**VBA в сводных таблицах**

В Excel 5 впервые была реализована поддержка нового макроязыка Visual Basic for Applications (VBA). VBA позволяет выполнять действия, которые обычно реализуются в Excel, но намного быстрее и точнее. Если вам доводилось прежде сталкиваться с VBA-программами, то вы знаете, что очень часто они позволяют с помощью всего одного щелчка мышью получать результаты, на которые в случае применения обычных средств Excel уходит несколько часов, а то и дней.[[1]](#footnote-1) Не стоит пугаться сложностей VBA. 90% программного кода генерируется благодаря функции записи макросов, а о том, как написать остальные 10%, будет рассказано ниже.

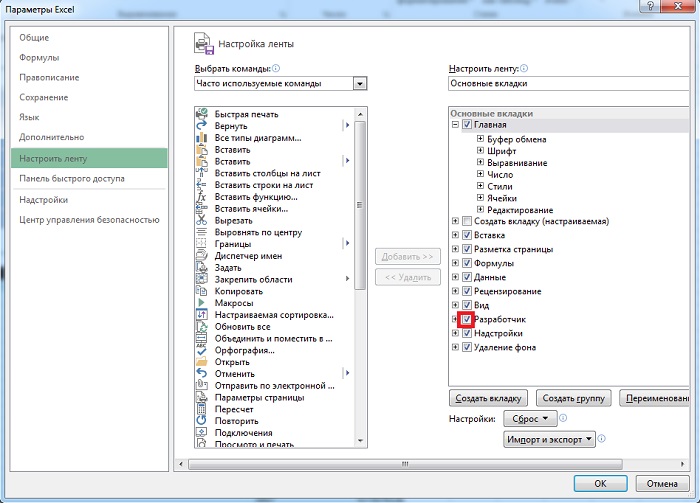


Рис. 1. Окно *Параметры Excel*

**Включение доступа к VBA в Excel**

По умолчанию доступ к VBA в Excel 2013 отключен. Прежде чем начать работать с VBA, нужно активизировать его в диалоговом окне Центр управления безопасностью. Выполните следующие действия:

1. Выберите команду *Файл* –> *Параметры*.
2. В диалоговом окне *Параметры Excel* выберите категорию *Настроить ленту* (рис. 1).
3. В находящемся справа списке отображается перечень основных вкладок Excel. По умолчанию флажок для вкладки *Разработчик* не установлен. Установите его, после чего вкладка *Разработчик* отобразится на ленте. Щелкните ОК для закрытия окна *Параметры Excel*.
4. Выберите вкладку ленты *Разработчик*. Нам понадобится группа команд *Код*, в состав которой входят кнопки *Visual Basic Editor*, *Макросы*, *Запись макроса* и *Безопасность макросов* (рис. 2).
5. Щелкните на кнопке *Безопасность макросов*. На экране появится диалоговое *окно Центр управления безопасностью*, в котором можно выбрать одну из четырех настроек, задающих уровень безопасности при работе с макросами. Названия этих настроек изменились по сравнению с названиями, применяемыми в более ранних версиях Excel.
6. Установите один из следующих переключателей.

* *Отключить все макросы с уведомлением*. При открытии рабочей книги, содержащей макросы, на экране появится сообщение о том, что в файле имеются макросы. Если хотите, чтобы эти макросы выполнялись, щелкните на кнопке *Параметры* и установите флажок *Включить это содержимое*. Это позволит VBA выполнять макросы, но вам придется явным образом разрешать их запуск при загрузке Excel.
* *Включить все макросы*. Поскольку она разрешает выполнение абсолютно всех макросов, содержащихся в рабочей книге (в том числе и зловредных), разработчики из Microsoft настоятельно не рекомендуют ее использовать.

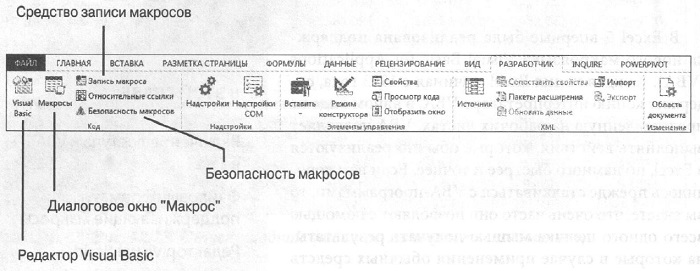


Рис. 2. Доступ к инструментам VBA реализуется через вкладку *Разработчик*

**Файловые форматы, поддерживающие макросы**

Файловый формат, используемый по умолчанию в Excel 2013, — Книга Excel с расширением .xslx. Этот формат не поддерживает макросы, т.е. в открытой книге можно создавать макросы и запускать их, но нельзя сохранять в файле. Ниже перечислены файловые форматы, которые полностью поддерживают макросы:

* Книга Excel с поддержкой макросов с расширением .xlsm. Этот файловый формат основан на XML и предназначен для хранения рабочих книг Excel и макросов. Пожалуй, этот формат является предпочтительным по причине его компактности и меньшей склонности к появлению ошибок.
* Двоичная книга Excel с расширением .xlsb.
* Устаревший формат книга Excel 97–2003 с расширением .xls.

**Редактор Visual Basic**

Для запуска редактора Visual Basic в Excel используйте комбинацию клавиш Alt+F11 или выберите команду *Разработчик* –> *Visual Basic*. Редактор Visual Basic содержит три основных раздела (рис. 3). Если вы работаете в VBA впервые, то некоторые из этих элементов могут быть скрыты. Следуйте приведенным ниже инструкциям и отобразите все элементы.

* *Project Explorer (Проекты)*. На этой панели представляется иерархическая структура всех открытых рабочих книг. Раскройте иерархическое дерево, чтобы увидеть все рабочие листы и модули кода, представленные в рабочей книге. Если панель *Project Explorer* не отображается на экране, вызовите ее с помощью комбинации клавиш Ctrl+R.
* *Properties (Свойства)*. Используется при программировании пользовательских форм, написании стандартного кода и вызывается клавишей быстрого доступа F4.
* *Code*. Это область, где вводится код. Код хранится в одном или нескольких модулях, присоединенных к рабочей книге. Чтобы добавить модуль кода в рабочую книгу, используйте команду *Insert –> Module (Вставить –> Модуль)*.

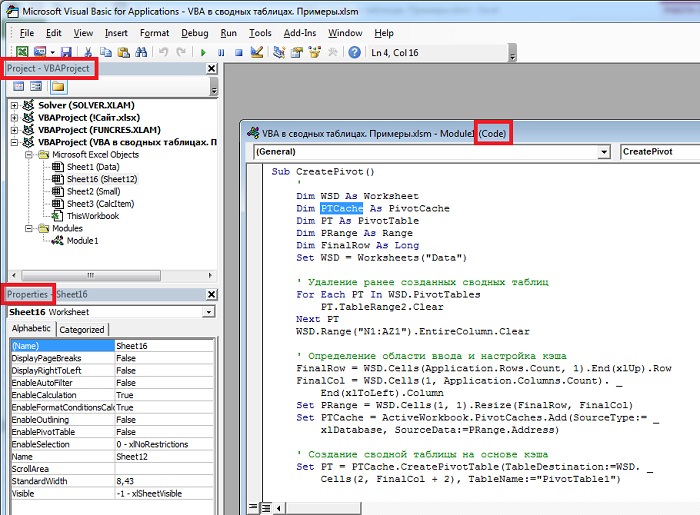


Рис. 3. Окно редактора Visual Basic

**Некоторые инструменты Visual Basic**

Visual Basic — это мощная среда разработки приложений, по которой написаны огромные книги (см., например, [Excel 2013: профессиональное программирование на VBA](http://www.dialektika.com/books/978-5-8459-1877-2.html)). Вот лишь некоторые из инструментов:

