователями элемент управления Excel Календарь (Calendar) может упростить ситуацию как для вас, так и для пользователей, работающих с вашей электронной таблицей. При помощи этого трюка можно добавить элемент управления Календарь (Calendar) в любую рабочую книгу Excel.

Если дата введена неправильно, Excel не сможет ее распознать. Иногда это означает, что вы не можете выполнять вычисления с числами, выглядящими как даты, но датами не являющимися. Кроме того, это означает, что любые диаграммы или сводные таблицы, которые основываются на этих данных, будут неверными. Хотя решить эту проблему может весьма разностороннее средство проверки данных Excel (которое мы рассматривали в главе 2), оно также далеко от совершенства.

При помощи этого трюка можно добавить элемент управления Календарь (Calendar) в любую рабочую книгу Excel. Для начала откройте рабочую книгу, куда собираетесь добавить календарь. Лучше использовать ваш файл Personal.xls, который можно открыть командой Окно ▶ Отобразить (Window ▶ Unhide). Если этот пункт меню недоступен, значит, у вас пока нет файла Personal.xls. Можно создать его, записав пустой макрос. Для этого выберите команду Сервис ▶ Макрос ▶ Начать запись (Tools ▶ Macro ▶ Record New Macro) и в поле Сохранить в (Store Macro In) выберите вариант Личная книга макросов (Personal Macro Workbook). Затем щелкните на кнопке ОК, выделите любую ячейку и остановите запись. Excel автоматически создаст файл Personal.xls.

Выберите команду Сервис ▶ Макрос ▶ Редактор Visual Basic (Tools ▶ Macro ▶ Visual Basic Editor) (сочетание клавиш Alt/Option+F11) и в VBE выберите команду Insert ▶ UserForm. Должна появиться панель инструментов Control. (Если панель не открылась, выберите команду View ▶ Toolbox.)

Правой кнопкой мыши щелкните панель инструментов Control и в контекстном меню выберите команду Additional Controls. Прокрутите список до флажка Элемент управления Календарь 10.0 (Calendar Control 10.0) (номер версии зависит от версии Excel). Установите этот флажок и щелкните на кнопке ОК. Щелкните календарь, который теперь представляет собой часть панели инструментов, а затем щелкните форму UserForm, которую добавили ранее.

При помощи маркеров изменения размера формы UserForm и элемента управления Календарь (Calendar) отрегулируйте их размер (рис. 7.6).

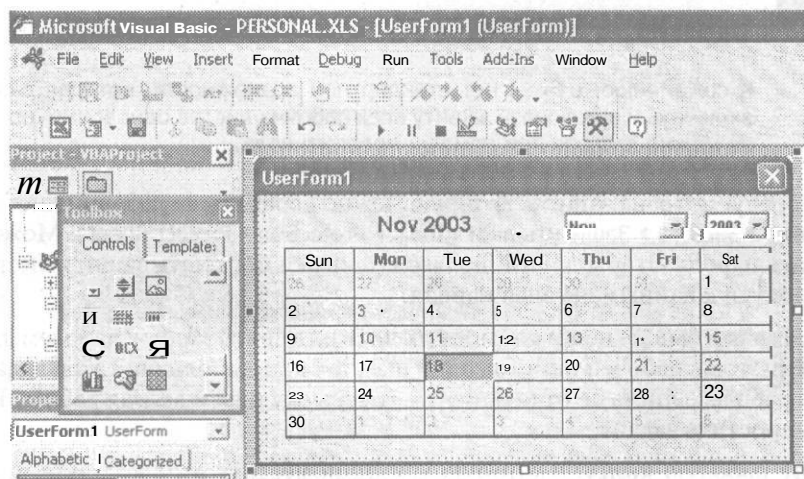


Рис. 7.6. Добавленный элемент управления Календарь

Удостоверьтесь, что форма UserForm выделена (см. рис. 7.6), и выберите команду View ► Properties Window (клавиша F4). В окне Properties выберите Caption и замените слово UserForm1 словом Calendar. Теперь выберите команду View ► Code (клавиша F7) и в частном модуле введите код, приведенный в листинге 7.14.

Листинг 7.14

```
Private Sub Calendar1_Click()  
ActiveCell = Calendar1.Value  
End Sub
```

```
Private Sub UserForm_Activate()  
Me.Calendar1.Value = Date  
End Sub
```

Выберите команду Insert ► Module и в общем модуле введите код из листинга 7.15.

Листинг 7.15

```
Sub ShowIt()  
UserForm1.Show  
End Sub
```

Закройте окно, чтобы вернуться в Excel. Выберите команду Сервис ► Макрос ► Макросы (Tools ► Macro ► Macros) (сочетание клавиш Alt+F8) и затем выберите Show-It. Щелкните кнопку Параметры (Options) и назначьте макросу клавишное сочетание — все готово. Нажмите выбранное сочетание клавиш, раскроется календарь с выбранной сегодняшней датой. Щелкните любую дату, и она будет вставлена в активную ячейку.

**ТРЮК
№ 90**

Включение и снятие защиты паролем для всех рабочих листов Excel одновременно

К сожалению, в Excel нет стандартной возможности, которая разрешает включать и выключать защиту всех рабочих листов одновременно; однако ее можно создать при помощи простого кода.

В Excel предусмотрена защита, которую можно добавить на рабочий лист командой Сервис ► Защита ► Защитить лист (Tools ► Protection ► Protect Sheet). Можно также указать пароль, чтобы другой пользователь не смог снять защиту листа и получить доступ к данным, не зная пароля.

Иногда, однако, бывает необходимо включить или снять защиту паролем для всех рабочих листов в рабочей книге за один шаг, так как выполнение этой задачи для каждого рабочего листа по отдельности — очень скучная и долгая работа. Вот как можно упростить задачу.

Откройте рабочую книгу, в которую хотите поместить этот код, или выберите команду Окно ► Отобразить (Window ► Unhide) и отобразите файл Personal.xls, чтобы

макрос был доступен в любой рабочей книге. Выберите команду Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor) (сочетание клавиш Alt/Option+F11) и в VBE выберите команду Insert ► UserForm. Должна появиться панель инструментов Control. Если панель не открылась, выберите команду View ► Toolbox.

На панели инструментов выберите текстовое поле (TextBox, обозначенное как ab|). Щелкните форму UserForm, чтобы добавить на нее текстовое поле. Поместите поле в левый верхний угол формы и отрегулируйте его размер.

Удостоверьтесь, что текстовое поле выделено, и выберите команду View ► Properties (клавиша F4). Прокрутите список в окне Properties для этого текстового поля, пока не увидите пункт PasswordChar, и в белом поле справа введите звездочку (*); На панели инструментов выберите элемент CommandButton, щелкните форму UserForm и поместите командную кнопку в правом верхнем углу формы.

Не снимая выделения кнопки CommandButton, выберите команду View ► Properties (клавиша R). Прокрутите список в окне Properties для командной кнопки, пока не увидите пункт Caption, и в белом поле справа введите текст ОК. Если вы пользуетесь Excel 97, прокрутите еще дальше до пункта TakeFocusOnClick и выберите для этого свойства значение False.

Теперь выделите форму UserForm и в окне Properties для нее найдите пункт Caption. Измените значение этого свойства на Protect/Unprotect all sheets. Теперь форма должна выглядеть, как на рис. 7.7.

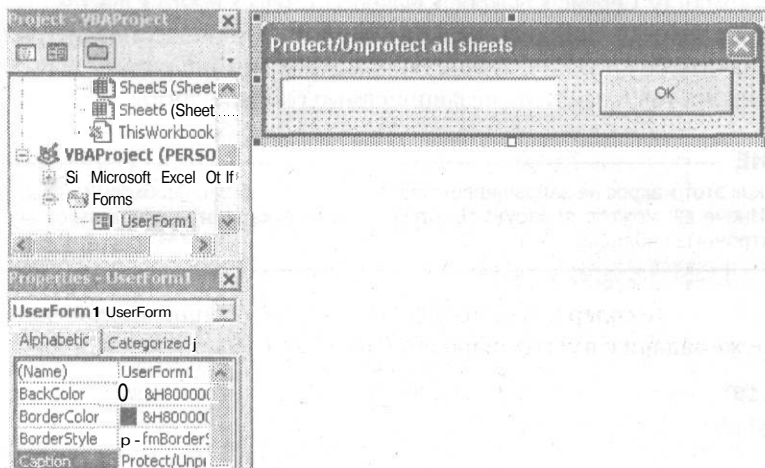


Рис. 7.7. Форма UserForm в VBE

Выберите команду View ► Code (клавиша F7) и введите код, показанный в листинге 7.16.

Листинг 7.16

```
Private Sub CommandButton1_Click()  
Dim wSheet As Worksheet  
For Each wSheet In Worksheets
```

продолжение ↗

Листинг 7.16 (продолжение)

```

If wSheet.ProtectContents = True Then
    wSheet.Unprotect Password:=TextBox1.Text
Else
    wSheet.Protect Password:=TextBox1.Text
End If
Next wSheet
Unload me
End Sub

```

Этот код обрабатывает все рабочие листы активной рабочей книги. Если один из них защищен, он снимает защиту, используя пароль, который вы ввели в текстовое поле. Если защита рабочего листа уже снята, макрос защищает его, используя пароль в текстовом поле.

Теперь выберите команду Insert ► Module и введите код из листинга 7.17.

Листинг 7.17

```

Sub ShowPass()
    UserForm1.Show
End Sub

```

Он необходим, чтобы открыть форму UserForm. Закройте окно, чтобы вернуться в Excel.

Выберите команду Сервис ► Макрос ► Макросы (Tools ► Macro ► Macros) (сочетание клавиш Alt/Option+F8). Выберите ShowPass и щелкните кнопку Параметры (Options), чтобы назначить сочетание клавиш. Этот макрос снимет защиту всех защищенных рабочих листов и защитит незащищенные рабочие листы.

ВНИМАНИЕ

Так как этот макрос не запрашивает подтверждения пароля, необходимо вводить его точно. Иначе вы можете обнаружить, что одна опечатка смогла спрятать от вас все ваши электронные таблицы.

Если вы защищаете содержимое только от себя, следующий макрос позволит выполнять те же задачи с пустым паролем (листинг 7.18).

Листинг 7.18

```

Option Explicit

Sub Protect_Unprotect()
    Dim wSheet As Worksheet

    For Each wSheet In Worksheets
        With wSheet
            If .ProtectContents = True Then
                .Unprotect Password:=""
            Else
                .Protect Password:=""
            End If
        End With
    Next wSheet
End Sub

```

End With
Next wSheet

End Sub

Хотя это не очень безопасно, этот макрос, безусловно, удобен.

Т Р Ю К
№ 91

Получение имени и пути рабочей книги

Иногда бывает необходимо, чтобы ячейка могла вернуть имя рабочей книги или даже имя файла и путь для этой рабочей книги. При помощи этого трюка вы легко получите имя и путь рабочей книги.

Три пользовательские функции, с которыми мы познакомимся в этом разделе, помещают в ячейку имя рабочей книги или имя файла и путь для этой рабочей книги. Два первых примера, `MyName` и `MyFullName`, не требуют никаких аргументов. Последний, `SheetName`, используется, чтобы при получении имени листа не вкладывать ПСТР (MID) и другие функции в функцию ЯЧЕЙКА (CELL), так как это обычно требует использования следующей громоздкой формулы:

```
=MID(CELL("filename", $A$1), FIND("]", CELL("filename", $A$1))+1, 255)
```

В русской версии Excel

```
=ПСТР(ЯЧЕЙКА("имяфайла", $A$1), НАЙТИ("]", ЯЧЕЙКА("имяфайла", $A$1))+1, 255)
```

Как видно, для получения простейшего результата приходится довольно много потрудиться, поэтому мы самостоятельно разработали пользовательскую функцию `SheetName`.

Чтобы применить эти пользовательские функции, нажмите сочетание клавиш **Alt/Option+F11**, выберите команду **Insert ▶ Module** и вставьте код, приведенный в листинге 7.19.

Листинг 7.19

```
Function MyName() As String
MyName = ThisWorkbook.Name
End Function
Function MyFullName() As String
MyFullName = ThisWorkbook.FullName
End Function
Function SheetName(rAnyCell)
Application.Volatile
SheetName = rAnyCell.Parent.Name
End Function
```

Сохраните функцию и закройте окно. Теперь эта функция появится в области **Определенные пользователем (User Defined)** диалогового окна **Мастер функций (Paste Function)** (сочетание клавиш **Shift+F3**).

Эти функции используются, как показано на рис. 7.8. Они не требуют аргументов. Формулы в столбце А показаны исключительно в демонстрационных целях и никак не влияют на результат.

	A	B
1	Formula	Result
2	=MyName()	Workbook Path.xls
3	=MyFullName()	C:\OzGrid\Learning\Workbook Path.xls
4	=CELL("filename")	C:\OzGrid\Learning\Workbook Path.xls]Sheet1
5	=sheetname(A1)	Sheet1

Рис. 7.8. Функции и результат их выполнения

ПРИМЕЧАНИЕ

В ячейке A4 на рис. 7.8 мы также поместили результат, возвращенный стандартной функцией ЯЧЕЙКА (CELL). Она возвращает имя рабочей книги, путь к файлу и имя активного листа. Функция ЯЧЕЙКА (CELL) — это стандартная функция, возвращающая информацию о текущей операционной системе, — другими словами, информацию о форматировании, местоположении и содержимом рабочей книги.

Т Р Ю К
№ 92

Как обойти ограничение Excel на три критерия условного форматирования

Чтобы при условном форматировании использовать больше трех критериев для данных, можно применить трюк с использованием VBA. В действительности, благодаря этому коду, можно использовать практически неограниченное количество критериев.

В Excel есть очень полезная возможность под названием условное форматирование (которое подробнее рассматривается в главе 2). Чтобы воспользоваться ею, нужно выбрать команду **Формат** ► **Условное форматирование** (Format ► Conditional Formatting) на панели меню рабочего листа. Эта возможность позволяет форматировать ячейки в зависимости от их содержимого. Например, можно изменить фоновый цвет всех ячеек, значение в которых больше 5, но меньше 10, на красный. Хотя это удобно, Excel поддерживает только три условия, которых иногда не хватает.

Указать более трех условий можно благодаря коду Excel VBA, который запускается автоматически, когда пользователь изменяет указанный диапазон. Чтобы увидеть, как это работает, предположим, есть шесть отдельных условий в диапазоне A1:A10 на определенном рабочем листе. Введите некоторые данные (рис. 7.9).

Сохраните рабочую книгу, перейдите на рабочий лист, правой кнопкой щелкните ярлычок с его именем, в контекстном меню выберите команду **Исходный текст** (View Code) и введите код из листинга 7.20.

Листинг 7.20

```
Private Sub Worksheet_Change(ByVal Target As Range)
```

```
Dim iColor As Integer
```

```
If Not Intersect(Target, Range("A1:A10")) Is Nothing Then
```

```

Select Case Target
Case 1 To 5
icolor = 6
Case 6 To 10
icolor = 12
Case 11 To 15
icolor = 7
Case 16 To 20
icolor = 53
Case 21 To 25
icolor = 15
Case 26 To 30
icolor = 42
Case Else
'Whatever
End Select
Target.Interior.ColorIndex = icolor
End If

End Sub
    
```

Закройте окно, чтобы вернуться на рабочий лист. Результат должен выглядеть, как на рис. 7.10.

	A
1	Numbers
2	3
3	
4	8
5	
6	13
7	
8	18
9	23
10	28
11	3
12	7
13	12
14	17
15	23
16	28

Рис. 7.9. Данные для эксперимента с условным форматированием

	A
1	Numbers
2	3
3	
4	
5	
6	13
7	
8	
9	23
10	
11	3
12	7
13	12
14	17
15	23
16	28

Рис. 7.10. Данные после ввода кода

Фоновый цвет каждой ячейки должен измениться в зависимости от числа, переданного переменной icolor, которая, в свою очередь, передает это число Target.Interior.ColorIndex. Передаваемое число определяется строкой Case x To x. Например, если вы введете число 22 в любую ячейку в диапазоне A1:A10, то переменной icolor будет передано число 15, которое затем эта переменная (теперь имеющая значение 15) передает Target.Interior.ColorIndex, делая ячейку серой. Целью всегда является ячейка, значение в которой было изменено, что и вызвало запуск кода.

Выполнение процедур на защищенных рабочих листах

Макросы Excel представляют собой замечательный способ экономии времени и исключения ошибок. Однако рано или поздно вы пытаетесь выполнить ваш любимый макрос Excel на рабочем листе, к которому была применена защита, с паролем или без, и получаете ошибку выполнения. Избежать ее можно при помощи следующего трюка.

Если вы когда-либо пытались выполнить макрос Excel на защищенном рабочем листе, то знаете, что, как только этот рабочий лист упоминается в макросе, макрос, вероятнее всего, перестает работать и появляется ошибка выполнения.

Один из способов обойти это — использовать код, например, следующий, чтобы снять защиту, а потом снова защитить рабочий лист, как это показано в листинге 7.21.

Листинг 7.21

```
Sub MyMacro()
```

```
Sheet1.Unprotect Password:="Secret"
```

```
' ВАШ КОД
```

```
Sheet1.Protect Password:="Secret"
```

```
End Sub
```

Как можно видеть, этот код снимает защиту листа Sheet1 с паролем Secret, выполняет код, а затем снова включает защиту паролем. Этот код работает, но у него есть несколько недостатков. Например, он может вызвать ошибку и остановиться до того, как дойдет до строки кода `Sheet1.Protect Password:="Secret"`. Это, конечно же, приведет к тому, что рабочий лист останется незащищенным. Еще один недостаток заключается в том, что потребуется похожий код для всех макросов и всех рабочих листов.

Еще один способ обойти проблему — использовать `UserInterfaceOnly`, необязательный аргумент метода `Protect`, которому можно присвоить значение `True` (значение по умолчанию равно `False`). Если значение этого аргумента равно `True`, Excel позволяет всем макросам Excel VBA выполняться на рабочих листах, защищенных с паролем или без пароля.

Однако если вы используете на рабочем листе метод `Protect`, аргумент `UserInterfaceOnly` которого равен `True`, а затем сохраните рабочую книгу, весь рабочий лист (а не только интерфейс) будет полностью защищен при очередном открытии рабочей книги. Чтобы снова присвоить аргументу `UserInterfaceOnly` значение `True` после открытия рабочей книги, необходимо еще раз применить метод `Protect`, присваивающий истинное значение `UserInterfaceOnly`.

Чтобы избежать неразберихи, используйте событие `Workbook_Open`, которое выполняется при открытии рабочей книги. Так как это событие принадлежит объекту `Workbook ЭтаКнига (ThisWorkbook)`, необходимо поместить следующий код

в частный модуль объекта ЭтаКнига (ThisWorkbook). Для этого в версиях Excel для Windows правой кнопкой мыши щелкните значок Excel и в контекстном меню выберите команду Исходный текст (View Code). На Macintosh откройте объект Workbook в окне Projects в VBE. Затем введите код, показанный в листинге 7.22.

Листинг 7.22

```
Private Sub Workbook_Open()  
    'Если у вас разные пароли  
    'для всех рабочих листов.  
  
    Sheets(1).Protect Password:="Secret", UserInterfaceOnly:=True  
    Sheets(2).Protect Password:="Carrot", UserInterfaceOnly:=True  
  
    "Повторить необходимое количество раз."  
End Sub
```

Предыдущий код подходит для случаев, когда для всех рабочих листов, на которых вы хотите выполнить этот макрос, выбраны разные пароли или если вы не хотите защищать все рабочие листы. Можно присвоить аргументу `UserInterfaceOnly` значение `True`, не снимая перед этим защиту листа.

Если вы хотите присвоить аргументу `UserInterfaceOnly` значение `True` для всех рабочих листов и у всех рабочих листов одинаковые пароли, используйте следующий код, который нужно поместить туда же, куда и предыдущий.

Листинг 7.23

```
Private Sub Workbook_Open()  
    Dim wSheet As Worksheet  
  
    For Each wSheet In Worksheets  
        wSheet.Protect Password:="Secret", UserInterfaceOnly:=True  
    Next wSheet  
End Sub
```

Теперь каждый раз при открытии рабочей книги будет выполняться этот код, присваивая свойству `UserInterfaceOnly` значение `True` и позволяя макросу работать, одновременно запрещая любые другие изменения.



Распространение макросов

Хотя макросы можно распространять вместе с рабочими книгами, если вы хотите распространить только функциональность макроса, необходимо использовать возможности надстроек Excel.

Надстройка Excel — это не что иное, как рабочая книга Excel, сохраненная как надстройка при помощи параметра **Файл** ▶ **Сохранить как** > **Надстройка Microsoft Excel (*.xla)** (**File** ▶ **Save As** ▶ **Microsoft Excel Add-in (*.xla)**). Когда вы сохраните и заново откроете эту рабочую книгу, она станет скрытой и вы сможете увидеть ее только

в Project Explorer в VBE. Она скрыта не так, как файл Personal.xls, так как его можно увидеть (и отобразить) командой Окно ► Отобразить (Windows ► Unhide).

Закончив настройку рабочей книги, которую хотите использовать как надстройку, необходимо сохранить ее копию. Ее можно записать в любую папку, однако стоит запомнить, куда именно вы сохранили копию.

Откройте любую рабочую книгу и в меню Сервис (Tools) выберите команду Надстройки (Add-Ins), а затем щелкните кнопку Обзор (Browse). Найдите свою надстройку, выделите ее и щелкните кнопку ОК.

Удостоверьтесь, что теперь ваша надстройка есть в поле Список надстроек (Add-Ins Available) и соответствующий ей флажок установлен. Щелкните кнопку ОК, чтобы установить надстройку. Большую часть кода можно сохранить в надстройке Excel без особых изменений, однако необходимо обратить внимание на следующее:

- Объект ЭтаКнига (ThisWorkbook) всегда ссылается на надстройку, а не на рабочую книгу пользователя. Вместо него используйте ActiveWorkbook.
- В ActiveWorkbook нельзя ссылаться на листы при помощи кодовых имен (CodeName).
- Необходимо всегда восстанавливать исходную настройку панелей инструментов и т. п. Нет ничего хуже, чем **надстройка**, изменяющая все ваши настройки Excel без вашего ведома.
- Всегда включайте какую-нибудь обработку ошибок (да, большинство надстроек когда-нибудь вызовут ошибки).
- Помните, что пользователь может применить различные типы защиты к своим данным. *Никогда* не снимайте защиту любой части рабочей книги пользователя при помощи кода. Просто выводите сообщение с просьбой отключить защиту.
- Активно пользуйтесь рабочим листом, который присутствует в надстройке. Мы применяем рабочий лист (или листы) для хранения настроек пользователя, например, панелей инструментов.
- Удержание клавиши Shift не сможет запретить выполнение событий надстройки рабочей книги (однако, удерживая ее, можно запретить выполнение обычного файла .xls).
- Если необходимо просмотреть, обновить или модифицировать рабочую книгу надстройки, когда надстройка установлена, перейдите в VBE и в окне Properties выберите свойство IsAddin. Присвойте этому свойству значение False. При сохранении рабочей книги как надстройки это свойство получает значение True.
- Примените защиту к модулям надстройки, выбрав команду Tools ► VBAProject Properties ► Protection.

После того как надстройка создана, необходимо, чтобы пользователи могли легко выполнять макрос внутри нее. Для этого лучше всего использовать события Workbook_AddinInstall и Workbook_AddinUnInstall в частном модуле объекта ЭтаКнига (ThisWorkbook). Дважды щелкните объект ЭтаКнига (ThisWorkbook) для файла *.xla и вы попадете в частный модуль, куда следует поместить этот код (рис. 7.11).

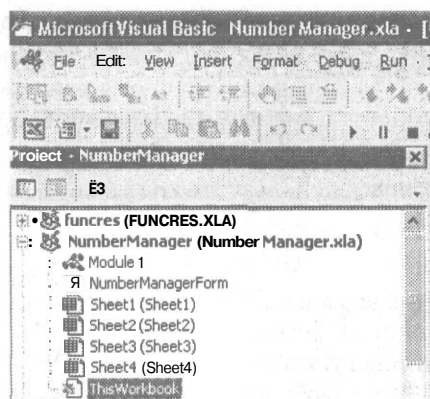


Рис. 7.11. Окно Project Explorer, в котором выбран объект ЭтаКнига (ThisWorkbook) для надстройки с именем NumberManager.xla

Простой пример кода приведен в листинге 7.24.

Листинг 7.24

```
Option Explicit
Dim cControl As CommandBarButton
Private Sub Workbook_AddinInstall()

On Error Resume Next 'На всякий случай
'Удалить любой существующий элемент меню, который мог остаться.
Application.CommandBars("Worksheet Menu Bar").Controls("Super _
Code").Delete
'Добавить новый элемент меню и присвоить его переменной CommandBarButton
Set cControl = Application.CommandBars("Worksheet Menu Bar").Controls.Add
"Работа с переменной
With cControl
.Caption = "Super Code"
.Style = msoButtonCaption
.OnAction = "MyGreatMacro" 'Макрос, хранимый в стандартном модуле
End With
On Error GoTo 0

End Sub

Private Sub Workbook_AddinUninstall()

On Error Resume Next 'На случай, если его уже нет.
Application.CommandBars("Worksheet Menu Bar").Controls("Super Code").Delete
On Error GoTo 0

End, Sub
```

Это весь код, который понадобится для добавления одного элемента меню (с именем Super Code) в конец существующей панели меню рабочего листа после

установки надстройки в меню Сервис ► Надстройки (Tools ► Add-ins). Если щелкнуть элемент меню Super Code, будет выполнен макрос (который находится в стандартном модуле надстройки). Помните, что предыдущий код необходимо поместить в частный модуль объекта ЭтаКнига (ThisWorkbook) для данной надстройки.

Если вы **хотите**, чтобы элемент меню Super Code был добавлен, скажем, перед элементом меню Формат (Format), используйте код из листинга 7.25.

Листинг 7.25

```
Option Explicit
Dim cControl As CommandBarButton

Private Sub Workbook_AddinInstall()

Dim iContIndex As Integer
    On Error Resume Next 'На всякий случай
        'Удалить любой существующий элемент меню, который мог остаться.
        Application.CommandBars("Worksheet Menu Bar").Controls("Super _
            Code").Delete
        'Передать индекс элемента меню "Format" (Формат) в переменную.
        'Использовать метод FindControl для поиска его значения Index.
        'Идентификационный номер используется на случай пользовательской _
        'настройки
        iContIndex = Application.CommandBars.FindControl(ID:=30006).Index
        'Добавить новый элемент меню и присвоить его переменной CommandBarButton
        'Для выбора места нового элемента использовать число, переданное _
        'нашей переменной типа Integer.
        Set cControl = Application.CommandBars("Worksheet Menu Bar") _
            .Controls.Add(Before:=iContIndex)
        'Работа с переменной
        With cControl
            .Caption = "Super Code"
            .Style = msoButtonCaption
            .OnAction = "MyGreatMacro" 'Макрос, хранимый в стандартном модуле
        End With
    On Error GoTo 0
End Sub
```

В этом случае вам не придется изменять код Workbook_AddinUninstall().

В этих примерах весь код элемента меню находится в Workbook_AddinInstall и Workbook_AddinUnInstall. Это не представляет **проблемы**, если код добавляет только один элемент меню. Если, однако, вы хотите добавить несколько элементов и, возможно, вложенное меню, необходимо поместить код элемента меню в **процедуру** (или две) внутри стандартного модуля. Затем используйте код из листинга 7.26.

Листинг 7.26

```
Private Sub Workbook_AddinInstall()
    Run "AddMenus"
End Sub
```

```
Private Sub Workbook_AddinUninstall()  
    Run "DeleteMenu"  
End Sub
```

В стандартный модуль поместите код из листинга 7.27.

Листинг 7.27

```
Sub AddMenus()  
    Dim cMenu1 As CommandBarControl  
    Dim cbMainMenuBar As CommandBar  
    Dim iHelpMenu As Integer  
    Dim cbcCutomMenu As CommandBarControl  
  
    '(1)Удалить существующий. На случай, если он не существует, _  
    необходимо использовать On Error Resume next.  
    On Error Resume Next  
    Application.CommandBars("Worksheet Menu Bar").Controls("&New _  
        Menu").Delete  
  
    '(2)Присвоить переменной CommandBar панель меню рабочего листа  
    Set cbMainMenuBar = _  
        Application.CommandBars("Worksheet Menu Bar")  
  
    '(3)Вернуть последовательный номер меню Help (Справка). Затем его _  
    можно использовать для выбора места для пользовательского элемента меню.  
    iHelpMenu = cbMainMenuBar.Controls("Help").Index  
  
    '(4)Добавить элемент управления к "Панели меню рабочего листа" перед Help.  
    'Присвоить его переменной CommandBarControl  
    Set cbcCutomMenu = _  
        cbMainMenuBar.Controls.Add(Type:=msoControlPopup, Before:=iHelpMenu)  
  
    '(5)Дать элементу управления заголовок  
    cbcCutomMenu.Caption = "&New Menu"  
  
    '(6)Работа с новым элементом управления, добавить элемент управления _  
    второго уровня, дать ему заголовок и сообщить, какой макрос выполнять _  
    (OnAction).  
    With cbcCutomMenu.Controls.Add(Type:=msoControlButton)  
        .Caption = "Menu 1"  
        .OnAction = "MyMacro1"  
    End With  
  
    '(6a)Добавить еще один элемент управления второго уровня. присвоить _  
    заголовок и сообщить, какой макрос выполнять (OnAction)  
    With cbcCutomMenu.Controls.Add(Type:=msoControlButton)  
        .Caption = "Menu 2"  
        .OnAction = "MyMacro2"  
    End With
```

продолжение ↗

Листинг 7.27 (продолжение)

```

'Повторить шаг "ба" для каждого элемента меню, который хотите добавить

'Добавить еще меню, которое будет открывать еще одно меню
'Присвоить его переменной CommandBarControl
Set cbcCutomMenu = cbcCutomMenu.Controls.Add(Type:=msoControlPopup)
'Дать элементу управления заголовок
cbcCutomMenu.Caption = "Ne&xt Menu"

'Добавить элемент управления к созданному только что вложенному меню
With cbcCutomMenu.Controls.Add(Type:=msoControlButton)
    .Caption = "&Charts"
    .FaceId = 420
    .OnAction = "MyMacro2"
End With

On Error GoTo 0
End Sub

Sub DeleteMenu()
    On Error Resume Next
        Application.CommandBars("Worksheet Menu Bar").Controls("&New _
Menu").Delete

    On Error GoTo 0
End Sub

```

При работе со свойством OnAction могут возникать проблемы, если в рабочей книге пользователя есть макрос с таким же именем, что и макрос в вашей надстройке. Чтобы обезопасить себя, можно применить способ, приведенный в листинге 7.28.

Листинг 7.28

```

With cbcCutomMenu.Controls.Add(Type:=msoControlButton)
    .Caption = "&Charts"
    .FaceId = 420
    .OnAction = ThisWorkbook.Name & "!MyMacro2"
End With

```

Так вы гарантируете, что Excel будет знать, какой из макросов следует запустить, когда пользователь щелкает кнопку. Благодаря этим фрагментам кода вы сможете легко распространять и использовать макросы с максимальной эффективностью.

Соединение Excel с остальным миром

Трюки с 95 по 100

С давних времен Excel имеет связи с другими членами семейства Microsoft Office, а также с базами данных. С ростом всемирной сети в Excel также появились возможности экспортирования файлов HTML, что значительно упростило публикацию информации, созданной в Excel. Однако в последние несколько лет Microsoft создала функции, позволяющие использовать Excel как интерфейс для информации из любых источников и отправлять информацию из Excel в другие формы, какие бы это ни были информация и формы. В то же время несколько разработчиков создали способы помещения и получения информации из электронных таблиц Excel без использования Excel, и они также стали важными выходами для отправки и получения данных.

Т Р Ю К
№ 95

Загрузка документа XML в Excel

Если кто-либо пришлет файл XML, содержащий данные в **таблицах**, вам не придется читать весь текст и все угловые скобки с тегами. Можно загрузить этот документ напрямую в Excel, сообщить Excel, как следует отобразить этот документ, и работать с данными при помощи карт.

В последние несколько лет язык XML (Extensible Markup Language, букв. «расширяемый язык разметки») стал распространенным форматом обмена информацией, и нет ничего необычного в том, что люди и организации отправляют друг другу файлы XML. Простые структуры, лежащие в основе XML, делают обмен информацией чрезвычайно простым, причем неважно, используют ли все стороны одно и то же программное обеспечение. Однако до недавних пор, хотя общие утилиты XML стали широко распространены, заполнить пробел между документами XML и пользовательским интерфейсом было еще достаточно трудно. Excel 2003 упрощает эту задачу, по крайней мере, для данных в сетке таблицы.

ПРИМЕЧАНИЕ

В этом трюке используются возможности Excel, доступные только в Excel 2003 для Windows. Более ранние версии Excel не поддерживают их; эти возможности не поддерживаются и в существующих, и планируемых версиях Excel для Macintosh.

Начнем с простого документа XML, приведенного в листинге 8.1.

Листинг 8.1. Простой документ XML для анализа в Excel

```
<?xml version="1.0" encoding="UTF-8"?>
<sales>

  <sale>
    <date>2003-10-05</date>
    <ISBN>0596005385</ISBN>
    <Title>Office 2003 XML Essentials</Title>
    <PriceUS>34.95</PriceUS>
    <quantity>200</quantity>
    <customer ID="1025">Zork's Books</customer>
  </sale>

  <sale>
    <date>2003-10-05</date>
    <ISBN>0596002920</ISBN>
    <Title>XML in a Nutshell, 2nd Edition</Title>
    <PriceUS>39.95</PriceUS>
    <quantity>90</quantity>
    <customer ID="1025">Zork's Books</customer>
  </sale>

  <sale>
    <date>2003-10-05</date>
    <ISBN>0596002378</ISBN>
    <Title>SAX2</Title>
    <PriceUS>29.95</PriceUS>
    <quantity>300</quantity>
    <customer ID="1025">Zork's Books</customer>
  </sale>

  <sale>
    <date>2003-10-05</date>
    <ISBN>0596005385</ISBN>
    <Title>Office 2003 XML Essentials</Title>
    <PriceUS>34.95</PriceUS>
    <quantity>10</quantity>
    <customer ID="1029">Books of Glory</customer>
  </sale>

  <sale>
    <date>2003-10-05</date>
    <ISBN>0596002920</ISBN>
    <Title>XML in a Nutshell, 2nd Edition</Title>
    <PriceUS>39.95</PriceUS>
    <quantity>25</quantity>
    <customer ID="1029">Books of Glory</customer>
  </sale>
```

```
<sale>
<date>2003-10-07</date>
<ISBN>0596002378</ISBN>
<Title>SAX2</Title>
<PriceUS>29.95</PriceUS>
<quantity>5</quantity>
<customer ID="1029">Books of Glory</customer>
</sale>
```

```
<sale>
<date>2003-10-18</date>
<ISBN>0596002378</ISBN>
<Title>SAX2</Title>
<PriceUS>29.95</PriceUS>
<quantity>15</quantity>
<customer ID="2561">Title Wave</customer>
</sale>
```

```
<sale>
<date>2003-10-21</date>
<ISBN>0596002920</ISBN>
<Title>XML in a Nutshell. 2nd Edition</Title>
<PriceUS>39.95</PriceUS>
<quantity>15</quantity>
<customer ID="9021">Books for You</customer>
</sale>
```

```
</sales>
```

Этот документ можно открыть непосредственно в Excel 2003 командой **Файл** ▶ **Открыть** (File ▶ Open). Откроется диалоговое окно (рис. 8.1).

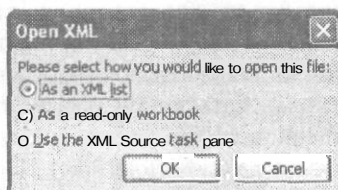


Рис. 8.1. Открытие файла XML в Excel 2003

Если вы выберете переключатель XML-список (As an XML list), то увидите предупреждение, что Excel создаст собственную схему для этого документа, не имеющего схемы (рис. 8.2). Щелкнув кнопку ОК, вы увидите, какой способ Excel выбрал для представления информации в открываемом документе в виде электронной таблицы (рис. 8.3).

Обратите внимание, что Excel ожидал встретить формат даты, который используется для элемента даты, поэтому теперь даты, импортированные как 2003-10-05, отображаются как 10/5/2003.

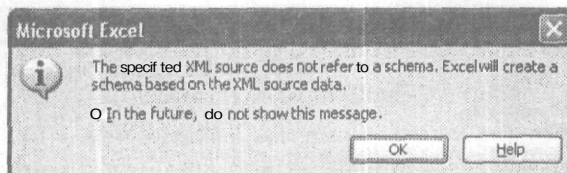


Рис. 8.2. Предупреждение Excel 2003 об отсутствии ссылок на схему

	A	B	C	D	E	F	G	H
1	date	ISBN	Title	PriceUS	quantity	customer	ID	
2	10/5/2003	596005385	Office 2003 XML Essentials	34.95	200	Zork's Books	1025	
3	10/5/2003	596002920	XML in a Nutshell, 2nd Edition	39.95	90	Zork's Books	1025	
4	10/5/2003	596002378	SAX2.....	29.95	300	Zork's Books	1025	
5	10/5/2003	596005385	Office 2003 XML Essentials	34.95	10	Books of Glory	1029	
6	10/5/2003	596002920	XML in a Nutshell, 2nd Edition	39.95	25	Books of Glory	1029	
7	10/7/2003	596002378	SAX2	29.95	5	Books of Glory	1029	
8	10/18/2003	596002378	SAX2.....	29.95	15	Title Wave	2551	
9	10/21/2003	596002920	XML in a Nutshell, 2nd Edition	39.95	15	Books for You	19021	
10	*							
11								

Рис. 8.3. Данные XML, представленные как XML-список в Excel 2003

Теперь, когда документ загружен в Excel, обрабатывать данные можно так же, как любые другие данные в Excel, — вставлять их в формулы, создавать именованные диапазоны, строить диаграммы на основе содержимого и т. д. Чтобы помочь вам, в Excel предусмотрено несколько встроенных возможностей анализа данных. Раскрывающиеся списки в заголовках столбцов позволяют выбирать способ сортировки данных (по умолчанию данные выводятся в том порядке, в котором они записаны в исходном документе). Можно также включить отображение строки итогов Итог (Total); для этого можно воспользоваться панелью инструментов Список (List) или щелкнуть правой кнопкой мыши в любом месте списка и в контекстном меню выбрать команду Список ▶ Строка итогов (List ▶ Total Row). Когда строка итогов появится, выбрать вид итоговой информации можно будет в раскрывающемся меню (рис. 8.4).

Данные можно обновить, добавив при этом в обновляемую область информацию из документа XML с такой же структурой. Если у вас есть еще один документ с такой структурой, можете щелкнуть список правой кнопкой мыши, в контекстном меню выбрать команду XML ▶ Импорт (XML ▶ Import) и выбрать второй документ. После добавления данных список может выглядеть, как на рис. 8.5.

Кроме того, после редактирования данные можно экспортировать обратно в файл XML, щелкнув список правой кнопкой мыши и выбрав в контекстном меню команду XML ▶ Экспорт (XML ▶ Export). Это превращает Excel в очень удобный инструмент редактирования простых XML-документов с табличной структурой.