* В процессе написания кода программа Excel может предложить вам выбирать ключевые слова из специального раскрывающегося списка возможных вариантов. Эта опция, называемая *автозаполнением*, позволяет создавать код быстрее, исключая ошибки ввода.
* Чтобы узнать смысл любого ключевого слова, наведите на него указатель мыши и нажмите клавишу F1. Вам может понадобиться установочный DVD, поскольку файл справки VBA не устанавливается с Excel 2013 по умолчанию.
* Excel проверяет каждую строку кода после ввода. Строки с ошибками выделяются красным цветом. Комментарии выделяются зеленым цветом. Можно добавить комментарий, указав в начале строки апостроф. Используйте комментарии, чтобы описать назначение каждого раздела кода.
* Несмотря на наличие описанной выше системы, осуществляющей проверку ошибок ввода, Excel может столкнуться с ошибкой на этапе выполнения кода. Если это произошло, щелкните на кнопке *Debug (Отладка)*. Строка, вызвавшая ошибку, будет выделена желтым цветом. Наведите указатель на любую переменную, чтобы увидеть ее текущее значение.
* В режиме отладки используйте меню *Debug* для пошагового выполнения кода. Можете переключаться между Excel и VBA, чтобы видеть эффект выполнения строки кода на рабочем листе.
* Другими эффективными средствами отладки являются точки прерывания, окна *Watch (Контрольное значение), Object Browser (Обозреватель объектов) и Immediate (Отладка)*. Информацию о них можно найти в справочной системе Excel.

**Средство записи макросов**

Excel предлагает средство записи макросов, эффективность которого составляет лишь 90%. К сожалению, остальные 10% приходятся на долю разочарованных пользователей. Макрос, создаваемый с помощью средства записи макросов, сможет обрабатывать лишь данные, определенные в жестко заданных ячейках. Это весьма эффективно, если обрабатываемая база данных занимает постоянный диапазон ячеек, например, A1:L87601. Но если диапазон данных изменится, созданный ранее макрос окажется неработоспособным. Это один из недостатков средства записи макросов. Было бы намного лучше, если бы средство записи макросов Excel могло идентифицировать обрабатываемый диапазон с данными, например, после нажатия клавиши End. На практике профессионалы используют средство записи макросов для получения базового кода, который в дальнейшем приходится очищать и улучшать.

**Объектноориентированный код**

VBA — это объектноориентированный язык. Большинство строк VBA-кода вводится согласно синтаксису *Существительное.Глагол*, который в VBA трактуется как *Объект.Метод*. Объектами могут выступать рабочие книги, рабочие листы, ячейки или диапазоны ячеек. К методам могут относиться операции в Excel, такие как *.Сору*, *.Paste*, *.PasteSpecial*.

Многие методы содержат «наречия» — параметры, применяемые для настройки выполняемых операций. Если вы видите конструкцию со знаком присваивания, то знайте, что она описывает работу данного метода.

Еще одна разновидность операций — это присваивание значений характеристикам объекта. В VBA эти характеристики называются свойствами. Например, когда вы вводите выражение ActiveCell.Font.Colorlndex = 3, то указываете окрасить текст активной ячейки в красный цвет. Отметим, что для управления свойствами используется только знак равенства =, а не :=.

**Профессиональные хитрости**

Чтобы создавать эффективный VBAкод, следует освоить несколько простых методик.

**Написание кода для обработки диапазонов любых размеров.** Средство записи макросов строго определяет диапазон управляемых данных, например, А1:L87601.Такой код хорошо обрабатывает текущий набор данных, но он может быть неприменим к новым наборам данных. Поэтому целесообразно создавать код, способный обрабатывать диапазоны данных различных размеров.

Для ссылки на ячейку средство записи макросов использует синтаксис Range("Н12"). Однако есть более эффективный способ создания ссылки, к примеру, на ячейку в строке 12 столбца 8, заключающийся в применении синтаксиса Cells(12,8). Точно так же средство записи макросов будет ссылаться на прямоугольный диапазон ячеек с помощью синтаксиса Range("A1:L87601"). Однако для большей гибкости следует применить метод Cells() для указания верхнего левого угла диапазона, а затем метод Resize() — для определения количества строк и столбцов в диапазоне. Альтернативный синтаксис описания предыдущего диапазона имеет следующий вид: Cells(1,1). Resize(87601,12). Этот способ обеспечивает большую гибкость, поскольку появляется возможность заменить любое число переменной.

В пользовательском интерфейсе Excel можно использовать клавишу End для перехода в конец диапазона данных. Многие полагают, что для определения последней строки диапазона, начинающегося в ячейке А1, следует нажать комбинацию клавиш End+🡓. Не стоит рассчитывать на такой метод, поскольку данные, передаваемые из другой системы, могут оказаться неполными. Если программа последние пять лет импортирует 500 тыс. строк из устаревшей компьютерной системы, в один прекрасный момент в наборе данных может появиться нулевое значение. Это приведет к образованию пустой ячейки в середине набора данных. Именно в этой ячейке перестанет выполняться выражение Range("A1").End(xlDown), что приведет к потере остальных данных. Существование пустой ячейки может привести к тому, что в ежедневном отчете будут отсутствовать тысячи строк данных, что подорвет доверие к отчету и к вам лично. Выполните дополнительную операцию, чтобы перейти к последней строке рабочего листа, что позволит резко уменьшить вероятность возникновения ошибок.

Если поместить курсор в последнюю строку рабочего листа и нажать комбинацию клавиш End+🡑, то курсор перейдет к последней строке с данными. В VBA эквивалентный код имеет следующий вид:

Range("А1048576").End(xlUp).Select

Вам не нужно выбирать саму ячейку — требуется лишь определить номер последней строки. Следующий код указывает строку и сохраняет ее номер в переменной FinalRow:

FinalRow = Range("А1048576").End(xlUp).Row

В имени переменной FinalRow нет ничего особенного. Этой переменной можно присвоить другое имя, например, х, у и т.п. Но поскольку в VBA лучше использовать описательные названия переменных, то для описания итоговой строки рекомендуются такие имена, как FinalRow.

Программа Excel 2013 допускает использование 1 048 576 строк и 16 384 столбцов на листе. Файлы, сохраненные в режиме совместимости Excel 2003, поддерживают 65 536 строк и 256 столбцов. Чтобы рабочую книгу можно было открывать как в Excel 2003, так и в Excel 2013, предыдущий код можно изменить следующим образом:

FinalRow = Cells(Application.Rows.Count,1).End(xlUp).Row

Можно также найти последний столбец в наборе данных. Если вы точно уверены, что набор данных начинается со строки 1, то можете использовать клавишу End в комбинации с клавишей 🡐, чтобы перейти от ячейки XFD1 к последнему столбцу таблицы, в котором содержатся данные. Чтобы обеспечить выполнение кода в устаревших версиях Excel, можно использовать следующий код:

FinalCol = Cells(1, Application.Columns.Count).End(xlToLeft).Column

**Использование объектных переменных.** В популярных языках программирования переменная содержит одно значение. Чтобы присвоить значение 4 переменной х, можно использовать выражение х = 4.