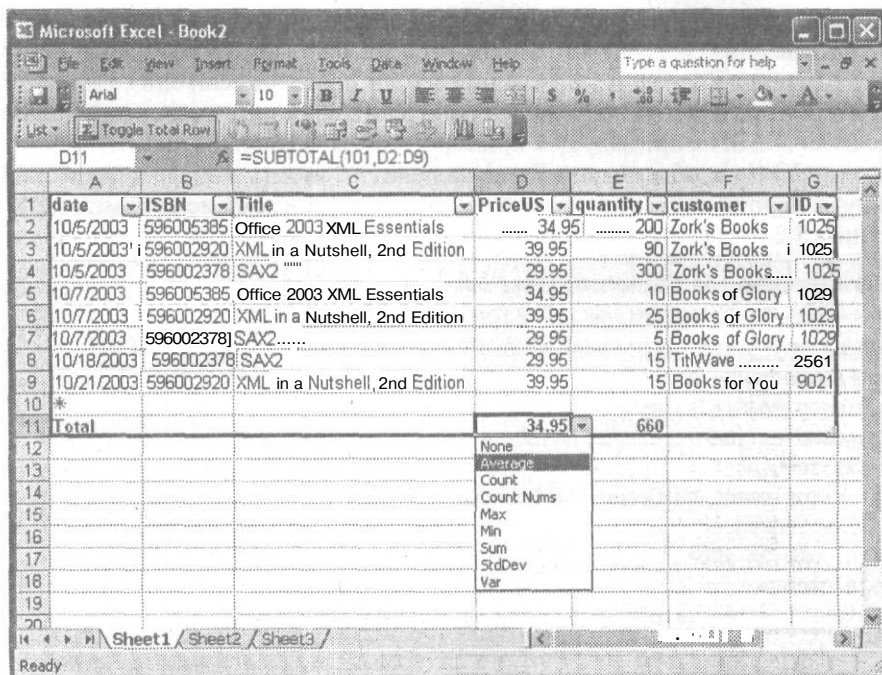


Рис. 8.4. Выбор итогов для списка XML в Excel 2003

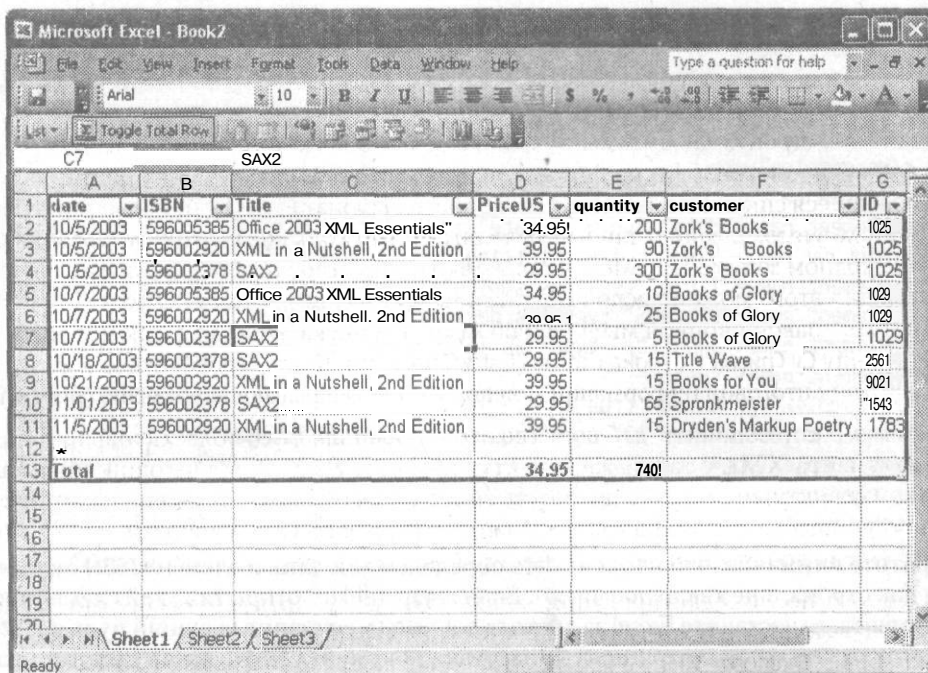


Рис. 8.5. Тот же XML-список с обновленными данными

Если данные достаточно простые, вы чаще всего можете доверить Excel выбор способа представления содержимого файла и использовать предусмотренные настройки по умолчанию. Если же данные усложняются, особенно если они содержат даты или текст, который выглядит как числа (обратите внимание на отсутствующие ведущие нули в ISBN в предыдущем примере!), то вы, возможно, захотите использовать схемы XML, чтобы указать Excel, как следует читать данные и какие данные подойдут к данной карте. Для нашего документа XML-схема может выглядеть, как в листинге 8.2.

Листинг 8.2. Схема для данных о продаже книг

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="sales">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="sale"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="sale">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="date"/>
        <xs:element ref="ISBN"/>
        <xs:element ref="Title"/>
        <xs:element ref="PriceUS"/>
        <xs:element ref="quantity"/>
        <xs:element ref="customer"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="date" type="xs:date"/>
  <xs:element name="ISBN" type="xs:string"/>
  <xs:element name="Title" type="xs:string"/>
  <xs:element name="PriceUS" type="xs:decimal"/>
  <xs:element name="quantity" type="xs:integer"/>
  <xs:element name="customer">
    <xs:complexType mixed="true">
      <xs:attribute name="ID" use="required" type="xs:integer"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Обратите внимание, что элемент date определен как дата, а элемент ISBN определен как строка, а не как целое число. Если вы начнете с открытия этой схемы, а не документа, то заставите Excel загружать документ, сохранив ведущий ноль в ISBN. На этот раз вы создадите список до загрузки документа XML, начав с пустого рабочего листа.

Вам понадобится открыть область задач Источник XML (XML Source). Если она еще не открыта, выберите команду Вид ► Область задач (View ► Task Pane) или нажмите сочетание клавиш **Ctrl+F1**. Затем в раскрывающемся списке сверху области задач выберите Источник XML (XML Source) и вы увидите что-то похожее на рис. 8.6.

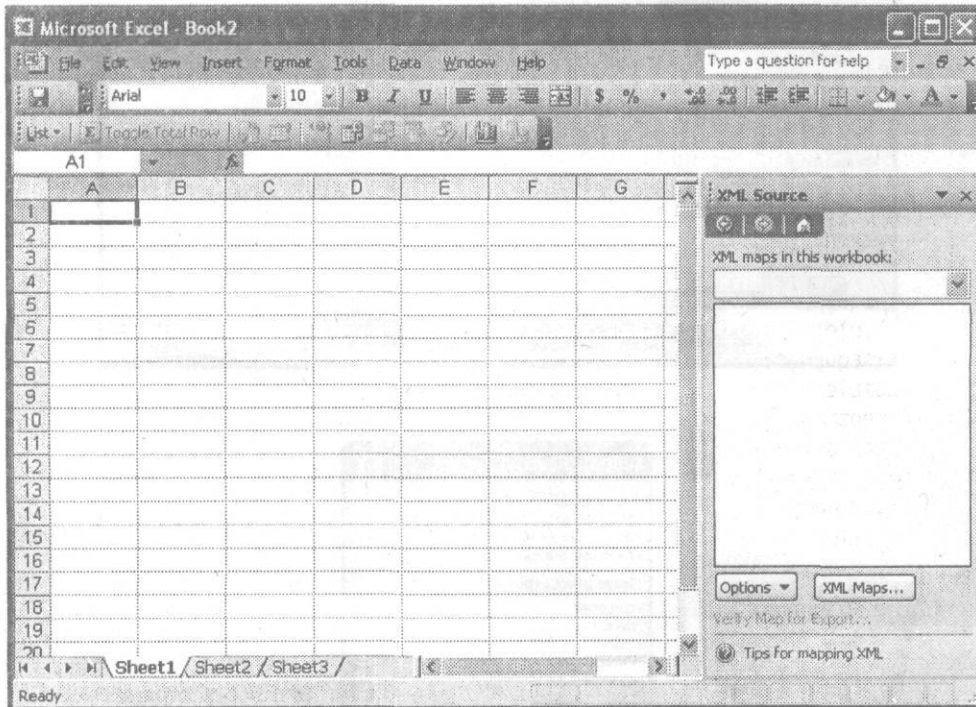


Рис. 8.6. Область задач Источник XML

Чтобы загрузить схему, щелкните кнопку Карты XML (XML Maps). Откроется диалоговое окно Карты XML (XML Maps) (рис. 8.7).

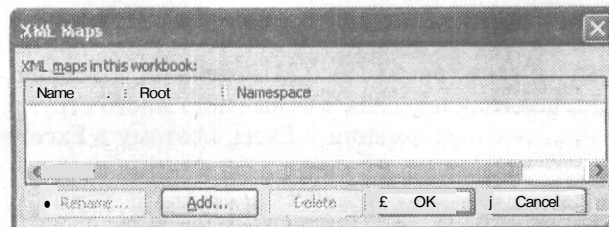


Рис. 8.7. Диалоговое окно Карты XML

Щелкните кнопку Добавить (Add), чтобы открыть схему, и выберите схему (рис. 8.8). Если схема не ограничивает документы одним начальным элементом, появится диалоговое окно с просьбой выбрать корневой элемент (рис. 8.9). Так как документы в этом примере начинаются с элемента sales, выберите «sales».

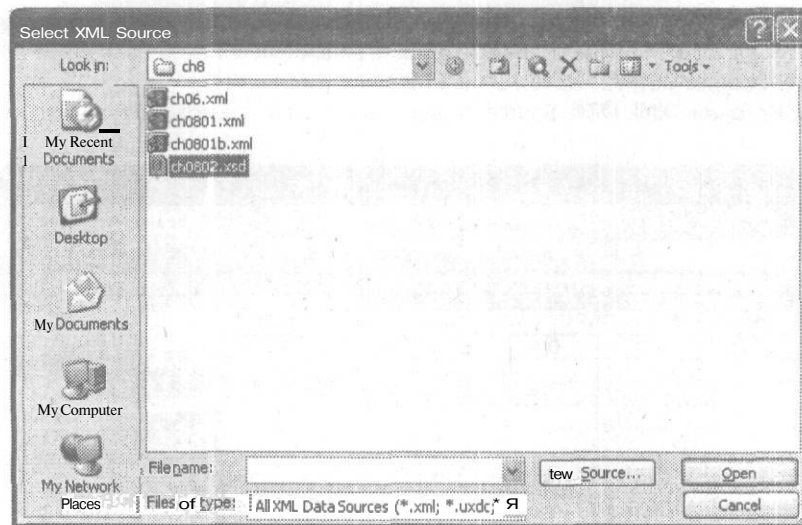


Рис. 8.8. Выбор XML-схемы

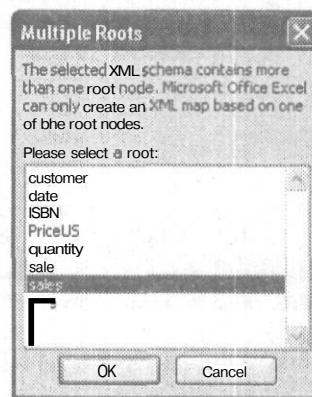


Рис. 8.9. Выбор корневого элемента для схемы

Когда вы щелкнете кнопку ОК, появится предупреждение о возможных сложностях с интерпретацией схем (рис. 8.10). XML-схема (XML Schema) — это огромная спецификация, поддерживающая чрезвычайно много структур, не отвечающих способу восприятия информации в Excel, поэтому в Excel есть некоторые ограничения.

В диалоговом окне Карты XML (XML Maps) (рис. 8.11) Excel сообщит, что схема была добавлена к электронной таблице.

Если вы щелкнете кнопку ОК, то вернетесь в главное окно Excel и в области задач Источник XML (XML Source) появится диаграмма, отображающая структуру схемы (рис. 8.12). Теперь, когда у вас есть структура, можно создать список. Самый простой способ сделать это, особенно с небольшими документами, как наш, — перетащить значок sales на ячейку A1 (рис. 8.12).

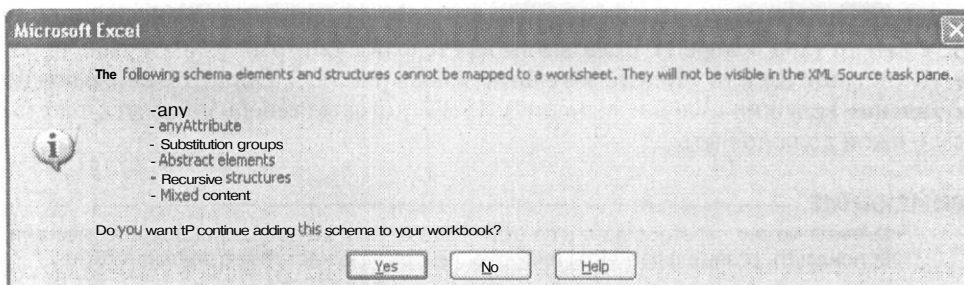


Рис. 8.10. Предупреждение об обработке схемы

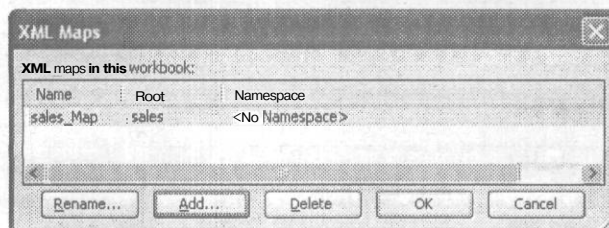


Рис. 8.11. Схема, загруженная как карта XML

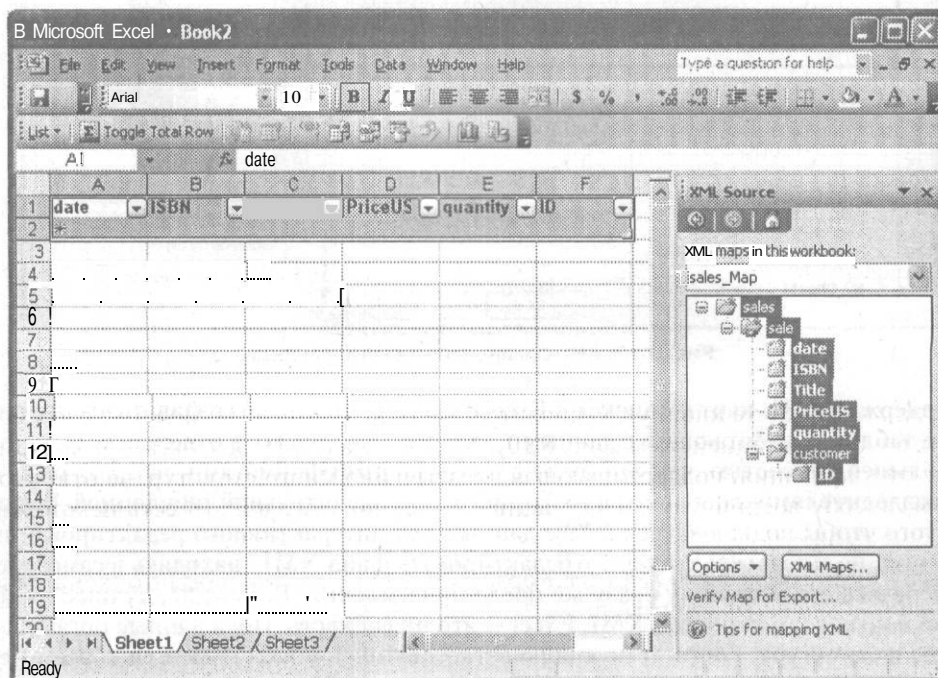


Рис. 8.12. XML-список, созданный на основе схемы в области задач XML

Теперь, обустроив дом для данных, нужно заселить его. Можно щелкнуть кнопку Импорт XML-данных (Import XML Data) на панели инструментов Список (List) или

щелкнуть правой кнопкой мыши список и в контекстном меню выбрать кнопку XML ► Импорт (XML ► Import). Если вы выберете файл, который уже открывали ранее (в листинге 8.1), то увидите результат, как на рис. 8.13. Обратите внимание на добавление ведущих нулей к значениям ISBN, которые теперь являются текстовыми, как и должно быть.

ПРИМЕЧАНИЕ

Элементы можно перетаскивать и по отдельности, если вы хотите поменять их местами, или помещать разные фрагменты информации в разные места электронной таблицы.

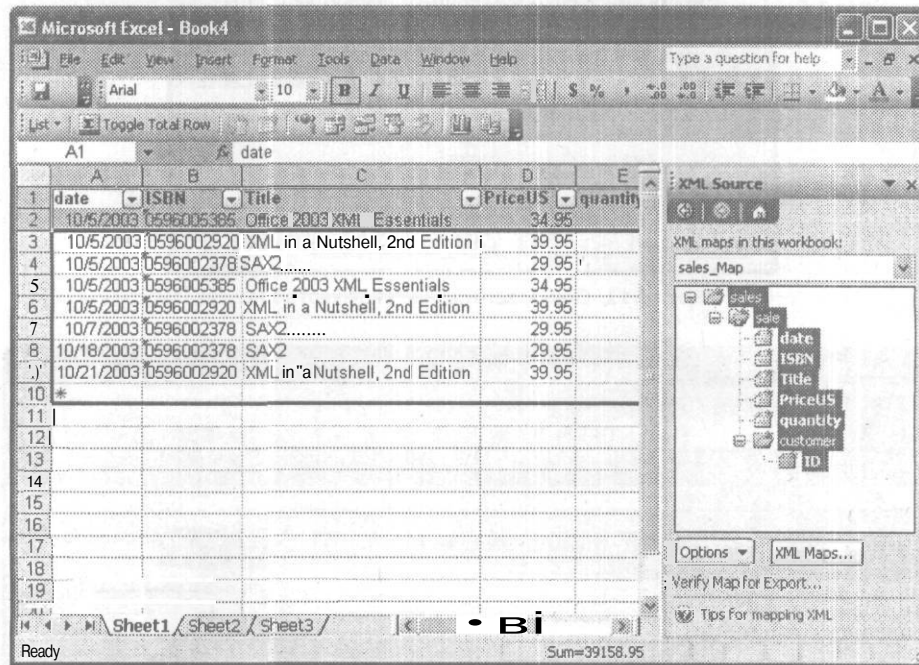


Рис. 8.13. XML-список, заполненный данными

Поддержка XML-карт и списков в Excel означает, что можно создавать электронные таблицы, работающие с данными, которые поступают в отдельных файлах, с большей гибкостью, чем это было с предыдущими форматами, например CSV (с разделителями-запятыми) или форматом с разделительной табуляцией. Вместо того чтобы подключаться к базе данных для интерактивного редактирования данных, пользователь сможет отредактировать файл XML, находясь в самолете, и передать его заказчику сразу же после приземления. Возможно, лучшее свойство новых возможностей XML Excel — это их гибкость. Пока данные организованы в структуру, соответствующую сетке таблицы, у Excel имеется совсем немного правил относительно того, какие виды XML можно передавать туда. Несколько раз щелкнув мышью и совершенно ничего не программируя, можно интегрировать данные XML в электронные таблицы.

Саймон Сен-Лорен

Т Р Ю К
№ 96

Сохранение в виде SpreadsheetML и выделение данных

Начиная с Excel XP, в Excel появилась возможность экспорта XML. SpreadsheetML обеспечивает представление электронных таблиц в формате XML, при этом сохраняя все форматирование и информацию формул.

Хотя есть несколько способов прочитать файлы электронных таблиц Excel без использования Excel (подробнее об этом — в разделе «Трюк № 100. Создание электронных таблиц Excel в других средах разработки»), один из самых простых — экспортировать XML-файлы с использованием словаря Microsoft SpreadsheetML. SpreadsheetML пока не завершен: очевидно, отсутствуют диаграммы и код VBA, но он представляет основные компоненты электронной таблицы, включая формулы, именованные диапазоны и форматирование.