Рассмотрим отдельную ячейку в Excel. Существует много свойств, описывающих ячейку. Ячейка может содержать числовое значение, для нее указываются размер и цвет шрифта, номер строки и столбца, определяются формула, примечание и т.п. В VBA можно создать «суперпеременную», которая будет содержать всю информацию о ячейке или даже целом объекте. Выражение х=Range("A1") присваивает переменной х текущее значение ячейки А1. Для создания объектной переменной применяется ключевое слово Set.

Set х = Range("A1")

Вы только что создали «суперпеременную», которая содержит все свойства ячейки. Вместо использования переменной с одним значением можно работать с переменной, в которой предоставляется доступ к значениям многих свойств, связанных с этой переменной. Можете использовать выражение х.Formula, чтобы увидеть формулу в ячейке А1, или х.Font.colorIndex, чтобы определить цвет ячейки.

В примерах, приводимых в этой главе, часто используется объектная переменная РТ. Она применяется в тех случаях, когда определяется ссылка на сводную таблицу. Если, например, в коде содержится ссылка ActiveSheet.PivotTables("PivotTablel"), можно воспользоваться объектной переменной РТ, чтобы уменьшить объем кода.

**Использование блока With/End With для уменьшения объема кода.** Работая со сводными таблицами, приходится часто изменять значения их отдельных параметров. Ниже приводится пример кода (подробнее он объясняется далее), каждая строка которого изменяет значение параметра сводной таблицы.

PT.NullString=0

РТ.RepeatAllLabels xlRepeatLabels

РТ.ColumnGrand=False

PT.RowGrand-False

PT.RowAxisLayout xlTabularRow

PT.TableStyle2="PivotStyleMedium10"

PT.TableStyleRowStripes=True

В процессе анализа этих строк кода инерпретатор VBA должен постоянно расшифровывать, что означает ссылка РТ. Чтобы ускорить выполнение кода, укажите ссылку РТ один раз. Для каждой строки кода, которая начинается точкой, подразумевается ссылка на объект, указанный с помощью ключевого слова With. Блок кода завершается выражением End With.

With РТ

.NullString=0

.RepeatAllLabels xlRepeatLabels

.ColumnGrand=False

.RowGrand-False

.RowAxisLayout xlTabularRow

.TableStyle2="PivotStyleMedium10"

.TableStyleRowStripes=True

End With

**Версии программы**

В каждой из трех последних версий Excel сводные таблицы непрерывно совершенствовались. И если вам приходится использовать код для одной из современных версии Excel, учтите, что этот код может оказаться неработоспособным в предыдущих версиях.

* В Excel 2013 появилась модель данных PowerPivot. Можно добавлять таблицы в модель данных, создавать связи и формировать сводные таблицы. Соответствующий код VBA не поддерживается в Excel 2010 либо более ранних версиях. Также в этой версии Excel появилась функция xlDistinctCount и временн*ы*е шкалы.
* В Excel 2010 появились срезы, команда *Повторять все подписи элементов*, именованные наборы данных и множество функций вычисления итогов: xlPercentOfParentColumn, xlPercentOfParentRow, xlPercentRunningTotal, xlRankAscending и xlRankDescending. Соответствующий код VBA не поддерживается в Excel 2007 или более ранних версиях.
* В Excel 2007 появилась функция преобразования ячеек в формулы (ConvertToFormulas), макет xlCompactRow, функция xlAtTop, используемая для отображения промежуточных итогов сверху, стили таблиц (TableStyles) и функция SortUsingCustomLists (сортировка с помощью пользовательских списков). Макросы, в которых используется этот код, не поддерживаются в предыдущих версиях Excel.

**Создание сводной таблицы средствами VBA**

Если ваша цель проанализировать данные с помощью сводной таблицы, не следует создавать последнюю с использованием VBA для. Этот раздел полезен в том случае, если вы пишете код VBA для анализа данных, а сводная таблица является промежуточным этапом этого анализа. Можете применить сводную таблицу для подведения итогов по набору данных, а затем использовать эти итоги в других расчетах.

Начиная с версии Excel 2007 пользовательский интерфейс сводных таблиц претерпел значительные изменения, но VBA-код, который используется для управления им, остается прежним. Разработчики сделали правильный выбор, иначе миллионы приложений, написанных на VBA, попросту перестали бы выполняться в Excel 2007. Несмотря на изменение названий областей сводной таблицы в Excel, в VBA они продолжают носить старые имена: "область страницы" (Page Fields), "область столбцов" (Column Fields), "область строк" (Row Fields) и "область данных" (Data Fields).

В Excel 2000 и более поздних версиях, прежде чем создавать сводную таблицу, нужно сформировать объект кеша сводной таблицы, применяемый для описания области ввода данных. Здесь и далее я не буду приводить VBA-код, а отсылаю к соответствующему фрагменту кода, хранящемуся в Excel-файле, приложенному к этой заметке, в Module1. Итак, для создания сводной таблицы перейдите на лист Data Excel-файла и запустите макрос (процедуру) Sub CreatePivot1(). Эта процедура включает 4 блока: определения, удаление ранее созданных сводных таблиц, определение области ввода и настройка кэша, создание сводной таблицы на основе кэша. После выполнения этой процедуры вы получите пустую сводную таблицу (рис. 4).

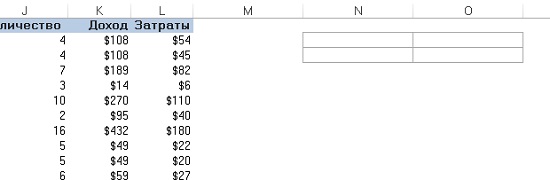


Рис. 4. Процедура Sub CreatePivot1() создаст пустую сводную таблицу из четырех ячеек

Чтобы проследить за дальнейшим написанием кода перейдите к Sub CreatePivot(). Если в области списка полей сводной таблицы был установлен флажок *Отложить обновление макета*, то Excel не станет вновь вычислять сводную таблицу после перетаскивания в таблицу каждого поля. По умолчанию программа вычисляет сводную таблицу после выполнения каждого этапа построения макета таблицы. Таким образом, сводная таблица будет вычисляться много раз, пока не будет получен итоговый результат. Чтобы ускорить выполнение кода, временно отключите вычисление сводной таблицы с помощью свойства ManualUpdate.

В методе .AddFields указывается одно или несколько полей, которые должны находиться в области строк, столбцов или страниц сводной таблицы. Параметр RowFields позволяет определять поля, которые добавлены в область заголовков строк списка полей сводной таблицы. Параметр ColumnFields соответствует области заголовков столбцов, а параметр PageFields — области фильтров отчета. В нашем программном коде заданы два поля в области заголовков строк и одно поле в области заголовков столбцов. (Если в одну из областей сводной таблицы было добавлено единственное поле, например, поле Регион в области столбцов сводной таблицы, имя поля заключается в кавычки. Если добавляется несколько полей, список полей включают в состав функции массива.)

Несмотря на то что поля строк, столбцов и страниц в сводной таблице могут обрабатываться с помощью метода .AddFields, для добавления полей в область данных лучше воспользоваться кодом, описанным в следующем разделе.

**Добавление полей в область данных.** В процессе добавления полей в область данных сводной таблицы пользователю доступен ряд настроек. Не полагайтесь на Excel, а воспользуйтесь возможностями, предлагаемыми этими настройками. Предположим, создается отчет сводной таблицы, демонстрирующий распределение дохода. В процессе составления сводной таблицы нужно просуммировать величину дохода. Если явно не указать выполняемый вид вычисления, Excel просматривает всю электронную таблицу, на основе которой формируется сводная таблица. Если столбцы данных, включающие доход, на 100% числовые, Excel выполняет корректное суммирование. Если хотя бы одна из ячеек пустая или содержит текст, Excel переходит от суммирования к подсчету записей, содержащих сведения о доходе. В итоге получаем противоречивые результаты.