ПРИМЕЧАНИЕ

В этом трюке используются возможности Excel, доступные только в Excel 2002 и Excel 2003 для Windows. В предыдущих версиях Excel они не поддерживаются; эти возможности не поддерживаются и в существующих, и планируемых версиях Excel для Macintosh.

Самый простой способ познакомиться с SpreadsheetML — сохранить электронную таблицу в виде XML. Такая электронная таблица (рис. 8.14) включает данные, формулы, именованные диапазоны и ячейки, а также простое форматирование.

ID Number	Critter	Price	Quantity	Total
4627	Diplodocus	\$22.50	127	\$2,857.50
3912	Brontosaurus	\$17.50	74	\$1,295.00
9845	Triceratops	\$12.00	91	\$1,092.00
9625	Vulcanodon	\$19.00	108	\$2,052.00
5903	Stegosaurus	\$18.50	63	\$1,165.50
1824	Monoclonius	\$16.50	133	\$2,194.50
9728	Megalosaurus	\$23.00	128	\$2,944.00
8649	Barosaurus	\$17.00	91	\$1,547.00
Total:				\$15,147.50

Рис. 8.14. Тестовая электронная таблица для SpreadsheetML

Сохранив электронную таблицу в формате XML Spreadsheet (*.xml), который можно выбрать в диалоговом окне сохранения файла — **Файл** ▶ **Сохранить как** (File ▶ Save As), — вы получите длинный документ XML с разметкой (листинг 8.3). Ключевые слова и фрагменты выделены жирным шрифтом.

Листинг 8.3. Документ SpreadsheetML

```

<?xml version="1.0"?>
<?mso-application progid="Excel.Sheet"?>
<Workbook xmlns="urn:schemas-microsoft-com:office:spreadsheet"
  xmlns:o="urn:schemas-microsoft-com:office:office"
  xmlns:x="urn:schemas-microsoft-com:office:excel"
  xmlns:ss="urn:schemas-microsoft-com:office:spreadsheet"
  xmlns:html="http://www.w3.org/TR/REC-html40">
<DocumentProperties xmlns="urn:schemas-microsoft-com:office:office">
  <Author>Simon St.Laurent</Author>
  <LastAuthor>Simon St.Laurent</LastAuthor>
  <Created>2003-12-03T15:48:38Z</Created>
  <LastSaved>2004-01-26T21:04:14Z</LastSaved>
  <Company>O'Reilly & Associates</Company>
  <Version>11.5703</Version>
</DocumentProperties>
<ExcelWorkbook xmlns="urn:schemas-microsoft-com:office:excel">
  <WindowHeight>6150</WindowHeight>
  <WindowWidth>8475</WindowWidth>
  <WindowTopX>120</WindowTopX>
  <WindowTopY>30</WindowTopY>
  <ProtectStructure>False</ProtectStructure>
  <ProtectWindows>False</ProtectWindows>
</ExcelWorkbook>
<Styles>
  <Style ss:ID="Default" ss:Name="Normal">
    <Alignment ss:Vertical="Bottom"/>
    <Borders/>
    <Font/>
    <Interior/>
    <NumberFormat/>
    <Protection/>
  </Style>
  <Style ss:ID="s21">
    <NumberFormat ss:Format="mmm\ -yy"/>
  </Style>
  <Style ss:ID="s22">
    <NumberFormat ss:Format="&quot;#&quot;#.#0.00"/>
  </Style>
  <Style ss:ID="s23">
    <Font x:Family="Swiss" ss:Bold="1"/>
  </Style>
</Styles>
<Names>
  <NamedRange ss:Name="Critters" ss:RefersTo="=Sheet1!R4C2:R11C2"/>
  <NamedRange ss:Name="Date" ss:RefersTo="=Sheet1!R1C2"/>
  <NamedRange ss:Name="ID" ss:RefersTo="=Sheet1!R4C1:R11C1"/>
  <NamedRange ss:Name="Price" ss:RefersTo="=Sheet1!R4C3:R11C3"/>
  <NamedRange ss:Name="Quantity" ss:RefersTo="=Sheet1!R4C4:R11C4"/>
  <NamedRange ss:Name="Total" ss:RefersTo="=Sheet1!R12C5"/>
</Names>

```

```

<Worksheet ss:Name="Sheet1">
  <Table ss:ExpandedColumnCount="5" ss:ExpandedRowCount="12" x:FullColumns="1"
    x:FullRows="1">
    <Column ss:AutoFitWidth="0" ss:Width="73.5"/>
    <Column ss:AutoFitWidth="0" ss:Width="96.75"/>
    <Column ss:Index="5" ss:AutoFitWidth="0" ss:Width="56.25"/>
    <Row>
      <Cell ss:StyleID="s23"><Data ss:Type="String">Sales for:</Data></Cell>
      <Cell ss:StyleID="s21"><Data ss:Type="DateTime">2004-01-01T00:00:00.000</
Data><NamedCell
        ss:Name="Date"/></Cell>
    </Row>
    <Row ss:Index="3" ss:StyleID="s23">
      <Cell><Data ss:Type="String">ID Number</Data></Cell>
      <Cell><Data ss:Type="String">Critter</Data></Cell>
      <Cell><Data ss:Type="String">Price</Data></Cell>
      <Cell><Data ss:Type="String">Quantity</Data></Cell>
      <Cell><Data ss:Type="String">Total</Data></Cell>
    </Row>
    <Row>
      <Cell><Data ss:Type="Number">4627</Data><NamedCell ss:Name="ID"/></Cell>
      <Cell><Data ss:Type="String">Diplodocus</Data><NamedCell ss:Name="Critters"/
></Cell>
      <Cell ss:StyleID="s22"><Data ss:Type="Number">22.5</Data><NamedCell
        ss:Name="Price"/></Cell>
      <Cell><Data ss:Type="Number">127</Data><NamedCell ss:Name="Quantity"/></
Cell>
      <Cell ss:StyleID="s22" ss:Formula="=RC[-2]*RC[-1]"><Data
        ss:Type="Number">2857.5</Data></Cell>
    </Row>
    <Row>
      <Cell><Data ss:Type="Number">3912</Data><NamedCell ss:Name="ID"/></Cell>
      <Cell><Data ss:Type="String">Brontosaurus</Data><NamedCell
        ss:Name="Critters"/></Cell>
      <Cell ss:StyleID="s22"><Data ss:Type="Number">17.5</Data><NamedCell
        ss:Name="Price"/></Cell>
      <Cell><Data ss:Type="Number">74</Data><NamedCell ss:Name="Quantity"/></Cell>
      <Cell ss:StyleID="s22" ss:Formula="=RC[-2]*RC[-1]"><Data
        ss:Type="Number">1295</Data></Cell>
    </Row>
    <Row>
      <Cell><Data ss:Type="Number">9845</Data><NamedCell ss:Name="ID"/></Cell>
      <Cell><Data ss:Type="String">Triceratops</Data><NamedCell
        ss:Name="Critters"/></Cell>
      <Cell ss:StyleID="s22"><Data ss:Type="Number">12</Data><NamedCell
        ss:Name="Price"/></Cell>
      <Cell><Data ss:Type="Number">91</Data><NamedCell ss:Name="Quantity"/></Cell>
      <Cell ss:StyleID="s22" ss:Formula="=RC[-2]*RC[-1]"><Data
        ss:Type="Number">1092</Data></Cell>
    </Row>
  </Table>

```

продолжение ↗

Листинг 8.3 (продолжение)

```

<Row>
  <Cell><Data ss:Type="Number">9625</Data><NamedCell ss:Name="ID"/></Cell>
  <Cell><Data ss:Type="String">Vulcanodon</Data><NamedCell ss:Name="Critters"/></Cell>
  <Cell ss:StyleID="s22"><Data ss:Type="Number">19</Data><NamedCell ss:Name="Price"/></Cell>
  <Cell><Data ss:Type="Number">108</Data><NamedCell ss:Name="Quantity"/></Cell>
</Row>
<Row>
  <Cell ss:StyleID="s22" ss:Formula="=RC[-2]*RC[-1]"><Data ss:Type="Number">2052</Data></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="Number">5903</Data><NamedCell ss:Name="ID"/></Cell>
  <Cell><Data ss:Type="String">Stegosaurus</Data><NamedCell ss:Name="Critters"/></Cell>
  <Cell ss:StyleID="s22"><Data ss:Type="Number">18.5</Data><NamedCell ss:Name="Price"/></Cell>
  <Cell><Data ss:Type="Number">63</Data><NamedCell ss:Name="Quantity"/></Cell>
  <Cell ss:StyleID="s22" ss:Formula="=RC[-2]*RC[-1]"><Data ss:Type="Number">1165.5</Data></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="Number">1824</Data><NamedCell ss:Name="ID"/></Cell>
  <Cell><Data ss:Type="String">Monoclonius</Data><NamedCell ss:Name="Critters"/></Cell>
  <Cell ss:StyleID="s22"><Data ss:Type="Number">16.5</Data><NamedCell ss:Name="Price"/></Cell>
  <Cell><Data ss:Type="Number">133</Data><NamedCell ss:Name="Quantity"/></Cell>
  <Cell ss:StyleID="s22" ss:Formula="=RC[-2]*RC[-1]"><Data ss:Type="Number">2194.5</Data></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="Number">9728</Data><NamedCell ss:Name="ID"/></Cell>
  <Cell><Data ss:Type="String">Megalosaurus</Data><NamedCell ss:Name="Critters"/></Cell>
  <Cell ss:StyleID="s22"><Data ss:Type="Number">23</Data><NamedCell ss:Name="Price"/></Cell>
  <Cell><Data ss:Type="Number">128</Data><NamedCell ss:Name="Quantity"/></Cell>
  <Cell ss:StyleID="s22" ss:Formula="=RC[-2]*RC[-1]"><Data ss:Type="Number">2944</Data></Cell>
</Row>
<Row>
  <Cell><Data ss:Type="Number">8649</Data><NamedCell ss:Name="ID"/></Cell>
  <Cell><Data ss:Type="String">Barosaurus</Data><NamedCell ss:Name="Critters"/></Cell>
  <Cell ss:StyleID="s22"><Data ss:Type="Number">17</Data><NamedCell ss:Name="Price"/></Cell>

```

```

    <Cell><Data ss:Type="Number">91</Data><NamedCell ss:Name="Quantity"/></Cell>
    <Cell ss:StyleID="s22" ss:Formula="=RC[-2]*RC[-1]"><Data
ss:Type="Number">1547</Data></Cell>
  </Row>
  <Row>
    <Cell ss:Index="4" ss:StyleID="s23"><Data ss:Type="String">Total :</Data></
Cell>
    <Cell ss:StyleID="s22" ss:Formula="=SUM(R[-8]C:R[-1]C)"><Data
ss:Type="Number">15147.5</Data><NamedCell
    ss:Name="Total"/></Cell>
  </Row>
</Table>
<WorksheetOptions xmlns="urn:schemas-microsoft-com:office:excel">
  <Print>
    <ValidPrinterInfo/>
    <HorizontalResolution>600</HorizontalResolution>
    <VerticalResolution>600</VerticalResolution>
  </Print>
  <Selected/>
  <Panes>
    <Pane>
      <Number>3</Number>
      <ActiveRow>11</ActiveRow>
      <ActiveCol>4</ActiveCol>
    </Pane>
  </Panes>
  <ProtectObjects>False</ProtectObjects>
  <ProtectScenarios>False</ProtectScenarios>
</WorksheetOptions>
</Worksheet>
<Worksheet ss:Name="Sheet2">
  <WorksheetOptions xmlns="urn:schemas-microsoft-com:office:excel">
    <ProtectObjects>False</ProtectObjects>
    <ProtectScenarios>False</ProtectScenarios>
  </WorksheetOptions>
</Worksheet>
<Worksheet ss:Name="Sheet3">
  <WorksheetOptions xmlns="urn:schemas-microsoft-com:office:excel">
    <ProtectObjects>False</ProtectObjects>
    <ProtectScenarios>False</ProtectScenarios>
  </WorksheetOptions>
</Worksheet>
</Workbook>

```

Первая выделенная строка, `<?mso-application progid="Excel.Sheet">`, — это оператор обработки XML, который сообщает Windows (в действительности, компоненту, который Office 2003 добавляет в Windows), что этот документ XML в действительности является электронной таблицей Excel. При отображении этого файла в Windows на нем появляется значок Excel, дважды щелкнув который, можно запустить Excel.

Корневой элемент документа, `Worksheet`, находится сразу же после оператора обработки. Его атрибуты определяют пространства имен, которые используются для различных частей `SpreadsheetML`. Следующие несколько строк описывают в основном метаданные, способ отображения окна и информацию о форматировании, и далее ничто не стоит пристального внимания, пока вы не попадаете к элементам `Names` и `Worksheet`.

Элемент `Names` определяет именованные диапазоны и ячейки в документе. Следующие два элемента `NamedRange` описывают именованный диапазон `Quantity`, который начинается в строке 4, столбце 4 и заканчивается в строке 11, столбце 4, и именованный диапазон `Total`, который представляет собой всего лишь ячейку в строке 12, столбце 5:

```
<NamedRange ss:Name="Quantity" ss:RefersTo="=Sheet1!R4C4:R11C4"/>
<NamedRange ss:Name="Total" ss:RefersTo="=Sheet1!R12C5"/>
```

Соль электронной таблицы заключается в элементе `Worksheet`. Он начинается с определения размера фактической таблицы данных

```
<Worksheet ss:Name="Sheet1">
  <Table ss:ExpandedColumnCount="5" ss:ExpandedRowCount="12" x:FullColumns="1"
  x:FullRows="1">
```

На этом листе с именем `Sheet1` используются 5 столбцов и 12 строк. (Атрибуты `x:FullColumns` и `x:FullRows` находятся в другом пространстве имен, который не будет использоваться в Excel при компоновке листа.) Фактическая информация в этой таблице хранится в элементах `Row` и `Cell`:

```
<Row>
  <Cell ss:StyleID="s23"><Data ss:Type="String">Sales for:</Data></Cell>
  <Cell ss:StyleID="s21"><Data ss:Type="DateTime">2004-01-01T00:00:00.000</
Data><NamedCell
  ss:Name="Date"/></Cell>
</Row>
<Row ss:Index="3" ss:StyleID="s23">
  <Cell><Data ss:Type="String">ID Number</Data></Cell>
```

Этот элемент `Row`, первый в электронной таблице, содержит два элемента `Cell` **Первый**, отформатированный как `s23` (то есть на электронной таблице он выделен жирным) и принадлежащий типу данных `String`, содержит текст «Sales for:». Вторая ячейка отформатирована как `s21` (обычный текст) и принадлежит типу данных `DateTime`. Ее содержимое описано в многословном формате ISO 8601. Эта ячейка также является частью именованного диапазона, в данном случае `Date`.

Большинство остальных элементов `Row` описаны по тому же шаблону, но есть несколько элементов, заслуживающих **дополнительного** внимания. Во втором элементе `Row` есть дополнительный атрибут — `ss:Index`:

```
<Row ss:Index="3" ss:StyleID="s23">
```

Excel не представляет пустые строки или пустые столбцы пустыми элементами `Row` или `Cell`. Он просто добавляет атрибут `ss:Index` к следующему элементу `Row` или `Cell`, хранящему данные, чтобы сообщить, где вы находитесь. Это требует, чтобы программы, обрабатывающие этот XML-код, были немного более внимательные при сборке таблицы. Далее следует обратить внимание на формулы:

```
<Cell ss:StyleID="s22" ss:Formula="=SUM(R[-8]C:R[-1]C)"><Data
ss:Type="Number">15147.5</Data><NamedCell
ss:Name="Total" /></Cell>
```

На рис. 8.15 этой ячейке присвоено имя Total, значение \$15,147.50 и формула =SUM(E4:E11). Все эти компоненты представлены здесь. Но вы должны собрать их, зная стиль s22 (определенного ранее в этом документе как денежный числовой формат), значение 15147.5 и формулу, которая при помощи относительных ссылок говорит: «сумма значений в том же столбце, где нахожусь я, начиная с ячейки, которая находится на 8 строк выше, и заканчивая ячейкой выше на 1 строку».

Может показаться, что обработать такой код совсем непросто, но на самом деле это не так уж сложно, если у вас есть утилиты для обработки XML. Для выделения информации можно использовать C#, Java, Perl, Python, VB или ваш любимый язык программирования, поддерживающий XML, но мы для демонстрации будем использовать XSLT.

ПРИМЕЧАНИЕ

Существует множество обработчиков XSLT, начиная от утилит командной строки и заканчивая приложениями Windows. Вы можете попробовать Archtags XRay, которую можно загрузить с сайта <http://architag.com/xray/>, или SAXON Михаэля Кея (Michael Kay), которую можно загрузить с сайта <http://saxon.sourceforge.net/>. Microsoft предлагает различные утилиты для обработки XSLT, включая утилиту командной строки, которую можно загрузить с сайта <http://msdn.microsoft.com/library/default.asp?url=/downloads/list/xmlgeneral.asp>. Если это выглядит достаточно увлекательно, чтобы вы решили изучить XSLT, рекомендуем приобрести книгу Майка Фицджеральда (Mike Fitzgerald) «Learning XSLT» (издательство O'Reilly).

Таблица стилей в листинге 8.4, если применить ее к XML в листинге 8.3, позволит создать намного более простой XML-код в листинге 8.5.