Из-за подобной путаницы следует отказаться от использования аргумента DataFields для метода AddFields. Вместо этого измените свойство поля на xlDataField, а в качестве функции укажите xlSum. В процессе настройки поля данных можно изменить ряд свойств, находящихся в одном и том же блоке With...End With. Свойство Position используется в тех случаях, когда в область данных нужно включить несколько полей. Этому свойству присваивается значение 1 (для первого поля), значение 2 (для второго поля) и т.д.

По умолчанию Excel переименовывает перемещенное в область данных поле Доход, в результате чего последнее получает немного странное имя Сумма по полю Доход. Если хотите присвоить этому полю более осмысленное имя, воспользуйтесь свойством .Name. Обратите внимание на то, что в качестве имени поля, перемещенного в область данных, вместо недопустимого названия "Доход" можно использовать имя "Доход " (с пробелом в конце). Вовсе не обязательно для полей, находящихся в области данных, указывать числовой формат, хотя для улучшения читаемости сводной таблицы можно задать формат принудительно, например, "# ##0".

**Форматирование сводной таблицы.** В версии Excel 2007 компания Microsoft представила пользователям сжатый макет. Этот макет применяется и в Excel 2013 наравне с двумя другими макетами. По умолчанию для сводной таблицы выбирается табличный макет. Это наиболее подходящий макет для представления данных. Чтобы принудительно выбрать табличный макет для сводной таблицы, воспользуйтесь следующей строкой кода: PT.RowAxisLayout xlTabularRow

В случае выбора табличного макета каждое поле, относящееся к области строк, находится в отдельном столбце. Промежуточные итоги всегда отображаются в нижней части каждой группы. Эти особенности макета приводят к тому, что сводная таблица будет наиболее громоздкой и в то же время самой подходящей для выполнения дальнейшего анализа. Зачастую настройками интерфейса пользователя Excel по умолчанию определяется сжатый макет. При выборе этого макета несколько полей столбцов сворачиваются в единственный столбец, находящийся в левой части сводной таблицы. Для выбора этого макета воспользуйтесь следующим кодом: PT.RowAxisLayout xlCompactRow

Единственное ограничение табличного макета заключается в невозможности отображения промежуточных итогов в верхней части каждой группы. Если вы нуждаетесь в этой возможности, выберите макет в виде структуры и выведите итоги в верхней части группы с помощью кода:

PT.RowAxisLayout xlOutlineRow  
PT.SubtotalLoaction xlAtTop

Сводная таблица наследует заданные по умолчанию настройки табличного стиля. Если хотите изменить формат сводной таблицы, укажите требуемый стиль явно. Например, задайте чередование строк и средний оттенок, выбираемый для заливки сводной таблицы:

PT.ShowTableStyleRowStripes = True  
PT.TableStyle2 = "PivotStyleMedium10"

Итак, мы присвоили значения всем настройкам, требуемым для корректного генерирования сводной таблицы с помощью VBA. Теперь осталось присвоить свойству ManualUpdate значение False, и Excel пересчитает и отобразит сводную таблицу.

Сводная таблица, полученная после выполнения процедуры Sub CreatePivot() показана на рис. 5.

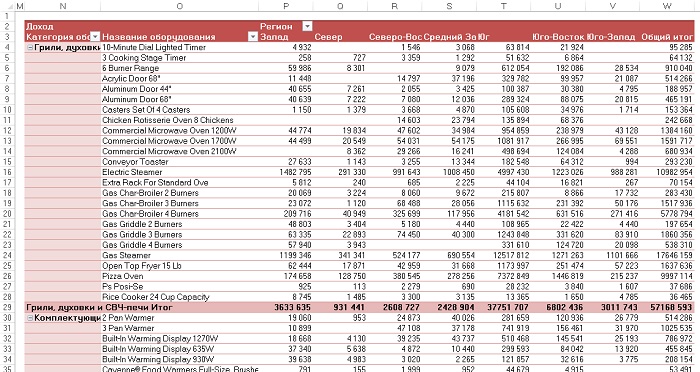


Рис. 5. Эта сводная таблица была получена менее чем за секунду в результате выполнения 50 строк кода (листинг кода см. в Excel-файле в Module1)

**Ограничения, присущие сводным таблицам**

Как и при создании сводных таблиц с помощью пользовательского интерфейса, в процессе формирования сводных таблиц с помощью кода VBA следует учитывать ряд ограничений.

**Заполнение пустых ячеек в области данных.** Многих пользователей раздражает наличие пустых ячеек в области данных сводной таблицы. Соответствующий пример приведен на рис. 5, где в регионе Север отсутствуют продажи оборудования под названием 10-Minute Dial Lighted Timer. Соответствующая ячейка пуста; уместнее было бы отобразить в ней значение 0.

Для отображения нулей в пустых ячейках можно воспользоваться настройкой *Для пустых ячеек отображать*, находящейся в окне *Параметры сводной таблицы*. Для вывода этого окна на экран выберите контекстную вкладку *Параметры* и щелкните на кнопке *Параметры*. Эту же задачу решает следующая строка кода: PT.NullString = "0"

Обратите внимание на то, что средство записи макросов Excel всегда заключает нули в кавычки. Независимо от того, укажете ли вы "0" либо 0, пустые ячейки в области данных сводной таблицы будут всегда заполнены числовыми нулями.

**Заполнение пустых ячеек в области строк.** Еще в Excel 2010 появилась давно ожидаемая настройка, позволяющая заполнять пустые ячейки в расположенных слева столбцах сводной таблицы (путем повторения названий строк). Эта проблема появляется в случае, если в области строк отображаются два или большее количество полей. По умолчанию вместо повторения надписей, таких как Грили, духовки и СВЧ-печи, в левой колонке в этом случае остаются пустые ячейки (см. рис. 5). Помимо команд интерфейса, для решения этой проблемы в Excel 2013 можно воспользоваться следующим кодом: РТ.RepeatAllLabels xlRepeatLabels

**Невозможность перемещения или изменения части отчета.** Несмотря на богатую функциональность сводных таблиц, они имеют некоторые ограничения. Вы не сможете перемещать или изменять только часть сводной таблицы. Попытайтесь, к примеру, запустить макрос, который удаляет столбец W сводной таблицы. Макрос тут же известит вас о том, что возникла ошибка 1004. Чтобы обойти указанное ограничение, можно воспользоваться одной из следующих двух стратегий. Первая стратегия заключается в том, чтобы найти эквивалентные команды в интерфейсе Excel, используемом для создания и изменения сводных таблиц. В частности, с помощью команд Excel можно решить следующие задачи:

* удаление столбца общих итогов;
* удаление строки общих итогов;
* добавление пустых строк между разделами;
* сокрытие промежуточных итогов для внешних полей строк.

Вторая стратегия состоит в преобразовании сводной таблицы в значения. После этого можно выполнять операции вставки, вырезания и очистки. Обе стратегии рассматриваются ниже.

**Управление итогами.** Изначально в состав любой сводной таблицы входят одна строка и один столбец общих итогов. С помощью команд интерфейса Excel 2013 можно скрыть один либо оба этих элемента. Для удаления столбца общих итогов, отображаемого в правой части сводной таблицы, воспользуйтесь кодом: РТ.ColumnGrand = False. Для удаления строки общих итогов применяется код: PT.RowGrand = False.