Листинг 8.4. Таблица стилей XSLT для выделения содержимого из SpreadsheetML в листинге 8.3

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://simonstl.com/ns/dinosaurs/"
  xmlns:ss="urn:schemas-microsoft-com:office:spreadsheet"
  >

  <xsl:output method="xml" omit-xml-declaration="yes" indent="yes" encoding="US-
  ASCII"/>

  <xsl:template match="/">
    <xsl:apply-templates select="ss:Workbook"/>
  </xsl:template>

  <xsl:template match="ss:Workbook">
    <dinosaurs>
      <xsl:apply-templates select="ss:Worksheet[@ss:Name = 'Sheet1']"/>
    </dinosaurs>
  </xsl:template>
```

продолжение ↗

Листинг 8.4 (продолжение)

```

<xsl:template match="ss:Worksheet">
  <date><xsl:value-of select="ss:Table/ss:Row/ss:Cell[@ss:StyleID = 's21']" /
</date>
  <xsl:apply-templates select="ss:Table" />
</xsl:template>

<xsl:template match="ss:Table">
  <xsl:apply-templates select="ss:Row[position( ) > 2]" />
  <!--Обратите внимание, так как Excel пропускает пустую строку, третья строка
находится в позиции 2-->
</xsl:template>

<xsl:template match="ss:Row[ss:Cell[4]]">
  <sale>
    <IDnum><xsl:apply-templates select="ss:Cell[1]" /></IDnum>
    <critter><xsl:apply-templates select="ss:Cell[2]" /></critter>
    <price><xsl:apply-templates select="ss:Cell[3]" /></price>
    <quantity><xsl:apply-templates select="ss:Cell[4]" /></quantity>
    <total><xsl:apply-templates select="ss:Cell[5]" /></total>
  </sale>
</xsl:template>

<xsl:template match="ss:Row">
  <total><xsl:apply-templates select="ss:Cell[2]" /></total>
</xsl:template>

</xsl:stylesheet>

```

Основная часть таблицы стилей — это шаблон, описывающий все строки, с четырьмя или более дочерними элементами ячеек. Он выделяет информацию из ячеек и помещает ее в элементы XML, которые отражают данные, позволяя получить результат, показанный в листинге 8.5.

Листинг 8.5. Информация, выделенная из SpreadsheetML в пользовательский словарь XML

```

<dinosaurs xmlns="http://simonstl.com/ns/dinosaurs/" xmlns:ss="urn:schemas-
microsoft-com:off-ice:spreadsheet">
  <date>2004-01-01T00:00:00.000</date>
  <sale>
    <IDnum>4627</IDnum>
    <critter>Diplodocus</critter>
    <price>22.5</price>
    <quantity>127</quantity>
    <total>2857.5</total>
  </sale>
  <sale>
    <IDnum>3912</IDnum>
    <critter>Brontosaurus</critter>
    <price>17.5</price>
    <quantity>74</quantity>
    <total>1295</total>

```

```
</sale>
<sale>
<IDnum>9845</IDnum>
<critter>Triceratops</critter>
<price>12</price>
<quantity>91</quantity>
<total>1092</total>
</sale>
<sale>
<IDnum>9625</IDnum>
<critter>Vulcanodon</critter>
<price>19</price>
<quantity>108</quantity>
<total>2052</total>
</sale>
<sale>
<IDnum>5903</IDnum>
<critter>Stegosaurus</critter>
<price>18.5</price>
<quantity>63</quantity>
<total>1165.5</total>
</sale>
<sale>
<IDnum>1824</IDnum>
<critter>Monoclonius</critter>
<price>16.5</price>
<quantity>133</quantity>
<total>2194.5</total>
</sale>
<sale>
<IDnum>9728</IDnum>
<critter>Megalosaurus</critter>
<price>23</price>
<quantity>128</quantity>
<total>2944</total>
</sale>
<sale>
<IDnum>8649</IDnum>
<critter>Barosaurus</critter>
<price>17</price>
<quantity>91</quantity>
<total>1547</total>
</sale>
<total>15147.5</total>
</dinosaurs>
```

Это те же данные, но в совсем другой форме. В данном случае информация формул была отброшена, но, так как Excel поддерживает значения наряду с формулами, этому приложению не требуется понимать формулы.

**Т Р Ю К
№ 97**

Создание электронных таблиц при помощи SpreadsheetML

Хотя экспорт электронных таблиц в формате XML — полезная возможность, Excel также позволяет импортировать информацию подобным образом и создавать электронные таблицы с использованием SpreadsheetML.

Excel поддерживает SpreadsheetML как для импорта, так и для экспорта информации, обеспечивая передачу информации в обоих направлениях. Можно открыть документ SpreadsheetML, внести какие-то изменения и заново открыть его в Excel. (Например, это единственный способ редактировать схемы для карт XML.) Возможно, еще важнее то, что можно генерировать документы SpreadsheetML из любых данных, обеспечивая относительно простой и автоматизируемый путь для преобразования необработанной информации в электронные таблицы Excel.

Чтобы продемонстрировать это, код в примере 8.6 преобразует XML листинга 8.5 предыдущего трюка обратно в Excel, используя XSLT. Таблица стилей в листинге 8.6 использует исходную электронную таблицу как шаблон, и создает XML, очень похожий на SpreadsheetML, первоначально сохраненный в Excel. В этом примере пропущено некоторое форматирование, чтобы создать видимые различия.

ПРИМЕЧАНИЕ

В этом трюке используются возможности Excel, доступные только в Excel 2002 и Excel 2003 для Windows. В предыдущих версиях Excel они не поддерживаются; эта возможности не поддерживаются и в текущих, и планируемых версиях Excel для Macintosh.

Листинг 8.6. XSLT для преобразования пользовательского словаря XML обратно в SpreadsheetML

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:d="http://simonstl.com/ns/dinosaurs/"
  xmlns:ss="urn:schemas-microsoft-com:office:spreadsheet"
  xmlns="urn:schemas-microsoft-com:office:spreadsheet"
>

<xsl:output method="xml" omit-xml-declaration="no" indent="yes" encoding="US-
ASCII"/>

<xsl:template match="/">
  <xsl:apply-templates select="d:dinosaurs" />
</xsl:template>

<xsl:template match="d:dinosaurs">

<xsl:processing-instruction name="mso-application">progid="Excel.Sheet"</
xsl:processing-instruction>
<Workbook xmlns="urn:schemas-microsoft-com:office:spreadsheet"
  xmlns:o="urn:schemas-microsoft-com:office:office"
```

```

xmlns:x="urn:schemas-microsoft-com:office:excel"
xmlns:ss="urn:schemas-microsoft-com:office:spreadsheet"
xmlns:html="http://www.w3.org/TR/REC-html40">
<DocumentProperties xmlns="urn:schemas-microsoft-com:office:office">
  <Author>Simon St.Laurent</Author>
  <LastAuthor>Simon St.Laurent</LastAuthor>
  <Created>2003-12-03T15:48:38Z</Created>
  <LastSaved>2003-12-03T15:57:46Z</LastSaved>
  <Company>O'Reilly & Associates</Company>
  <Version>11.5606</Version>
</DocumentProperties>
<ExcelWorkbook xmlns="urn:schemas-microsoft-com:office:excel">
  <WindowHeight>6150</WindowHeight>
  <WindowWidth>8475</WindowWidth>
  <WindowTopX>120</WindowTopX>
  <WindowTopY>30</WindowTopY>
  <ProtectStructure>False</ProtectStructure>
  <ProtectWindows>False</ProtectWindows>
</ExcelWorkbook>
<Styles>
  <Style ss:ID="Default" ss:Name="Normal">
    <Alignment ss:Vertical="Bottom"/>
    <Borders/>
    <Font/>
    <Interior/>
    <NumberFormat/>
    <Protection/>
  </Style>
  <Style ss:ID="s21">
    <NumberFormat ss:Format="mmm\ -yy"/>
  </Style>
  <Style ss:ID="s22">
    <NumberFormat ss:Format="&quot;:&quot;#,#0.00"/>
  </Style>
</Styles>
<Worksheet ss:Name="Sheet1">
  <Table ss:ExpandedColumnCount="5" ss:ExpandedRowCount="{count(d:sale)+4}"
x:FullColumns="1"
x:FullRows="1">
    <Column ss:AutoFitWidth="0" ss:Width="73.5"/>
    <Column ss:AutoFitWidth="0" ss:Width="96.75"/>
    <Column ss:Index="5" ss:AutoFitWidth="0" ss:Width="56.25"/>
    <Row>
      <Cell><Data ss:Type="String">Sales for:</Data></Cell>
      <Cell ss:StyleID="s21"><Data ss:Type="DateTime"><xsl:value-of
select="d:date"/></Data></Cell>
    </Row>
    <Row ss:Index="3">
      <Cell><Data ss:Type="String">ID Number</Data></Cell>
      <Cell><Data ss:Type="String">Criter</Data></Cell>

```

продолжение ↗

Листинг 8.6 (продолжение)

```

    <Cell><Data ss:Type="String">Price</Data></Cell>
    <Cell><Data ss:Type="String">Quantity</Data></Cell>
    <Cell><Data ss:Type="String">Total</Data></Cell>
  </Row>

  <xsl:apply-templates select="d:sale" />

  <Row>
    <Cell ss:Index="4"><Data ss:Type="String">Total:</Data></Cell>
    <Cell ss:StyleID="s22" ss:Formula="=SUM(R[-1]C:R[-1]C)"><Data
ss:Type="Number"></Data></Cell>
  </Row>
</Table>
<WorksheetOptions xmlns="urn:schemas-microsoft-com:office:excel">
  <Print>
    <ValidPrinterInfo/>
    <HorizontalResolution>600</HorizontalResolution>
    <VerticalResolution>600</VerticalResolution>
  </Print>
  <Selected/>
  <Panes>
    <Pane>
      <Number>3</Number>
      <ActiveRow>12</ActiveRow>
      <ActiveCol>1</ActiveCol>
    </Pane>
  </Panes>
  <ProtectObjects>False</ProtectObjects>
  <ProtectScenarios>False</ProtectScenarios>
</WorksheetOptions>
</Worksheet>
<Worksheet ss:Name="Sheet2">
  <WorksheetOptions xmlns="urn:schemas-microsoft-com:office:excel">
    <ProtectObjects>False</ProtectObjects>
    <ProtectScenarios>False</ProtectScenarios>
  </WorksheetOptions>
</Worksheet>
<Worksheet ss:Name="Sheet3">
  <WorksheetOptions xmlns="urn:schemas-microsoft-com:office:excel">
    <ProtectObjects>False</ProtectObjects>
    <ProtectScenarios>False</ProtectScenarios>
  </WorksheetOptions>
</Worksheet>
</Workbook>
</xsl:template>

<xsl:template match="d:sale">
  <Row>

```

```

    <Cell><Data ss:Type="Number"><xsl:value-of select="d:IDnum" /></
Data><NamedCell ss:Name="ID"/></Cell>
    <Cell><Data ss:Type="String"><xsl:value-of select="d:critter" /></
Data><NamedCell ss:Name="Critters"/></Cell>
    <Cell ss:StyleID="s22"><Data ss:Type="Number"><xsl:value-of select="d:price"
/></Data><NamedCell
    ss:Name="Price"/></Cell>
    <Cell><Data ss:Type="Number"><xsl:value-of select="d:quantity" /></
Data><NamedCell ss:Name="Quantity"/></Cell>
    <Cell ss:StyleID="s22" ss:Formula="=RC[-2]*RC[-1]"><Data
ss:Type="Number"><xsl:value-of select="d:total" /></Data></Cell>
  </Row>
</xsl:template>

<xsl:template match="d:date" />
<xsl:template match="d:total" />

</xsl:stylesheet>

```

Некоторые фрагменты этого примера заслуживают особого внимания. Обратите внимание, что SpreadsheetML помещается в XSLT; SpreadsheetML становится частью таблицы стилей. Помимо этого, один фрагмент SpreadsheetML воссоздать этим способом невозможно: это упомянутый ранее оператор обработки, который сообщает Windows, что данный документ описывает электронную таблицу Excel. Поэтому необходимо использовать следующий код:

```

<xsl:processing-instruction name="mso-application">progid=
"Excel.Sheet"</xsl:processing-instruction>

```

Так как XSLT не позволяет использовать пространство имен по умолчанию (без префикса) для ссылки на содержимое этого пространства, все ссылки на содержимое исходного документа содержат префикс d:, например, d:sale, d:date и т. п.

Кроме того, поскольку именованные диапазоны будут изменяться в зависимости от количества элементов sale в исходном документе, эта таблица стилей не будет генерировать элемент Names и его содержимое. В любом случае Excel заново создаст именованные диапазоны на основе элементов NamedCell. И опять основной частью этой таблицы стилей является фрагмент, где генерируются элементы Row и Cell, как показано в листинге 8.7.

Листинг 8.7

```

<xsl:template match="d:sale">
  <Row>
    <Cell><Data ss:Type="Number"><xsl:value-of select="d:IDnum" />
    </Data><NamedCell ss:Name="ID"/></Cell>

```

Элемент `xsl:template` соберет все элементы sale в исходном документе и создаст элемент Row, содержащий элементы Cell, соответствующие его содержимому. Если вы откроете в Excel SpreadsheetML, который создает эта таблица стилей (которая очень похожа на листинг 8.3, за исключением именованных диапазонов и некоторого форматирования), то получите результат, показанный на рис. 8.15.

ID Number	Critter	Price	Quantity	Total
4627	Diplodocus	\$22.50	127	\$2,857.50
3912	Brontosaurus	\$17.50	74	\$1,295.00
9845	Triceratops	\$12.00	91	\$1,092.00
9625	Vulcanodon	\$19.00	108	\$2,052.00
5903	Stegosaurus	\$18.50	63	\$1,165.50
1824	Monoclonius	\$16.50	133	\$2,194.50
9728	Megalosaurus	\$23.00	128	\$2,944.00
8649	Barosaurus	\$17.00	91	\$1,547.00
Total:				\$15,147.50

Рис. 8.15. Тестовая электронная таблица после того, как ее данные из SpreadsheetML были перенесены в другой словарь и обратно

SpreadsheetML, возможно, выглядит не очень привлекательно, но есть масса причин, почему его нужно использовать. Например, сохранение документа как SpreadsheetML обеспечивает лучший доступ к информации карты XML, описанной в предыдущем трюке, чем на данный момент предлагает GUI (Graphic User Interface, графический интерфейс пользователя) Excel. В долгосрочной перспективе очень важно, что SpreadsheetML — это переносимый формат, и вы можете обработать или создать его практически на любом компьютере с базовыми возможностями XML.

Саймон Сен-Лорен

Т Р Ю К
№ 98

Импорт данных напрямую в Excel

Используя Amazon Web Services в качестве источника данных, можно легко интегрировать оперативную информацию о книгах в электронные таблицы Excel. Этот пример импортирует рейтинговые данные о продажах определенных книг и подсчитывает средний рейтинг.

Этот трюк основывается на умении Excel выполнять веб-запросы, поэтому вам понадобится Excel 97 или более новая версия. Утилита Excel Web Queries (веб-запросы) преобразует простые таблицы HTML в электронные таблицы Excel.

ПРИМЕЧАНИЕ

Чтобы заставить все это работать, необходимо зарегистрироваться в службе Amazon и получить партнерский тег (associate tag). Для этого посетите сайт <http://www.amazon.com/associates/>. Затем необходимо перейти на <http://amazon.com/webservices/> и щелкнуть ссылку «Apply for a free developer's token» (Подать заявление на бесплатный маркер разработчика). Оба этих шага необходимы для того, чтобы начать использовать веб-службы Amazon.

В следующем коде используется несколько возможностей веб-служб Amazon и Excel. Кода вы поймете, как они соединяются, построение собственных запросов станет легкой задачей.

Этот трюк начинается стандартным запросом XML/HTTP. Мы хотим проанализировать рейтинги продаж серии Hacks издательства O'Reilly, поэтому построим стандартный запрос для получения этих результатов.

```
http://xml.amazon.com/onca/xml3?t=вставьте партнерский тег
&dev-t=вставьте маркер разработчика&PowerSearch=publisher:O'Reilly J
%20and%20keywords:Hack&type=heavy&mode=books&f=xml
```

В этом запросе при помощи Amazon Power Search указываются издательство (O'Reilly) и ключевое слово (Hack).

Далее в ответ на этот запрос необходимо получить данные Amazon и поместить их в форму, с которой сможет работать Excel. Так как веб-запросы Excel работают только с простым кодом HTML, придется преобразовать ответ Amazon. Как мы уже демонстрировали в последних двух трюках, таблицы стилей XSL — это простой способ выполнения нужной задачи.

Поместите следующий код в файл с именем excel_SalesRank.xml. Этот файл ограничит ответ Amazon только необходимыми полями и преобразует его в HTML.

Листинг 8.8

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html"/>
<xsl:template match="/">
<html xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:x="urn:schemas-microsoft-com:office:excel"
xmlns="http://www.w3.org/TR/REC-html40">
<body>
<table id="basic">
<tr>
<th bgcolor="#cccccc" colspan="3">Sales Data</th>
</tr>
<tr>
<th bgcolor="#999999">ASIN</th>
<th bgcolor="#999999">Title</th>
<th bgcolor="#999999">Sales Rank</th>
</tr>
<xsl:for-each select="ProductInfo/Details">
<tr>
<td><xsl:value-of select="Asin" /></td>
<td><xsl:value-of select="ProductName" /></td>
<td><xsl:value-of select="SalesRank" /></td>
</tr>
</xsl:for-each>
<tr><td colspan="3"></td></tr>
<tr>
<td bgcolor="#ffcc00" colspan="2" align="right">
```

продолжение ↗

Листинг 8.8 (продолжение)

```

        <b>Average Sales Rank</b>
    </td>
    <td bgcolor="#ffcc00">=ROUND(AVERAGE(C3:C<xsl:value-of
select="count(ProductInfo/Details) + 2" />),0)</td>
</tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

Этот файл принимает ответ AWS (Amazon Web Services) и преобразует его в простую таблицу HTML. Раздел `xsl:for-each` в цикле проходит через узел `Details` и добавляет строку таблицы для каждого очередного результата. Загрузив этот файл на общедоступный сервер, вы сможете просмотреть результат трансформации в веб-обозревателе, указав в запросе URL (Universal Resource Address, универсальный адрес ресурса) к файлу XSL. Измените URL с последнего шага, изменив `f=xml` на `f=http://example.com/excel_SalesRank.xsl`, как это показано в листинге 8.9.

Листинг 8.9

```

http://xml.amazon.com/onca/xml3?t=вставьте партнерский тег J
&dev-t=вставьте маркер разработчика&PowerSearch=publisher:O'Reilly J
%20and%20keywords:Hack&type=heavy&mode=books J
&f=http://example.com/excel_SalesRank.xsl

```

Вы должны увидеть таблицу (рис. 8.16) с данными из предыдущего запроса. Обратите внимание, что последняя ячейка таблицы содержит функцию Excel. Сейчас она выглядит, как ненужная тарарашина, но служит очень важной цели в Excel.

ASIN	Title	Sales Rank
0596004478	Google Hacks	14
0596004605	Mac OS X Hacks	278
0596004613	Linux Server Hacks	763
Average Sales Rank: =ROUND(AVERAGE(C3:C5),0)		

Рис. 8.16. HTML-таблица с данными продаж

Чтобы соединить два приложения, используйте файл Excel Query (IQY). Этот файл будет содержать всю информацию о запросе, включая URL, указывающий на данные. Создайте новый файл с именем `amzn_avg_sales.iqy` и добавьте код из листинга 8.10.

Листинг 8.10

WEB

1

```
http://xml.amazon.com/onca/xml3?t=insert associate tag J
&dev-t=insert developer token&PowerSearch=publisher:O'Reilly J
%20and%20keywords:Hack&type=heavy&mode=books&f=http://example.com J
/excel_SalesRank.xml
```

Первая строка сообщает Excel, что это веб-запрос. Число 1 — это номер версии веб-запроса (он всегда будет равен 1), за ним следует URL запроса AWS, который включает файл XSL. Сохраните файл и запомните, где он теперь находится.

Выполнение трюка

Чтобы выполнить трюк, дважды щелкните файл `amzn_avg_sales.iqy`. Запустится Excel, который соединится затем с сайтом по указанному URL и заполнит электронную таблицу (рис. 8.17).

Sales Data		
ASIN	Title	Sales Rank
596004478	Google Hacks	14
596004605	MacOSXHacks.....!	278
596004613	Linux Server Hacks	763
Average Sales Rank		352

Рис. 8.17. Электронная таблица Excel с рейтинговыми данными о продажах на Amazon

Теперь у вас есть некоторые полезные данные — средний рейтинг продаж книг, — помещенные в удобное для манипулирования и анализа данных приложение. Данные можно обновить в любой момент, щелкнув правой кнопкой мыши любую ячейку данных в электронной таблице и в контекстном меню выбрав команду Обновить данные (Refresh Data).

Трюк в трюке

Сложная часть этого трюка заключается в том, чтобы понять, как получить данные напрямую из AWS в Excel. Попав в Excel, данные становятся доступными для применения любых возможностей Excel: вычислений, рисования графиков, ввода данных пользователем и т. д. Вот несколько быстрых способов еще больше расширить этот пример.

Превращение запроса в динамический запрос

Чтобы не ограничивать данные статической информацией, встроенной в запрос в файле IQY, можно добавить немного интерактивности. Предположим, есть список номеров ASIN и вы хотите узнать средний рейтинг продаж, но еще не получили этот список на момент построения файла IQY. Excel предлагает возможность запросить информацию у пользователя перед созданием веб-запроса.