Удаление строк промежуточных итогов — довольно сложный процесс. Необходимость в выполнении подобной операции может возникнуть в том случае, когда в области строк находится несколько полей. В этом случае Excel автоматически отображает промежуточные итоги для внешних полей строки.

Знаете ли вы, что сводная таблица может включать несколько строк промежуточных итогов? Как правило, эти возможности не используются. Если же нужно ими воспользоваться, встаньте на любую ячейку в области строк, например, N10 на рис. 5, кликните правой кнопкой мыши, выберите опцию *Параметры поля…* перейдите на вкладку *Промежуточные итоги*, и выберите одну или несколько функций, применяемых для вычисления промежуточных итогов (рис. 6): Сумма, Количество, Среднее, Максимум, Минимум и др.

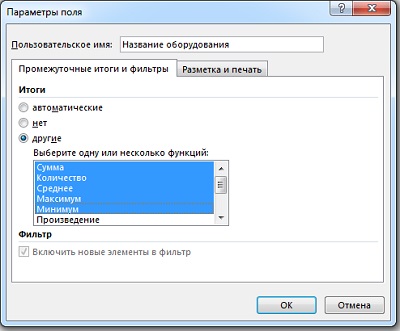


Рис. 6. Несмотря на то что возможность выбора различных функций формирования промежуточных итогов применяется редко, сам факт ее наличия серьезно усложняет VBA-код, применяемый для сокрытия промежуточных итогов

Для сокрытия промежуточных итогов, отображаемых для поля, свойству Subtotals следует присвоить массив, состоящий из 12 значений False. Первое значение False отключает отображение автоматических промежуточных итогов, второе значение False — отображение промежуточных итогов, сформированных с помощью функции *Сумма*, третье значение False отключает отображение промежуточных итогов, сформированных с помощью функции *Количество*, и т.д.:

РТ.PivotFields("Категория оборудования").Subtotals = Array( \_  
False, False, False, False, False, False, False, False, False, False, False, False)

Можно также вывести первый промежуточный итог. При этом скрываются остальные 11 промежуточных итогов. Затем можно скрыть первый промежуточный итог, чтобы гарантировать отключение всех промежуточных итогов:

PT.PivotFields("Категория оборудования").Subtotals(1) = True  
PT.PivotFields("Категория оборудования").Subtotals(1) = False

**Определение размера готовой сводной таблицы для ее преобразования в значения.** Если планируется преобразовать сводную таблицу в значения, потребуется скопировать всю сводную таблицу. Но в этом случае довольно сложно заранее предсказать размер сводной таблицы. Если вы каждый день выводите отчет по продажам, то можете иметь (или не иметь!) в наличии данные по объемам продаж, к примеру, для западного региона. Тогда сводная таблица может включать либо семь, либо шесть столбцов.

Для установки ссылок на сводную таблицу в Excel применяются два свойства диапазонов. Специальное свойство .TableRange2 включает все строки сводной таблицы, в том числе все раскрывающиеся списки полей страниц, которые отображаются в верхней части сводной таблицы. Область действия свойства .TableRangel находится ниже области расположения полей фильтра. Это свойство зачастую включает такие необязательные строки, как *Сумма по полю Доход*, которые отображаются в верхней части сводной таблицы.

Если нужно преобразовать сводную таблицу в значения, не перемещая при этом таблицу в новое местоположение, воспользуйтесь кодом:

РТ.TableRange2.Сору  
РТ.TableRange2.PasteSpecial xlPasteValues

Если в новое местоположение нужно скопировать лишь область данных сводной таблицы, для смещения на одну строку ниже места, выбираемого с помощью свойства .TableRange2, выбирается свойство .Offset: РТ.TableRange2.Offset(1,0).Copy

Эта ссылка копирует область данных и дополнительно еще одну строку для заголовков. Обратите внимание на рис. 7, где использование свойства .Offset без свойства .Resize приводит к копированию дополнительной строки. Но поскольку эта строка всегда пуста, нет необходимости в применении свойства .Resize для копирования большего числа дополнительных строк.

В коде не только копируется свойство РТ.TableRange2, но и выполняется метод PasteSpecial ячейки, расположенной на шесть строк ниже сводной таблицы. На этом этапе рабочий лист должен выглядеть так, как показано на рис. 7. Таблица в ячейке N2 представляет реальную сводную таблицу, а таблица в ячейке N57 содержит всего лишь скопированные итоговые сведения. В дальнейшем можете полностью очистить сводную таблицу, применив ко всей таблице метод Clear. Перед дальнейшей настройкой новой таблицы следует удалить кеш старой сводной таблицы из памяти, назначив свойству PTCache значение Nothing. Описанные действия собраны в процедуре Sub UsePivotToCreateValues (листинг см. в Module1).

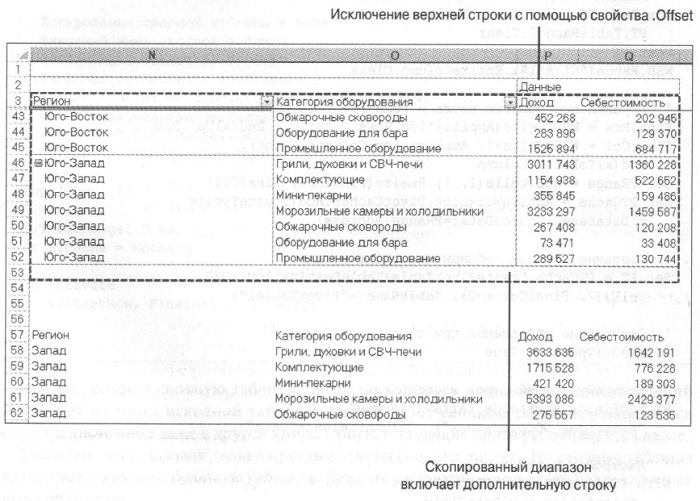


Рис. 7. Промежуточный результат выполнения макроса. Данные, находящиеся в нижней части таблицы, преобразованы в значения

**Отчет о доходах по рынкам сбыта и категориям оборудования**

Допустим перед вами поставлена задача создать отчет как на рис. 8. Необходимо отобразить группировку по категориям оборудования, далее по рынкам сбыта (по убыванию дохода), годы показать в верхней части отчета, вывести промежуточные итоги, создать структуру.

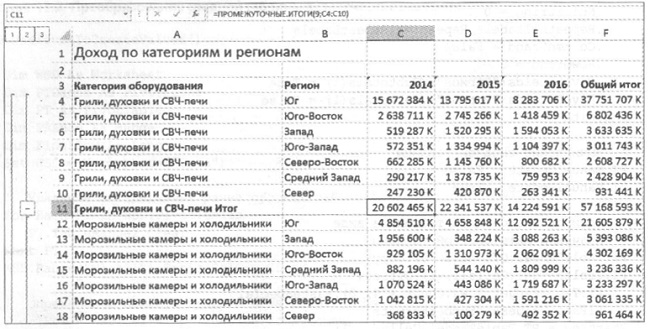


Рис. 8. Отчет начинается как сводная таблица, а завершается как обычный набор данных

Начать нужно с создания сводной таблицы (рис. 9.). Ниже приведен список задач, которые нужно решить при создании этой сводной таблицы:

* даты, представленные в исходном наборе данных, следует сгруппировать по годам;
* необходимо проконтролировать порядок сортировки в полях строк;
* следует заполнить пустые ячейки в сводной таблице, выбрать числовой формат и скрыть промежуточные итоги в полях категорий.

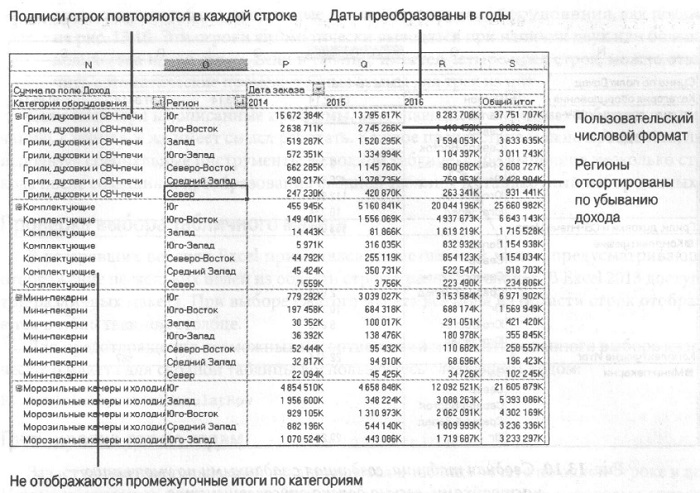


Рис. 9. Типичная просьба менеджера заключается в том, чтобы на основе исходных данных создать отчет о продажах оборудования

Чтобы создать такую сводную таблицу начните с кода:  
PT.AddFields RowFields:=Array("Категория оборудования","Регион"), ColumnFields:="Дата"  
' Настройка полей данных  
With РТ.PivotFields("Доход")

.Orientation xlDataField  
.Function = xlSum   
.Position = 1  
.NumberFormat = "# ##0"

End With

Будет создана сводная таблица, показанная на рис. 10.

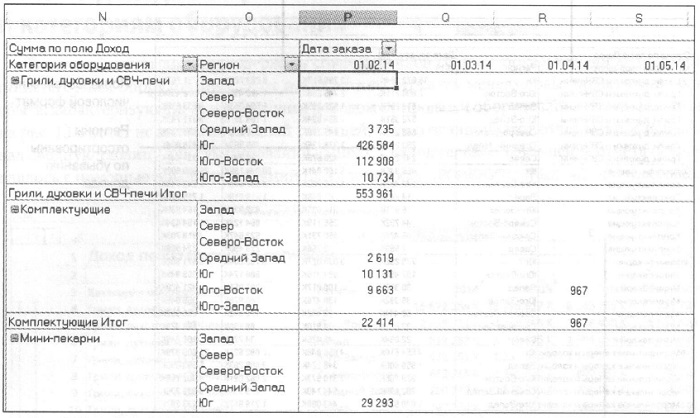


Рис. 10. Сводная таблица, созданная с заданными по умолчанию настройками, весьма далека от совершенства

Ниже описаны недостатки, присущие большинству стандартных сводных таблиц:

* Режим структуры очень неудобен. На рис. 10 значение *Грили, духовки и ВЧ-печи* появляется в столбце *Категория оборудования* только один раз, после чего выводится 6 пустых ячеек. К счастью, в Excel 2013 появился метод RepeatAllLabels, который позволит устранить эту проблему. Если вы намерены повторно использовать данные, повторите заголовки строк в каждой строке.
* Поскольку в исходном наборе данных содержатся сведения о продажах по дням, созданная по умолчанию сводная таблица будет включать более тысячи столбцов. Вряд ли целесообразно анализировать подобный отчет. Следует сгруппировать данные о ежедневных продажах по годам, воспользовавшись возможностями сводных таблиц.
* Отчет содержит пустые ячейки вместо нулевых значений. На рис. 10 ниже заголовка *Грили, духовки и СВЧ-печи* отображается диапазон пустых значений. Вместо пустых значений следует отображать нули.
* Названия нескольких столбцов и полей не всегда удовлетворяют запросам пользователей. Например, заголовок *Сумма по полю Доход* раздражает большинство пользователей.
* Алфавитный порядок сортировки не всегда приемлем. Так, менеджеры по продажам часто настаивают на том, чтобы в верхней части списка были помещены названия наиболее выгодных рынков сбыта. Лучше отсортировать отчет в порядке убывания объемов продаж.
* Оформление отчета, выполненное с помощью рамок, оставляет желать лучшего. Программа создает столько рамок, что отчет иногда трудно понять.
* В сводных таблицах отсутствует разумная логика использования разрывов страниц. Например, если требуется создать отчет для каждого менеджера по отдельности, не существует быстрого способа вывода данных о каждой единице оборудования на новой странице.
* Проблема нелогичности вставки разрывов страниц вызывает острое желание отказаться от строк с промежуточными суммами сводных таблиц и прибегнуть к функции ПРОМЕЖУТОЧНЫЕ.ИТОГИ для добавления промежуточных итогов с собственными разрывами страниц. В таком случае отключите строки промежуточных сумм сводных таблиц, выводимые в поле *Категория оборудования*, как показано на рис. 10. Эти строки автоматически выводятся при наличии двух или большего количества полей строк. Если в таблице имеется четыре поля строк, можно отключить автоматические промежуточные суммы для трех из них.

Даже несмотря на описанные проблемы, встречающиеся в сводных таблицах по умолчанию, с ними все же имеет смысл работать. Каждое препятствие можно преодолеть, либо используя специальные инструменты в сводной таблице, либо выполнив несколько строк кода после создания и преобразования сводной таблицы в стандартный набор данных.

**Проверка выбора табличного макета.** В устаревших версиях Excel применялся единственный макет, предусматривающий отображение нескольких полей из области строк в разных столбцах. В Excel 2013 доступны три различных макета. При выборе сжатого макета все поля из области строк отображаются в единственном столбце. Для предотвращения возможных несоответствий и гарантированного выбора классического макета для сводной таблицы воспользуйтесь следующим кодом: PT.RowAxisLayout slTabularRow

**Группировка дат по годам.** В Excel 2013 можно обратиться к возможностям группирования в Excel, выбрав произвольный уровень группировки (по месяцам, кварталам и т.д.). Группировка по дате в VBA выполняется довольно нестандартно. Метод .Group может применяться только к единственной ячейке сводной таблицы, а эта ячейка должна включать дату либо надпись поля *Дата*. Как показано на рис. 10, можно выбрать либо заголовок *Дата заказа*, отображенный в ячейке Р2, либо одну из дат в диапазоне РЗ:АРМЗ. Первый метод довольно ненадежный, особенно если сводная таблица будет создаваться в новом столбце. Два других способа более надежны.

Если вы не собираетесь использовать разное число полей строк, можете обозначить заголовок *Дата заказа*, находящийся в ячейке на пересечении строки 1 и столбца 3, как TableRange2. Эта ячейка может быть выделена с помощью следующей строки кода: РТ.TableRange2.Cells(1,3).Select. Если количество полей строк будет изменяться, добавьте комментарий, в котором упоминается, что для изменения числа полей строк следует заменить тройку другим числом.

Еще один способ выделения ячейки основан на использовании свойства LabelRange для поля *Дата заказа*. Следующий код всегда выбирает ячейку, содержащую заголовок поля *Дата заказа*: РТ.PivotFields("Дата заказа").LabelRange.Select.

Для группировки ежедневных дат по годам следует определить сводную таблицу, в качестве поля строки которой используется поле Дата. Для гарантированного создания сводной таблицы отключите свойство ManualCalculation. Для поиска подписи даты воспользуйтесь свойством LabelRange.

К ячейке, содержащей подпись даты, примените метод .Group. Для аргумента Periods этого метода следует определить массив, включающий семь булевых значений. Эти семь значений соответствуют секундам, минутам, часам, дням, месяцам, кварталам и годам. Например, для группировки по годам применяется следующий код.