Единственное, что необходимо сделать, — это поместить информацию с запросом к пользователю в URL веб-запроса туда, где должна появиться информация, введенная пользователем. Измените URL внутри `amzn_avg_sales.iqy` на такой:

```
http://xml.amazon.com/onca/xml3?t=вставьте партнерский тег&dev-t=вставьте маркер J  
разработчика&PowerSearch=isbn:["ASINs"."Введите список ASIN. J  
разделенных символами (|)."]&type=heavy&mode=books&f=http:// J  
example.com/excel_SalesRank.xsl
```

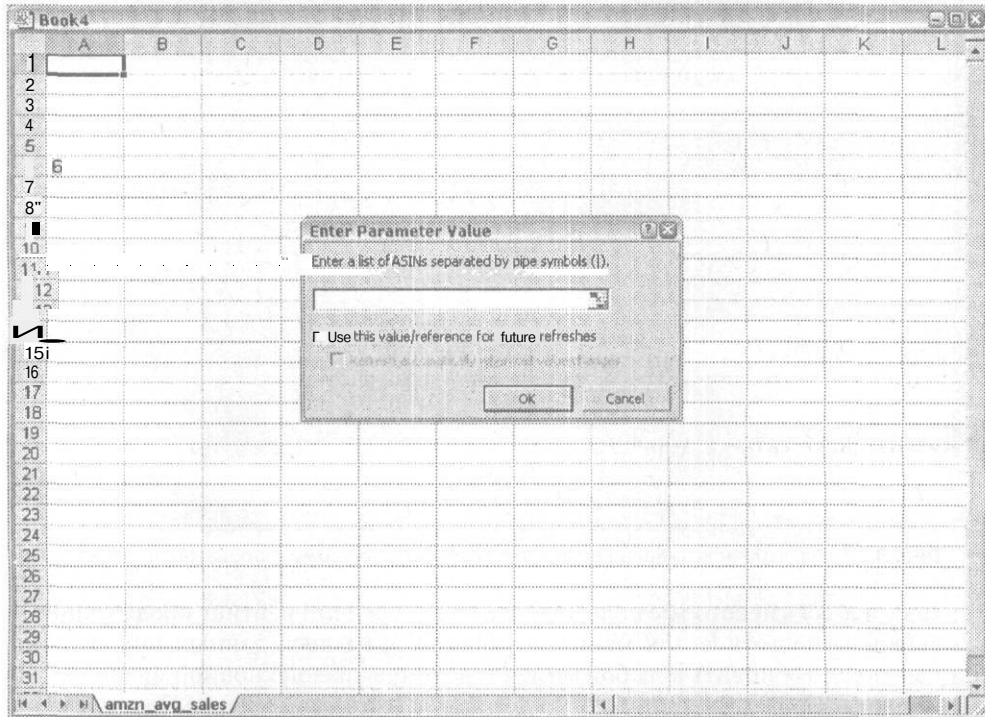


Рис. 8.18. Диалоговое окно Excel с просьбой ввести номера ASIN

Теперь при открытии файла появится сообщение с просьбой ввести номера ASIN (рис. 8.18).

Если вы не помните никаких номеров, попробуйте эти:

1565927141|0596003595|0596002246|0596002505

Вы должны получить средние рейтинги продаж для введенных номеров ASIN. Каждый раз они могут быть разными!

Использование других данных

Как и в предыдущем примере, изменение данных, с которыми вы работаете, подразумевает простое изменение URL в запросе AWS. Однако если вы хотите обработать другой набор данных (например, стоимость каждого элемента, а не рейтинг продаж), необходимо немного изменить таблицу стилей XSL.

Чтобы увидеть, как таблица стилей влияет на различия, создайте новый файл с именем `excel_PriceDiff.xml` и введите код из листинга 8.11.

Листинг 8.11

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html"/>
<xsl:template match="/">
<html xmlns:o="urn:schemas-microsoft-com:office:office"
xmlns:x="urn:schemas-microsoft-com:office:excel"
xmlns="http://www.w3.org/TR/REC-html40">
<body>
<table id="basic">
<tr>
<th bgcolor="#cccccc" colspan="4">Sales Data</th>
</tr>
<tr>
<th bgcolor="#999999">ASIN</th>
<th bgcolor="#999999">Title</th>
<th bgcolor="#999999">List Price</th>
<th bgcolor="#999999">Amazon Price</th>
</tr>
<xsl:for-each select="ProductInfo/Details">
<tr>
<td>
<xsl:value-of select="Asin" />
</td>
<td>
<xsl:value-of select="ProductName" />
</td>
<td>
<xsl:value-of select="ListPrice" />
</td>
<td>
<xsl:value-of select="OurPrice" />

```

продолжение ↗

Листинг 8.11 (продолжение)

```

</td>
</tr>
</xsl:for-each>
<tr><td colspan="3"></td></tr>
<tr>
  <td bgcolor="#ffcc00" colspan="3" align="right">
    <b>Average List Price</b>
  </td>
  <td bgcolor="#ffcc00">=ROUND(AVERAGE(C3:C<xsl:value-of
select="count(ProductInfo/Details) + 2" />),2)
  </td>
</tr>
<tr>
  <td bgcolor="#ffcc00" colspan="3" align="right">
    <b>Average Amazon Price</b>
  </td>
  <td bgcolor="#ffcc00">=ROUND(AVERAGE(D3:D<xsl:value-of
select="count(ProductInfo/Details) + 2" />),2)
  </td>
</tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

ASIN	Title	List Price	Amazon Price
596002246	Web Services Essentials (O'Reilly XML)	\$29.95	\$20.97
596003595	Writing Excel Macros with VBA, 2nd Edition	\$34.95	\$24.47
596002505	Programming .NET Web Services	\$39.95	\$27.97
1565927141	Excel 2000 in a Nutshell. A Power User's Quick Reference	\$29.95	\$20.97
		Average List Price	\$33.70
		Average Amazon Price	\$23.60

Рис. 8.19. Электронная таблица Excel с данными о цене

Загрузите этот файл XSL на общедоступный сервер и запомните URL. Создайте новый файл запроса с именем `amzn_price_diff.iqy` и используйте тот же код, что и в предыдущем примере, но измените значение переменной `f=` на URL новой таблицы стилей. Откройте файл и вы увидите новую электронную таблицу с ценой по прейскуранту и ценой Amazon (рис. 8.19).

Вывод результатов на диаграммы

Когда данные загружены в Excel, легко создавать диаграммы, чтобы с первого же взгляда получать представление о том, что они означают. Вот как добавить диаграмму в электронную таблицу:

1. Мы продолжим выполнение последнего примера. Запустите файл `amzn_price_diff.iqy`. Вы должны увидеть данные, включающие список книг, цену по прейскуранту и цену Amazon.
2. Выделите все ячейки, относящиеся к заголовкам Title, List Price и Amazon Price.
3. Выберите пункт меню Вставка ▶ Диаграмма (Insert ▶ Chart). Запустится мастер диаграмм.
4. Щелкните кнопку Готово (Finish).

Теперь вы должны получить привлекательное графическое представление данных (рис. 8.20).

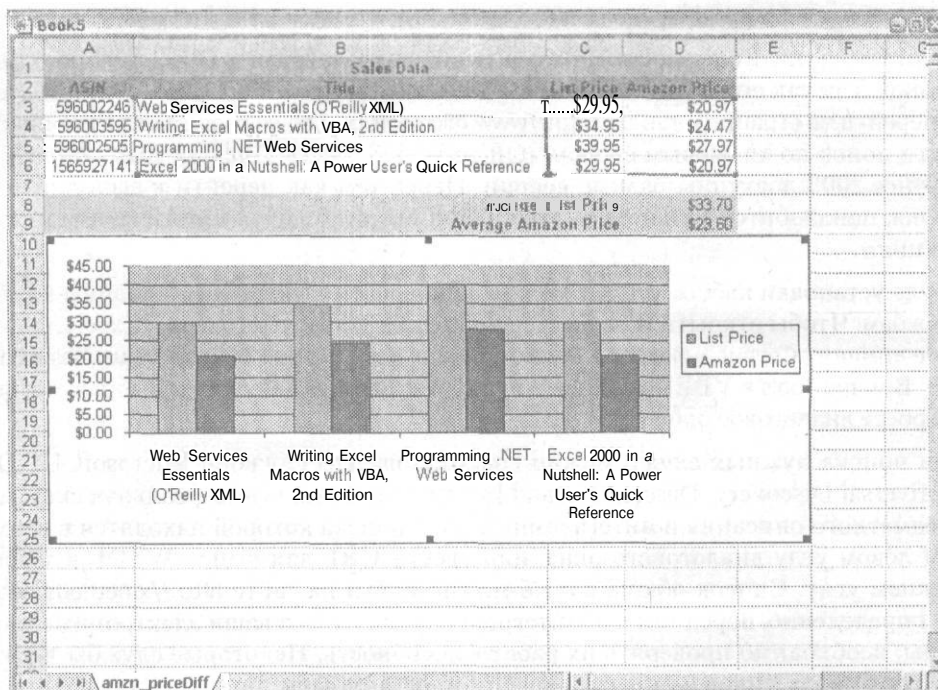


Рис. 8.20. Диаграмма для данных о цене

Пол Бош (Amazon Hacks)

Т Р Ю К
№ 99

Доступ к веб-службам SOAP из Excel

Если электронные таблицы должны иметь доступ к постоянно обновляющимся данным или если необходимо обращаться к службам на другом компьютере, установить соединение помогут веб-службы Excel (Excel Web Services).

Веб-службы, основанные на SOAP, — это ключевая составляющая планов Microsoft относительно .NET, а также широко используемая возможность в наборах утилит других производителей. SOAP (это сокращение ничего не означает) — это протокол, который использует XML для передачи информации между системами. На случай если вы будете изучать его, можем сказать, что это называется «вызывать процедуры» и «возвращать значения». Описывает веб-службы сопутствующая спецификация, WSDL (Web Service Definition Language, букв. «язык определения веб-служб»), чтобы приложения могли легко подключиться к этим службам. При помощи Microsoft Web Services Reference Tool можно создать файл WSDL и сгенерировать код VBA, который ваше приложение сможет применять для доступа к основанным на SOAP веб-службам.

ПРИМЕЧАНИЕ

В этом трюке используются возможности Excel, доступные только в Excel 2002 и Excel 2003 для Windows. В предыдущих версиях Excel они не поддерживаются, как не поддерживаются в существующих и планируемых версиях Excel для Macintosh.

Чтобы выполнить следующий пример, необходимо загрузить Office Web Services Toolkit. Так как его местоположение уже несколько раз изменялось, проще всего перейти на страницу поиска <http://www.microsoft.com/downloads/search.asp> и провести поиск по ключевым словам «Office Web Services Toolkit». Для Office XP и Office 2003 доступны разные версии. Перед тем как перейти к выполнению трюка, понадобится установить этот набор утилит, следуя прилагаемым инструкциям.

После установки набора утилит можно подключать электронные таблицы к веб-службам. Чтобы открыть Web Service References Tool (это его имя в Excel), выберите команду Сервис ► Макрос ► Редактор Visual Basic (Tools ► Macro ► Visual Basic Editor). В меню Tools в VBE вы увидите пункт Web Services References. Выбрав его, вы откроете диалоговое окно (рис. 8.21).

Для поиска нужных служб можно воспользоваться службой Microsoft UDDI (Universal Discovery, Description and Integration, букв. «универсальная система предметного описания и интеграции»), поля поиска которой находятся в верхнем левом углу диалогового окна, или ввести URL для файла WSDL в левом нижнем углу. Список общих служб можно найти на сайте <http://xmethods.net/>, но, определенно, перед тем как интегрировать службы в ваши электронные таблицы, необходимо проверять их работоспособность. Некоторые службы также требуют ввода лицензионных ключей и иногда оплаты, но в этом примере мы будем использовать бесплатную службу. Она возвращает IP-адрес для данного доменного имени.

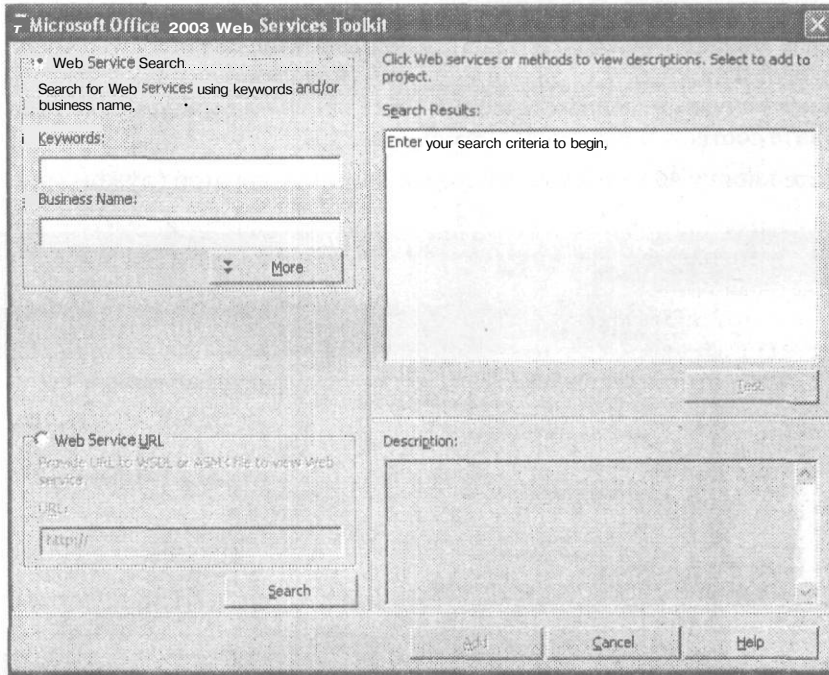


Рис. 8.21. Microsoft Office Web Services Toolkit в действии

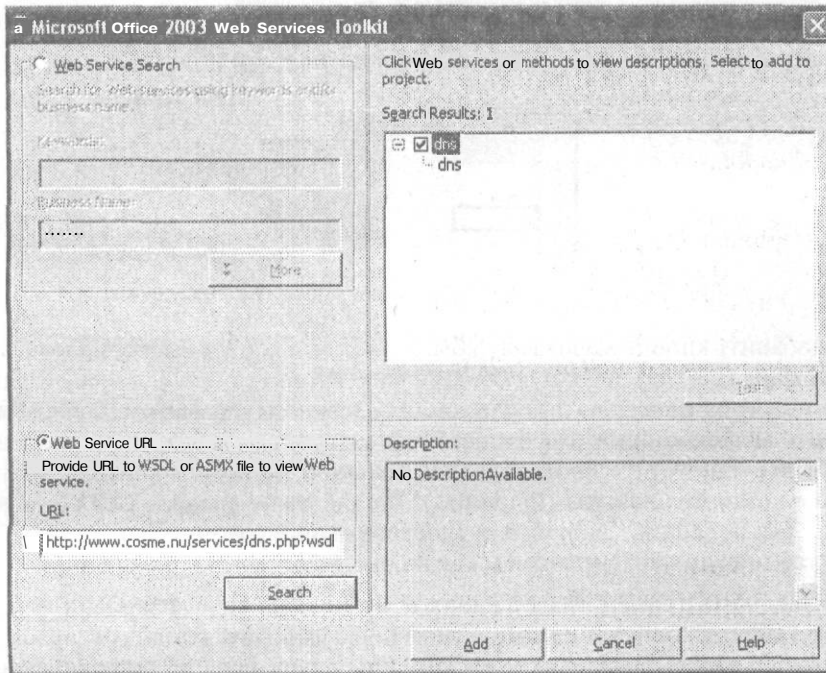


Рис. 8.22. Создание кода для веб-службы в Web Services Toolkit

Для начала нужно сообщить Excel, какую службу вы хотите использовать; в данном случае это <http://www.cosme.nu/services/dns.php?wsdl>. Введите это значение в поле URL в левом нижнем углу диалогового окна и щелкните кнопку Search. Результаты поиска служб доменных имен появятся в правом верхнем поле (рис. 8.22). Установите соответствующий флажок слева.

Щелкните кнопку Add, чтобы Excel создал код VBA для этой службы (рис. 8.23).

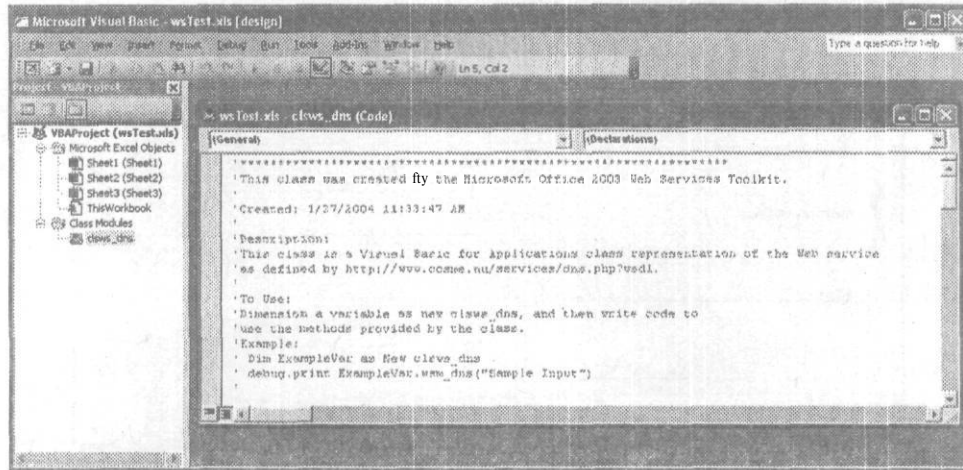


Рис. 8.23. Код VBA для доступа к службе доменных имен, генерированный Web Services Toolkit

Затем закройте VBE и создайте очень простую электронную таблицу (рис. 8.24).

	A3	A IP:
	A	B
1		
2	Name:	
3	IP:	
4		
5		

Рис. 8.24. Электронная таблица для добавления веб-служб

Чтобы добавить кнопку вызова службы, отобразите панель инструментов Элементы управления (Control), щелкнув область панелей инструментов правой кнопкой мыши и в контекстном меню выбрав команду Элементы управления (Control Toolbox). Щелкните значок кнопки, а затем поместите кнопку в любое место на электронной таблице. Щелкните кнопку правой кнопкой мыши и в контекстном меню выберите команду Свойства (Properties). В поле Name введите GetData, а в поле Caption — Get IP Address. Закройте диалоговое окно Properties. Теперь ваша электронная таблица должна выглядеть, как на рис. 8.25.

Чтобы завершить пример, правой кнопкой мыши щелкните добавленную на электронную таблицу кнопку и в контекстном меню выберите команду Исходный текст (View Code). В раскрывшемся окне введите процедуру, код которой приведен в листинге 8.12.

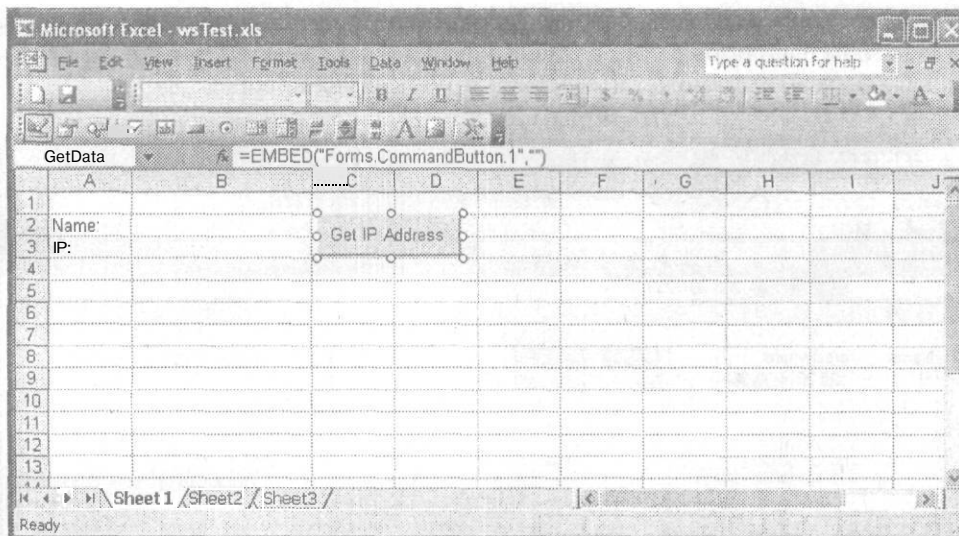


Рис. 8.25. Электронная таблица с кнопкой вызова веб-служб

Листинг 8.12

```
Private Sub GetData_Click()
    Dim info As New clsws_dns
    Dim name As String
    Dim IP As String

    name = Range("B2").Text

    IP = info.wsm_dns(name)

    Set IPRange = Range("B3")

    IPRange.Value = IP

End Sub
```

Это очень простой код. Он ссылается на объект, который для веб-службы был создан набором инструментов, и создает переменные для имени и IP-адреса. Кроме того, он берет имя из ячейки B2, вызывает веб-службу, передавая ей это имя в качестве аргумента, и затем помещает возвращенную переменную в ячейку B3. После того как вы введете код и закроете VBE, можно выйти из режима конструктора (design mode), удостоверившись, что значок с треугольником и линейкой слева на панели инструментов Элемент управления (Control) не выделен. Теперь на электронной таблице в ячейку B2 можно ввести доменное имя. Щелкните кнопку Get IP Address, и IP-адрес, соответствующий этому доменному имени, будет помещен в ячейку B3. На рис. 8.26 и 8.27 показана работа этой электронной таблицы с различными доменными именами.