PT.PivotFields("Дата заказа").LabelRange.Group \_

Periods:=(False, False, False, False, False, False, False, True)

После выполнения группировки по годам поле по-прежнему называется *Дата заказа*. В этом заключается отличие от группировки по нескольким полям.

Для группировки по месяцам, кварталам и годам используется следующий код:

PT.PivotFields("Дата заказа").LabelRange.Group \_

Periods:=(False, False, False, False, False, True, True, True)

После выполнения группировки по месяцам, кварталам и годам в поле *Дата заказа* отображаются месяцы. В сводной таблице появляется два новых виртуальных поля: Кварталы и Годы.

Для группировки по неделям выбирается период *День*, а затем используется аргумент *By* для группировки по периодам, состоящим из семи дней:

PT.PivotFields("Дата заказа").LabelRange.Group Ву:=7 \_

Periods: = (False, False, False, True, False, False, False, True)

Поскольку показанная на рис. 10 сводная таблица будет группироваться по годам, используется следующий код:

PT.PivotFields("Дата заказа").LabelRange.Group \_

Periods: =(False, False, False, False, False, False, False, True)

На рис. 11 показана сводная таблица с датами продаж, сгруппированными по годам.



Рис. 11. Группировка по годам выполнена с помощью метода .Group

**Исключение пустых ячеек.** Наличие пустых ячеек в сводной таблице раздражает многих пользователей. Причем существуют две категории пустых ячеек, от которых хотят избавиться многие пользователи. Во-первых, пустые ячейки могут отображаться в области значений в случае отсутствия определенных записей. Например, как показано на рис. 11, в компании не выполнялись продажи оборудования для баров в 2014 году, поэтому ячейки Р44:Р51 пусты. Поскольку наличие пробелов раздражает многих пользователей, в подобных случаях лучше отображать нули. Во-вторых, пустые ячейки могут также отображаться в области строк при наличии нескольких полей строк. Например, как показано на рис. 11, в ячейке N44 отображается название *Оборудование для бара*, а ячейки N45:N50 пусты.

Для замены пробелов нулями в области значений можно воспользоваться кодом: PT.NullString = "0". Несмотря на то что в коде свойство принимает текстовый нуль, Excel помещает в пустые ячейки реальный числовой нуль. Для заполнения пустых ячеек в области строк в Excel 2013 используется код: PT.RepeatAllLabels xlRepeatLabels.

**Изменение сортировки с помощью метода AutoSort.** Пользовательский интерфейс Excel содержит функцию *Автосортировка*, которая позволяет выводить данные в порядке уменьшения дохода. В VBA-коде, выполняющем аналогичную задачу, для сортировки по полям региона и категории оборудования в порядке уменьшения дохода используется метод AutoSort:

РТ.PivotFielda("Регион").AutoSort Order:=xlDescending, ­Field:="Сумма по полю Доход"

РТ.PivotFielda("Категория оборудования"). \_

AutoSort Order:=xlDescending, Field:="Сумма по полю Доход"

**Изменение числового формата, заданного по умолчанию.** Для числовых значений, отображаемых в области значений сводной таблицы, следует использовать подходящий числовой формат. Для отображения целых значений и пробелом в качестве разделителя разрядов используется код: PT.PivotFields("Сумма по полю Доход").NumberFormat = "# ##0"

Некоторые компании обслуживают клиентов, которые совершают закупки на тысячи или миллионы долларов. Для того чтобы вывести числа в тысячах, используйте пробел в числовом формате и добавьте букву К для вывода чисел в тысячах: РТ.PivotFields("Сумма по полю Доход").Number Format = "# ##0˽К". Чтобы вывести миллионы, воспользуйтесь форматом:  
"# ##0,0˽˽\М".

**Исключение промежуточных итогов в нескольких полях строк.** Сначала присвойте свойству Subtotals(1) значение True. Это приведет к отмене отображения десяти других возможных промежуточных итогов. Затем можно отключить отображение первого промежуточного итога, чтобы гарантировать удаление всех промежуточных итогов:

PT.PivotFields("Категория оборудования").Subtotals(1) = True

PT.PivotFields("Категория оборудования").Subtotals(1) = False

Чтобы удалить строку общих итогов, воспользуйтесь следующей строкой кода: PT.ColumnGrand = False.

На рис. 12 показана сводная таблица с заполненными пустыми ячейками в области строк и значений, с выводимыми в тысячах величинами и полями *Категория оборудования* и *Регион*, отсортированными по убыванию, с удаленными промежуточными итогами.

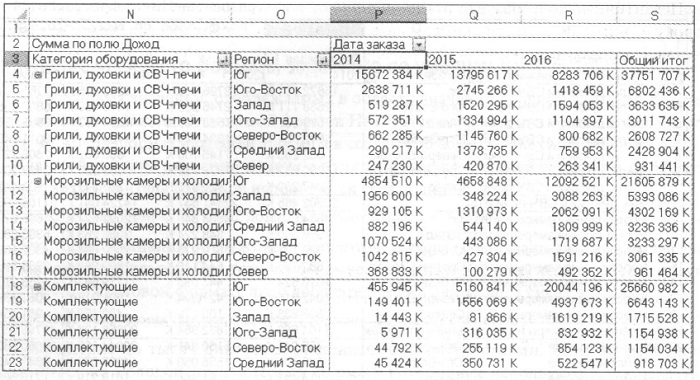


Рис. 12. Пустые ячейки заполнены, данные отсортированы, промежуточных итоги удалены

**Копирование завершенной сводной таблицы в виде значений в новую книгу.** Если вы планируете повторно использовать полученные в сводной таблице результаты, преобразуйте таблицу в значения. В этом разделе описан порядок копирования сводной таблицы в новую рабочую книгу. Чтобы повысить переносимость кода, присвойте объектные переменные исходной рабочей книге, новой рабочей книге и первому листу в новой рабочей книге. В верхнюю часть кода добавьте следующие инструкции:

Set WSD = Worksheets("Data")

Dim WSR As Worksheet

Dim WBO As Workbook

Dim WBN As Workbook

Set WBO = ActiveWorkbook

После успешного создания сводной таблицы сформируйте рабочую книгу Report.

' Создание новой пустой книги с одним листом

Set WBN = Workbooks.Add(xlWBATWorksheet)

Set WSR = WBN.Worksheets(1)

WSR.Name = "Отчет"

' Настройка заголовка отчета

With WSR.[A1]

.Value = "Доход по категориям оборудования и регионам"

.Style = "Title"

End With

Осталось внести в сводную таблицу несколько завершающих штрихов: убрать лишние рамки и изменить неуклюжие названия в первой строке сводной таблицы — *Сумма по полю Доход* и *Дата*. Для решения перечисленных проблем можно исключить первую строку диапазона РТ. TableRange2 из метода .Сору, а затем воспользоваться методом PasteSpecial (xlPasteValuesAndNumberFormats) для копирования данных на лист отчета.

В рассматриваемом примере свойство .TableRange2 включает лишь одну удаляемую строку (строка 2), которая показана на рис. 12. Если вы имеете дело с более сложной сводной таблицей, которая включает несколько полей столбцов и/или несколько полей страниц, придется удалять более чем одну строку отчета. Для этого удобно выполнить макрос, просмотреть полученный результат и определить, сколько строк нужно удалить. Для того чтобы предотвратить копирование строк в отчет, используйте свойство Offset. Скопируйте свойство TableRange2 и выполните смещение на одну строку.