Разрешение IP-адреса — одна из самых простых служб, но часто именно такие простейшие службы могут оказаться очень полезными в электронных таблицах:

например, для перевода валют, получения цен, получения почтового индекса и т. п. Чтобы использовать эти службы, не нужно даже знать о SOAP или WSDL, так как обо всем этом может позаботиться Web Services Toolkit.

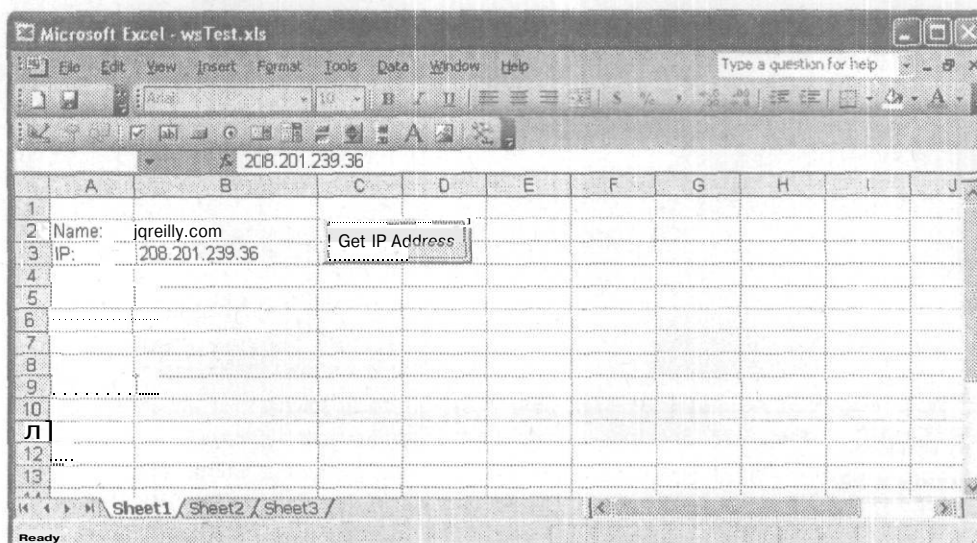


Рис. 8.26. IP-адрес доменного имени oreilly.com

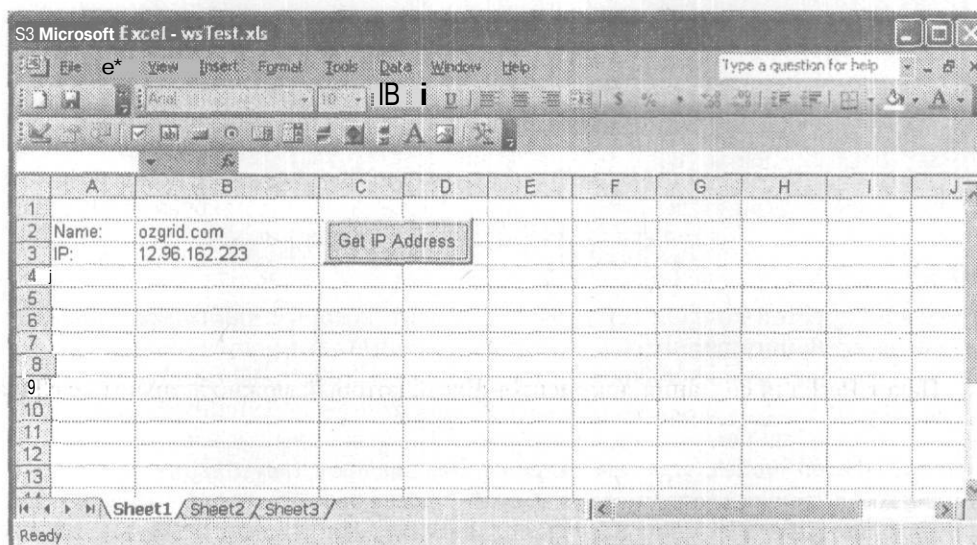


Рис. 8.27. IP-адрес доменного имени ozgrid.com

Однако необходимо упомянуть о некоторых моментах. Чтобы веб-службы могли успешно работать, компьютер должен быть подключен к сети. Определенно, вы не захотите создавать электронные таблицы, зависящие от веб-служб, если поль-

зователи, которые будут работать с ними, будут постоянно отключены от сети. (Такие электронные таблицы, как наша, которые используют веб-службы только для заполнения полей данными и не требуют постоянного подключения, вполне подходят для работы.)

Еще один важный вопрос заключается в том, что эта область находится в постоянном движении вперед. На момент написания этого раздела SOAP перешла от версии 1.1 к версии 1.2, и в процессе разработки уже находилась новая версия WSDL. Помимо этого, многие считают, что на смену UDDI также в итоге придут новые технологии. Но на сегодняшний момент можно только посоветовать обязательно проверять используемые службы и следить за выходом новых версий Office Web Services Toolkit.

Саймон Сен-Лорен



Создание электронных таблиц Excel в других средах разработки

Хотя Excel и другие программы для работы с электронными таблицами представляют собой традиционные интерфейсы построения и чтения электронных таблиц, иногда бывает необходимо создавать файлы .xls в других программах.

Создаете ли вы файлы Excel в базах данных или на сервере Apache или вы просто хотите представить информацию, хранящуюся в программе Java, пользователям Excel, есть множество пакетов, которые позволяют создавать и работать с файлами .xls без использования самой программы Excel. Хотя чаще всего эти пакеты использовать сложнее, чем SpreadsheetML, описанный в разделах «Трюк № 96. Сохранение в виде SpreadsheetML и выделение данных» и «Трюк № 97. Создание электронных таблиц при помощи SpreadsheetML», они обычно умеют создавать файлы, совместимые с любыми версиями Excel, начиная с Excel 97 (включая версии Excel для Macintosh), и часто поддерживают достаточно много возможностей Excel.

Некоторые из наиболее популярных (бесплатных) пакетов для работы с данными Excel:

Spreadsheet::WriteExcel

Пакет Perl для создания документов Excel, который можно загрузить с сайта <http://search.cpan.org/dist/Spreadsheet-WriteExcel/>

Spreadsheet::ParseExcel

Набор утилит, позволяющий программам на Perl читать файлы Excel, который можно загрузить с сайта <http://search.cpan.org/~kwitknr/Spreadsheet-ParseExcel-0.2602/>

Jakarta POI

Продукт Apache Project, обеспечивающий доступ к электронным таблицам Excel для чтения и записи при помощи Java API, который можно загрузить с сайта <http://jakarta.apache.org/poi/index.html>


```
* THIS SOFTWARE IS PROVIDED ``AS IS" AND ANY EXPRESSED OR IMPLIED
* WARRANTIES. INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
* DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
* ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
* SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
* LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
* USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
* ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
* OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
* OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
* SUCH DAMAGE.
```

```
* =====
*
* This software consists of voluntary contributions made by many
* individuals on behalf of the Apache Software Foundation. For more
* information on the Apache Software Foundation, please see
* <http://www.apache.org/>.
*/
```

```
import java.io.*;

import java.util.Random;

import org.apache.poi.poifs.filesystem.POIFSFileSystem;
import org.apache.poi.hssf.record.*;
import org.apache.poi.hssf.model.*;
import org.apache.poi.hssf.usermodel.*;
import org.apache.poi.hssf.util.*;

public class PoiDemo {

    public static void main (String[] args) throws Exception {
        short rownum;

        // создание целевого файла
        FileOutputStream out = new FileOutputStream("zingot.xls");

        // создание нового объекта рабочей книги; обратите внимание, что рабочая
        // книга и файл остаются совершенно разными отдельными объектами до самого
        // конца, когда рабочая книга записывается в файл.
        HSSFWorkbook wb = new HSSFWorkbook();

        // создание нового рабочего листа
        HSSFSheet ws = wb.createSheet();

        // создание ссылки на объект строки для дальнейшего использования
        HSSFRow r = null;

        // создание ссылки на объект ячейки
        HSSFCell c = null;
        // создание двух стилей ячеек - форматы необходимо
        // определять до того, как они будут использоваться
```

продолжение ↗

Листинг 8.13 (продолжение)

```

HSSFCellStyle cs1 = wb.createCellStyle();
HSSFCellStyle cs2 = wb.createCellStyle();
HSSFDataFormat df = wb.createDataFormat();
// создание двух объектов шрифта для форматирования
HSSFFont f1 = wb.createFont();
HSSFFont f2 = wb.createFont();

// шрифт 1 – 10 пунктов, жирный
f1.setFontHeightInPoints((short) 10);
f1.setBoldweight(HSSFFont.BOLDWEIGHT_BOLD);

// шрифт 2 – 10 пунктов, красный
f2.setFontHeightInPoints((short) 10);
f2.setColor( (short)HSSFFont.COLOR_RED );

// для первого стиля ячейки использовать шрифт 1 и выбрать формат данных
cs1.setFont(f1);
cs1.setDataFormat(df.getDataFormat("#.##0.0"));

// для второго стиля ячейки использовать шрифт 2, установить тонкую границу //
// текстовый формат
cs2.setBorderBottom(cs2.BORDER_THIN);
cs2.setDataFormat(HSSFDataFormat.getBuiltinFormat("text"));
cs2.setFont(f2);

// задать имя листа в кодировке Unicode
wb.setSheetName(0, "Test sheet",
    HSSFWorkbook.ENCODING_UTF_16 );

// создать лист с 10 строками (0-9)
for (rownum = (short) 0; rownum < 10; rownum++)
{
    // создать строку
    r = ws.createRow(rownum);

    //r.set rowNum(( short ) rownum);
    // создать шесть ячеек (0-5) (зачем использовалось += 2, станет
    // ясно позже)
    for (short cellnum = (short) 0; cellnum < 6; cellnum += 2)
    {
        // создать числовую ячейку
        c = r.createCell(cellnum);
        // заполнить числами в зависимости от позиции
        c.setCellValue(rownum * 10 + cellnum
            + (((double) rownum / 10)
            + ((double) cellnum / 100)));
        // создать строковую ячейку
        c = r.createCell((short) (cellnum + 1));

        // на каждой второй строке (вот почему +=2)
        if ((rownum % 2) == 0)
        {
            // присвоить этой ячейке стиль 1 из определенных нами

```

```

        c.setStyle(cs1);
        // присвоить ячейке строковое значение "Test"
        c.setEncoding( HSSFCell.ENCODING_UTF_16 );
        c.setCellValue( "Test" );
    }
    else
    {
        c.setStyle(cs2);
        // присвоить ячейке строковое значение "1... 2... 3..."
        c.setEncoding( HSSFCell.ENCODING_UTF_16 );
        c.setCellValue( "1... 2... 3..." );
    }
}
}

// использовать несколько формул
// перейти на строку вперед
rownum++;
r = ws.createRow(rownum);

//создать формулы.
for (short cellnum = (short) 0; cellnum < 6; cellnum += 2)
{
    //создать суммы для подходящих столбцов
    int column= 65+cellnum;
    char columnLabel=(char)column;
    String formula="SUM("+columnLabel+"1: "+columnLabel+"10)";
    c = r.createCell(cellnum);
    c.setStyle(cs1);
    c.setCellFormula(formula);
}

// записать эту рабочую книгу в выходной поток,
// не забыв закрыть файл
wb.write(out);
out.close();
}
}

```

Чтобы выполнить этот код, сначала необходимо загрузить последний бинарный файл POI и поместить главный файл POI jar (в данном случае poi-2.0-final-20040126.jar) в место, которое определяется путем к классам в соответствии с вашей платформой. При выполнении этот код не требует аргументов и создает один файл с именем zingot.xls. Если вы откроете этот файл, то увидите электронную таблицу (рис. 8.28 или 8.29, если вы работаете на Macintosh).

Логику в листинге 8.7 вряд ли можно назвать образцовой моделью конструирования электронной таблицы, но она демонстрирует базовую функциональность, необходимую для создания новых листов, ячеек и формул. Если вы будете использовать такой код для создания электронных таблиц для собственных приложений, то, несомненно, замените циклы ссылками на существующие структуры данных, целевые файлы будут более логичными и, вероятно, они будут чередоваться в зависимости от данных, а вы сможете воспользоваться преимуществами намного большего количества возможностей, чем показано здесь.

The screenshot shows the Microsoft Excel application window titled 'zingot.xls'. The spreadsheet contains a table with columns A through F and rows 1 through 13. The formula bar shows the formula for cell C13: `=SUM(C1:C10)`. The data in the spreadsheet is as follows:

	A	B	C	D	E	F
1	0	Test	2.02	Test	4.04	Test
2	10.1	2.3	12.12	1.2.3	14.14	1.2.3
3	20.2	Test	22.22	Test	24.24	Test
4	30.3	1.2.3	32.32	1.2.3	34.34	1.2.3
5	40.4	Test	42.42	Test	44.44	Test
6	50.5	1.2.3	52.52	1.2.3	54.54	1.2.3
7	60.6	Test	62.62	Test	64.64	Test
8	70.7	1.2.3	72.72	1.2.3	74.74	1.2.3
9	80.8	Test	82.82	Test	84.84	Test
10	90.9	1.2.3	92.92	1.2.3	94.94	1.2.3
11						
12						
13	454.5		474.7		494.9	

Рис. 8.28. Электронная таблица, созданная программой на Java, в Excel для Windows

The screenshot shows the Microsoft Excel application window titled 'zingot.xls' on a Macintosh. The spreadsheet content is identical to the Windows version shown in Figure 8.28. The data is as follows:

	A	B	C	D	E	F
1	0	Test	2.02	Test	4.04	Test
2	10.1	2.3	12.12	1.2.3	14.14	1.2.3
3	20.2	Test	22.22	Test	24.24	Test
4	30.3	1.2.3	32.32	1.2.3	34.34	1.2.3
5	40.4	Test	42.42	Test	44.44	Test
6	50.5	1.2.3	52.52	1.2.3	54.54	1.2.3
7	60.6	Test	62.62	Test	64.64	Test
8	70.7	1.2.3	72.72	1.2.3	74.74	1.2.3
9	80.8	Test	82.82	Test	84.84	Test
10	90.9	1.2.3	92.92	1.2.3	94.94	1.2.3
11						
12						
13	454.5		474.7		494.9	

Рис. 8.29. Электронная таблица, созданная программой на Java, в Excel для Macintosh

Еще одну возможность POI определенно стоит напомнить, если вы создаете электронные таблицы, предназначенные для использования на динамических веб-сайтах. Можно комбинировать POI с Cocomo, каркасом приложений от Apache, который использует документы XML и другие источники для генерации содержимого и к которому удобно обращаться из веб-обозревателей. Узнать об этом подробнее и изучить примеры вы сможете в статье XML.com на веб-сайте <http://www.xml.com/pub/a/2003/11/22/cocomo-excel.html>.

Саймон Сен-Лорен

Словарь терминов

Project Explorer

Окно в редакторе Visual Basic (Visual Basic Editor, VBE), в котором можно просмотреть все объекты, относящиеся к определенной открытой рабочей книге.

Visual Basic for Applications, VBA

Стандартный макроязык, используемый в большинстве продуктов Microsoft Office Suite. Слово «приложения» (Applications) можно заменить любой из программ семейства Office, в которых используется этот язык, например, Excel. Язык VBA создан на основе языка Visual Basic (VB), который, в свою очередь, произошел от языка Basic. Фундаментальное различие между VBA и VB заключается в том, что VBA (как подразумевает его название) можно использовать только в поддерживающих его приложениях, чаще всего — в одном из продуктов семейства Microsoft Office.

Аргументы (Arguments)

Информация, которая может потребоваться формуле или функции для возвращения значения. Для выполнения большинства функций в Excel требуется указать аргументы. Аргументы заключаются в круглые скобки и разделяются запятыми.

Булевский (Boolean)

Математический способ выражения логических операторов. Булевское значение — это Истина (True) или Ложь (False).

Диалоговое окно (Dialog, или Dialog box)

Всплывающее окно, которое запрашивает ввод информации пользователем, например, диалоговое окно Открытие документа (File Open).

Диапазон (Range)

Область, состоящая из одной или нескольких ячеек, смежных или несмежных.

Имя (Name)

Удобное для человеческого восприятия название, которое можно присвоить константе, диапазону ячеек или переменной.

Константа (Constant)

Именованный элемент, представляющий неизменное значение.

Круглые скобки (Parentheses)

Круглые скобки (), которые в математике используются для обозначения группировки. В Excel количество открывающих скобок всегда должно быть равным количеству закрывающих.

Лист (Sheet)

Общий термин, который используется для обозначения рабочего листа, листа диаграммы или листа макроса.

Макрос (Macro)

Действие или набор действий, которые используются для автоматизации задач. Excel позволяет записывать действия и в дальнейшем использовать их как макросы.

Мастер (Wizard)

Последовательность окон, которая помогает пользователю в выполнении последовательности шагов.

Метод (Method)

Процедура (см. Процедура (Procedure)), которая ведет себя как объект (см. Объект (Object)).

Модуль (Module)

Независимый модуль, в котором пишется и хранится код VBA.

Надстройка (Add-In)

Рабочая книга, сохраненная с расширением **.XLA**. Она открывается как скрытая рабочая книга, если установить ее командой меню Сервис ► Надстройки (Tools ► Add-Ins). Надстройки обеспечивают дополнительную функциональность Excel.

Несмежный (Non-Contiguous)

Диапазон ячеек, в котором границы некоторых ячеек не соединены. Например, диапазон A1:C20, E10:F100 представляет собой несмежный диапазон.

Объект (Object)

Компонент приложения (см. Приложение (Application)), например, рабочий лист, ячейка или диаграмма. В Excel предусмотрено более 100 объектов и они лежат в основе функциональности кода VBA.

Отладка (Debugging)

Процесс исключения всех возможных ошибок в коде.

Ошибка (Bug)

Ошибка в коде. К сожалению, некоторые ошибки могут оказаться незамеченными большинством строгих тестов, и, следовательно, от них бывает очень сложно избавиться.

Ошибка выполнения (Runtime Error)

Ошибка в коде VBA, которая происходит во время выполнения кода. Большинство ошибок выполнения присвоены числовые значения, помогающие устранить ошибки.

Панель инструментов (Toolbar)

Один из множества объектов Excel, который используется для хранения элементов меню. Строка меню рабочего листа — это панель инструментов, содержащая такие элементы меню, как Файл (File), Правка (Edit), Вид (View), Вставка (Insert) и др.

Панель инструментов Формы (Forms Toolbar)

Панель инструментов, на которой хранятся элементы управления, встроенные в Excel (то есть не элементы управления ActiveX).

Панель инструментов Элементы управления (Control Toolbox Toolbar)

Панель инструментов, содержащая элементы управления ActiveX (см. Элементы управления ActiveX (ActiveX Controls)).

Переменная (Variable)

Именованный элемент, содержащий значение, которое можно изменять во время использования переменной.

Подпроцедура (Sub)

Тип процедуры, которая не возвращает значения (см. Процедура (Procedure)). Подпроцедуры часто используются для обработки событий.

Пользовательская функция (User-Defined Function)

Функция (см. Функция (Function)), написанная в VBA специально для того, чтобы возвращать значение на основе индивидуальных вычислений.

Приложение (Application)

Термин для обозначения программы. Excel является приложением; некоторые пользователи называют приложениями свои электронные таблицы.

Процедура (Procedure)

Именованная последовательность операторов, выполняемая целиком. Например, функция и подпроцедура — это типы процедур (см. Функция (Function) и Подпроцедура (Sub)).

Рабочая книга (Workbook)

Контейнер для одного или нескольких листов или рабочих листов. Каждый файл .xls представляет собой рабочую книгу.

Рабочий лист (Worksheet)

Один из фундаментальных объектов Excel, в котором размещены ячейки. Все рабочие книги должны содержать, по крайней мере, один рабочий лист, и хотя бы один рабочий лист должен быть видимым. Единственное ограничение на количество открытых рабочих листов накладывается памятью, доступной на компьютере.

Разбирать (Parse)

Разделять команду или строку символов на составляющие ее компоненты.

Редактор Visual Basic (Visual Basic Editor, VBE)

Также называется средой Visual Basic. VBE — это интерфейс Excel, позволяющий пользователям получать доступ ко всем элементам, относящимся к VBA.

Свойство (Property)

Именованный атрибут объекта (см. Объект (Object)). Например, адрес определенной ячейки — это свойство ячейки или объекта диапазона.

Символы операций (Operators)

Математические символы, которые предписывают формуле или функции выполнить определенную задачу; например, символ операции + говорит, что необходимо выполнить сложение.

Скобки (Brackets)

См. Круглые скобки (Parentheses).

Смежный (Contiguous)

Диапазон ячеек, границы всех ячеек в котором соединены, между ними нет промежутков. Например, A1:C20 — это смежный диапазон.

Событие (Event)

Сообщение от Excel вашему коду о том, что что-то произошло. Например, щелчок мышью или закрытие-открытие рабочей книги — это события.

Строка (String)

Линейная последовательность символов. Например, слово «морковь» — это строка алфавитных символов.

Форма UserForm

Объект Excel, который можно использовать для размещения элементов управления ActiveX. Эти формы можно вставлять и модифицировать только в VBE.

Формула (Formula)

Последовательность значений, ссылок на ячейки, имен, функций или символов операций в ячейке, которые совместно производят новое значение. Формулы Excel начинаются с = (знак равенства).

Функция (Function)

Тип процедуры (см. Процедура (Procedure)), возвращающий значение. В Excel есть встроенные функции, например, СУММ (SUM), но можно создавать собственные пользовательские функции (см. Пользовательская функция (User-Defined Function)).

Цикл (Loop)

Непрерывное выполнение одной или нескольких строк кода Visual Basic for Applications (VBA), пока не будет выполнено условие остановки цикла.

Частный модуль (Private Module)

Модуль, который используется только определенным объектом, например рабочей книгой, листом или формой UserForm.

Шаблон (Template)

Предопределенный остов, используемый в качестве стандартной базы и в который можно вводить данные или модифицировать.

Электронная таблица (Spreadsheet)

Приложение или программа, в которой текст, числа или функции можно вводить в матрицу отдельных ячеек.

Элементы управления ActiveX (ActiveX Controls)

Полностью программируемые объекты, которые можно применять для настройки приложений. В Excel часто используются командные кнопки (Command button), переключатели (Option button) и др.

Алфавитный указатель

.slk 58
.xlm 58
.xls 23, 58
.xlt 37
.xlw 22, 23

A-X

AutoFilter 187
doughnut chart 160
Excel
 значок 50
IP-адрес
 разрешение 269
OpenOffice.org 59
PivotCharts 128
PivotTables 126
Project Explorer 32, 35
scatter chart 156
URL 40
VBE 29, 32, 35
Visual Basic
 редактор 29, 32
XLSTART 38
XML 235
 XML-схема 240, 242
XSLT 251

A

абсолютная ссылка 49, 51
автозаполнение 64
автофильтр 87, 187
адрес ячейки 49, 51
активная ячейка 47
анализ 107

Б-В

база гиперссылки 40
внешняя ссылка 52
время
 формат 193
вставка
 имени 124
 специальная 100, 130, 151, 177

вставка (*продолжение*)

 столбца 185
 строки 119
выделение
 ячеек 45
вычисления
 вручную 17

Г

гиперссылка 39, 124
 база 40
 группа 26

Д

данные
 повторяющиеся 15, 47
 проверка 46, 60, 122
 распределение 15
 сортировка 15
 тип 250
 формат 250
дата
 формат 177, 193
диаграмма
 сводная 56
диаграмма
 динамическая 147, 148
 заголовок 166
 кольцевая 160
 круговая
 объемная 140
 разрезанная 139
 с двумя рядами 141
 перетаскивание данных 150
 строка формул 151
 точечная 156
диаграммы
 сводные 128
диапазон
 именованный
 динамический 115, 143, 146, 148
 именованный 60, 61, 109, 125
 прокрутки 41
динамическая диаграмма 147, 148

З

заголовок 128
 диаграммы 166
 оси 166
 заливка 157
 защита
 листа 45
 формулы 45
 знак
 \$ 51
 %" 93
 = 52
 @ 52
 апостроф 34, 61, 79, 111
 восклицательный
 знак 166
 доллара 51, 81, 111, 170
 кавычка
 одиночная 111
 кавычки 66
 коммерческого at 52
 минус 191
 отображение 93
 подчеркивания 114
 равенства 52, 136, 166
 скобки
 фигурные 152
 значок
 Excel 50

И

имя 39, 53
 вставка 124
 кодовое 212
 присвоение 109
 создание 114
 составное 114
 исходный текст 26, 39
 итоги 96

К

камера 132
 книга
 личная макросов 33, 38
 рабочая 15
 скрытая 38
 кнопка
 выделения листа 45
 строки 41
 кнопки
 прокрутки ярлычков 40
 кольцевая диаграмма 160

константа 152
 контекстное меню
 Excel 50
 кнопка прокрутки
 ярлычков 40
 копирование
 значений 100
 формул 100
 круговая диаграмма
 объемная 140
 разрезанная 139
 с двумя рядами 141

Л

легенда 154
 линии сетки 154
 линия 71
 лист
 выделение 45
 добавление 37
 защита 45
 макроса XLM 58
 рабочий 15
 скрытие 53, 57
 ярлычок 26, 37
 личная книга
 макросов 33, 38

М

макрос
 назначение клавиши 44
 маркер
 заполнения 64, 176, 185
 массив 67, 171, 172, 183
 массив
 констант 152
 мастер
 функций 107
 масштаб 125
 меню
 контекстное 40
 метаданные 250
 метод
 Application.OnTime 210

Н

надстройки 107, 121

О

область
 прокрутки 41

область *(продолжение)*
рабочая 22, 23
область построения
формат 157
объект
Sheet 39
окна
рабочих книг 22
окно
скрытое 33
оператор
сравнения 206
ось
заголовок 166
формат 154, 157
отмена 95
относительная ссылка 49, 51
ошибка
#####" 106, 192
#N/A 119, 167, 178
#NUM!" 204
#REF! 56
#REF!" 62, 79
#VALUE!" 183
#ЗНАЧ!" 183
#ЧИСЛ!" 204
ошибки 178

П

панель инструментов
пользовательская 50
папка
XLSTART 38
переключатель 74
пересчет 17, 33
пересчитываемая
функция 33
повторяющиеся данные 15, 47
пользовательская панель
инструментов 50
правка
связи 54
примечание
настройка 83
проверка 101
данных 46, 60
проектирование
таблицы 15
процедура
Worksheet_Deactivate 51
пустые
столбцы 15
строки 15, 16
ячейки 15

Р

рабочая книга 15
печать 30
размер 55
сохранение 28
рабочая область 22, 23
рабочие листы
группировка 26
автоматическая 26
рабочий лист 15
добавление 31
защита 31
имя 31
скрытие 35
ярлычок 26
размер
книги 55
файла 55
рамка 122, 123, 151
распределение
данных 15
редактор Visual Basic 29, 32, 35
ряд данных
формат 154, 157

С

сводная диаграмма 56
сводная таблица 56
свойство
Application.ScreenUpdating 209
Enable 51
ScrollArea 42
Visible 35
связи
правка 54
сетка 154
символ
\$ 51
%" 93
= 52
@ 52
апостроф 34, 61, 79, 111
восклицательный знак 166
доллара 51, 81, 111, 170
кавычка
одиночная 111
кавычки 66
код 80
коммерческого at 52
минус 191
отображение 93
подчеркивания 114
равенства 52, 136, 166

символ *(продолжение)*

- скобки
 - фигурные 152
 - удаление 80
 - шаблона 122
- символическая ссылка 58
- скрытая книга 38
- скрытие
 - столбца 188
 - строки 168, 187
- событие
 - Before Print 30
 - Before Save 28
 - worksheet_Activate 42
 - листа 26
- события 28
- сортировка 15
- специальная вставка 151
- список 74, 128, 173
- ссылка 57
 - абсолютная 49, 51, 70, 170
 - внешняя 52
 - на целые столбцы 16
 - относительная 49, 51, 70, 111, 170
- столбец
 - вставка 185
 - выделение 41
- столбцы
 - пустые 15
 - цвет
 - чередование 69
- строка
 - вставка 119
 - выделение 41
 - кнопка 41
- строка формул
 - диаграмма 151
- строки
 - пустые 15, 16
 - цвет
 - чередование 69

T

- таблица
 - конец 55
 - правила 128
 - проектирование 15
 - сводная 56, 126
 - итог 135
 - формат 126
 - форматирование 16
- текст
 - выравнивание 80

- точечная диаграмма 156
- транспонирование 177

У

- указатель
 - листов 38
- условное
 - форматирование 47

Ф

- файл
 - размер 55
- фильтр
 - Автофильтр 87
- флажок 73
- формат 16, 93
 - HTM 58
 - HTML 58
 - SYLK 58
 - времени 193
 - даты 177, 193
 - области построения 157
 - оси 154, 157
 - пользовательский 90
 - ряда данных 154, 157
 - таблицы 126
 - числовой 90
- форматирование 16
 - условное 47, 62, 66, 67, 69, 172
- формула
 - защита 45
 - массива 17, 67, 171, 172, 183
- формулы
 - отображение 160
- функции 65
 - мастер 107
 - поиска 184
- функция
 - ADDRESS 76, 122, 175
 - AND 71, 74
 - AVERAGE 187
 - CELL 225
 - CHOOSE 189
 - CODE 80
 - COLUMN 175
 - CONCATENATE 149
 - COUNT 116, 118
 - COUNTA
 - 71, 115, 116, 119, 121, 137, 144
 - COUNTIF 48, 122, 182
 - DATEDIF 106
 - DCOUNT 180, 205
 - DCOUNTA 180, 181

функция (продолжение)

DGET 204
 DMAX 180
 DMIN 180
 DPRODUCT 180
 DSUM 68, 178, 183, 184
 EDATE 107
 EOMONTH 107
 FIND 81, 199
 GETPIVOTDATA 135
 HYPERLINK 124
 IF 137
 INDEX 149
 INDIRECT
 61, 78, 79, 122, 175, 194, 196
 LARGE 136
 LEFT 81
 MATCH 118, 119, 121, 122
 MAX 106, 116, 136, 193
 MID 200, 225
 MIN 106, 193
 MOD 71, 182
 MONTH 121
 N 170
 NETWORKDAYS 107
 NOT 172
 NOW 180
 OFFSET
 115, 116, 119, 121, 122, 137, 144, 146
 RAND 86, 87
 ROW 70, 71, 182
 SERIES 151
 SMALL 136
 SUBSTITUTE 191
 SUBTOTAL 187
 SUM 68, 105, 176
 SUMIF 67
 SUMPRODUCT 184
 TEXT 81, 193
 TODAY 121, 180
 TRIM 191
 VLOOKUP 184, 194
 WEEKNUM 107, 121
 АДРЕС 76, 122, 175
 БДПРОИЗВЕД 180
 БДСУММ 68, 178, 183, 184
 БИЗВЛЕЧЬ 204
 БСЧЁТ 205
 БСЧЁТА 180, 181
 ВПР 184, 194
 ВЫБОР 189
 ГИПЕРССЫЛКА 124
 ДАТАМЕС 107
 ДВССЫЛ
 61, 78, 79, 122, 175, 194, 196

функция (продолжение)

ДЕНЬНЕД 121
 ДМАКС 180
 ДМИН 180
 ЕСЛИ 137
 И 71, 74
 ИНДЕКС 149
 КОДСИМВ 80
 КОНЕЦМЕСЯЦА 107
 ЛЕВСИМВ 81
 МАКС 106, 116, 136, 193
 МЕСЯЦ 121
 МИН 106, 193
 НАИБОЛЬШИЙ 136
 НАИМЕНЬШИЙ 136
 НАЙТИ 81, 199
 НЕ 172
 ОСТАТ 71, 182
 пересчитываемая 33, 86, 180
 ПОДСТАВИТЬ 191
 ПОИСКПОЗ 118, 119, 121, 122
 ПОЛУЧИТЬ.ДААННЫЕ.СВОДНОЙ.
 ТАБЛИЦЫ 135
 пользовательская 65
 ПРОМЕЖУТОЧНЫЕ.ИТОГИ 187
 ПСТР 200, 225
 РЯД 151
 СЕГОДНЯ 121, 180
 СЖПРОБЕЛЫ 191
 СЛЧИС 86, 87
 СМЕЩ
 115, 116, 119, 121, 122, 137, 144, 146
 СРЗНАЧ 187
 СТОЛБЕЦ 175
 СТРОКА 70, 71, 182
 СУММ 68, 105, 176
 СУММЕСЛИ 67
 СУММПРОИЗВ 184
 СЧЁТ 116, 118, 180
 СЧЁТЕСЛИ 48, 122, 182
 СЧЁТЗ 71, 115, 116, 119, 121, 137, 144
 СЦЕПИТЬ 149
 ТДАТА 180
 ТЕКСТ 81, 193
 текстовая 81
 Ч 170
 ЧИСТРАБДНИ 107
 ЯЧЕЙКА 225

Ц-Ч

цвет 49
 чередование 69
 шрифта
 белый 63

числа
выравнивание 80

Ш

шаблон 36, 37
шкала 157
шрифт
цвет
белый 63

З

эффект
объемности 71
эффекты
трехмерные 71

Я

ярлычок
листа 26
ячейка
адрес 49, 51
активная 47
выравнивание 80
защита 44
именованная 109
формат
граница 71
ячейки
выделение 45, 47
выравнивание 16
объединение 16
пустые 15, 173

Рэина Холи, Дэвид Холи

Excel. Трюки

Перевела с английского Е. Шикарева

Главный редактор	<i>Е. Строганова</i>
Заведующий редакцией	<i>А. Кривоцов</i>
Руководитель проекта	<i>И. Шапошников</i>
Литературный редактор	<i>В. Рычков</i>
Художник	<i>Н. Биржаков</i>
Иллюстрации	<i>С. Романов</i>
Корректоры	<i>Н. Филатова, И. Смирнова</i>
Верстка	<i>А. Зайцев</i>

Лицензия ИД № 05784 от 07.09.01.

Подписано в печать 24.12.04. Формат 70X100/16. Усл. п. л. 23,22. Тираж 4000 экз. Заказ № 9060.

ООО «Питер Принт», 194044, Санкт-Петербург, пр. Б. Сампсониевский, д. 29а.

Налоговая льгота — общероссийский классификатор продукции ОК 005-93, том 2; 953005 — литература учебная.

Отпечатано с готовых диапозитивов в ФГУП «Печатный двор» им. А. М. Горького

Министерства РФ по делам печати, телерадиовещания и средств массовых коммуникаций.

197110, Санкт-Петербург, Чкаловский пр., 15.

ПРЕДСТАВИТЕЛЬСТВА ИЗДАТЕЛЬСКОГО ДОМА «ПИТЕР»
предлагают эксклюзивный ассортимент компьютерной, медицинской,
психологической, экономической и популярной литературы

РОССИЯ

Москва м. «Калужская», ул. Бутлерова, д. 176, офис 207, 240; тел./факс (095) 777-54-67;
e-mail: sales@piter.msk.ru

Санкт-Петербург м. «Выборгская», Б. Сампсониевский пр., д. 29а;
тел. (812) 103-73-73, факс (812) 103-73-83; e-mail: sales@piter.com

Воронеж ул. 25 января, д. 4; тел. (0732) 39-61 -70;
e-mail: piter-vrn@vmail.ru; piter@comch.ru

Екатеринбург ул. 8 Марта, д. 2676; тел./факс (343) 225-39-94, 225-40-20;
e-mail: piter-ural@r66.ru

Нижний Новгород ул. Премудрова, д. 31а; тел. (8312) 58-50-15, 58-50-25;
e-mail: piter@infonet.nnov.ru

Новосибирск ул. Немировича-Данченко, д. 104, офис 502;
тел./факс (3832) 54-13-09, 47-92-93, 11-27-18, 11-93-18; e-mail: piter-sib@risp.ru

Ростов-на-Дону ул. Калитвинская, д. 17в; тел. (8632) 95-36-31, (8632) 95-36-32;
e-mail: jupiter@rost.ru

Самара ул. Новосадовая, д. 4; тел. (8462)37-06-07; e-mail: piter-volga@sama.ru

УКРАИНА

Харьков ул. Суздальские ряды, д. 12, офис 10-11; тел. (057) 712-27-05; 712-40-88;
e-mail: piter@tender.kharkov.ua

Киев пр. Красных Казаков, д. 6, корп. 1; тел./факс (044) 490-35-68, 490-35-69;
e-mail: office@piter-press.kiev.ua

БЕЛАРУСЬ

Минск ул. Бобруйская, д. 21, офис 3; тел./факс (37517) 226-19-53; e-mail: piter@mail.by



Ищем зарубежных партнеров или посредников, имеющих выход на зарубежный рынок.
Телефон для связи: **(812) 103-73-73.**
E-mail: grigorjan@piter.com



Издательский дом «Питер» приглашает к сотрудничеству авторов.
Обращайтесь по телефонам: **Санкт-Петербург - (812) 327-13-11,**
Москва - (095) 777-54-67.



Заказ книг для вузов и библиотек: (812) 103-73-73.
Специальное предложение - e-mail: kozin@piter.com



EXCEL. ТРЮКИ *100 профессиональных примеров*

За последние десять лет Excel фундаментально изменил способы работы с числовыми данными, но большинство его возможностей все еще остается в тени. Книга «Excel. Трюки» помогает читателю перейти сразу же к практическим инструментам и методам анализа, обработки и представления данных.

Прочтите эту книгу и вы узнаете, как:

- разрабатывать информативные и сложные диаграммы;
- писать макросы, которые будут автоматически выполнять задачи максимально эффективно и предсказуемо;
- создавать сложные сводные таблицы, сохраняя в секрете исходные данные;
- автоматически форматировать данные в соответствии с заданными правилами;
- сразу же структурировать данные для дальнейшей обработки;
- использовать XML и веб-службы для публикации отчетов.

Эта книга написана для тех, кто знает, что Excel гораздо мощнее, чем кажется на первый взгляд. Она позволяет работать с Excel на потрясающе высоком уровне. Добейтесь большего от ваших данных и Excel.



Посетите веб-сайт издательства O'Reilly: www.oreilly.com

Посетите наш веб-магазин: <http://www.piter.com>

ISBN 5-469-00384-1



9 785469 003847