Педантичные пользователи отметят, что этот код копирует дополнительную пустую строку из нижней части сводной таблицы, но это не играет роли, поскольку она все равно пустая. После копирования удалите исходную сводную таблицу и очистите ее кеш:

' Копирование данных сводной таблицы в строку 3 листа Отчет

' Использование функции Offset для удаление строки заголовка из сводной таблицы

PT.TableRange2.Offset(1, 0).Copy

WSR.[A3].PasteSpecial Paste:=xlPasteValuesAndNumberFormats

PT.TableRange2.Clear

Set PTCache = Nothing

Заметьте, что вы использовали свойство PasteSpecial для вставки лишь значений и числовых форматов. Таким образом, вы избавились как от границ, так и от структуры сводной таблицы. Можно использовать вариант вставки *Без рамок* *(All Except Borders)*, но тогда данные останутся в сводной таблице, и вы не сможете вставлять новые строки в середину набора данных.

**Заключительное форматирование.** Последние этапы создания отчета сводятся к базовым операциям форматирования и добавления промежуточных итогов. Выделите полужирным шрифтом заголовки в строке 3. Настройте строки 1–3 так, чтобы вверху каждой страницы выводились три строки.

' Базовое форматирование

' Автонастройка столбцов, выделение полужирным заголовков, выравнивание справа

Range("A3").EntireRow.Style = "Заголовок 4"

Range("A3").CurrentRegion.Columns.AutoFit

Range("A3").EntireRow.HorizontalAlignment = xlRight

Range("A3:B3").HorizontalAlignment = xlLeft

' Повторение строк 1-3 сверху каждой страницы

WSR.PageSetup.PrintTitleRows = "$1:$3"

**Финальное оформление.** На вкладке *Данные* доступны средства формирования промежуточных итогов. На рис. 13 показано диалоговое окно *Промежуточные итоги.* Обратите внимание на переключатель *Конец страницы между группами*.

И наконец, после добавления новых промежуточных итогов в отчет нужно автоматически настроить ширину числовых столбцов.

Итоговый листинг кода см. в Excel-файле в Module1 процедура Sub CategoryRegionReport().

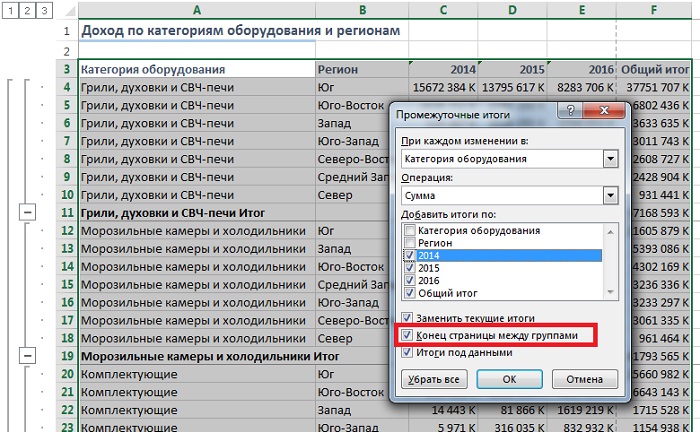


Рис. 13. С помощью автоматически формируемых промежуточных итогов можно добавлять разрыв страницы после названия каждой категории оборудования. Такой отчет будет удобен и понятен каждому менеджеру, поскольку он получит отчет, в котором будет указана информация о проданном им оборудовании

**Вычисления в сводной таблице**

В сводных таблицах можно размещать в области значений несколько полей и использовать 11 функций вычисления итогов (а не только сумму). В сводную таблицу можно также включать дополнительные вычисляемые поля либо вычисляемые элементы.

**Добавление нескольких полей в область значений сводной таблицы.** В область значений сводной таблицы можно включать несколько полей. Например, в одну и ту же сводную таблицу можно включать поля Доход, Затраты и Количество. Если в сводной таблице Excel 2013 два либо большее число полей данных, которые были созданы с помощью команд интерфейса Excel, поля в области значений расположены в столбцах. Если же для создания сводных таблиц используется VBA, поля значений размещены в строках. Листинг кода см. в Excel-файле, Module1, процедура Sub Figure14(). В результате создается довольно необычная сводная таблица (рис. 14).

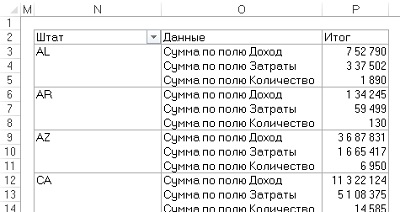


Рис. 14. С помощью VBA по умолчанию создается сводная таблица в формате, напоминающем формат сводных таблиц в Excel 2003

Для улучшения внешнего вида сводной таблицы переместите виртуальное поле *Данные* в область полей столбцов. Обратите внимание: *Данные* — это не столбец таблицы с оригинальными данными, а специальное имя, применяемое для определения расположения нескольких полей значений. Чтобы расположить несколько полей отчета в столбцах, воспользуйтесь кодом:  
РТ.AddFields RowFields := "Штат", ColumnFields := "Данные". После добавления виртуального поля *Данные* в область полей столбцов можно переходить к определению нескольких полей данных. Листинг кода см. в Excel-файле, Module1, процедура Sub Figure15 (). В результате выполнения этого кода формируется сводная таблица, показанная на рис. 15.

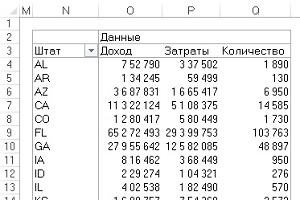


Рис. 15. Расположение нескольких полей значений в столбцах отчета сводной таблицы

**Функции вычисления итогов.** Во всех примерах сводных таблиц, приведенных в предыдущих разделах главы, использовалась функция вычисления итогов *Сумма*. Помимо этой функции, доступно еще 10 функций, выполняющих самые разные вычисления. Для выбора одной из этих функций задайте соответствующее значение для свойства .Function:

* xlAverage — среднее значение;
* xlCount — количество;
* xlCountNums — подсчет только числовых значений;
* xlMax — определение максимального значения;
* xlMin — определение минимального значения;
* xlProduct—умножение;
* xlStDev — стандартное отклонение по выборке;
* xlStDevP — стандартное отклонение по генеральной совокупности;
* xlSum — сумма;
* xlVar — дисперсия по выборки;
* xlVarP — дисперсия по генеральной совокупности.

Обратите внимание на то, что при добавлении поля в область значений сводной таблицы программа изменяет его название, добавляя название функции и слова «по полю». Например, название «Доход» превращается в название «Сумма по полю Доход», а название «Затраты» — в «Стандартное отклонение по выборке по полю Затраты». Если нужно ссылаться на эти поля в коде, используйте другие названия.

Внешний вид сводной таблицы можно существенно улучшить, если изменить свойство .Name для полей, находящихся в области значений. Если не хотите, чтобы в сводной таблице отображалось имя «Сумма по полю Доход», измените свойство поля .Name на «Итого по полю Доход». Это название более понятно, чем предыдущее. Помните о том, что в области значений невозможно использовать имя, которое совпадает с названием поля сводной таблицы. Например, в рассматриваемом примере нельзя использовать имя «Доход», хотя название «Доход » (с пробелом в конце) будет вполне допустимым.

При работе с текстовыми полями имеет смысл использовать лишь функцию *Количество*. Например, для подсчета количества записей в таблице можно добавить в нее текстовое поле, а затем воспользоваться функцией Количество.

В следующем примере вычисляются общий доход, количество записей, соответствующих заказчикам, и средняя величина дохода (рис. 16). Листинг кода см. в Excel-файле, Module1, процедура Sub Figure16().

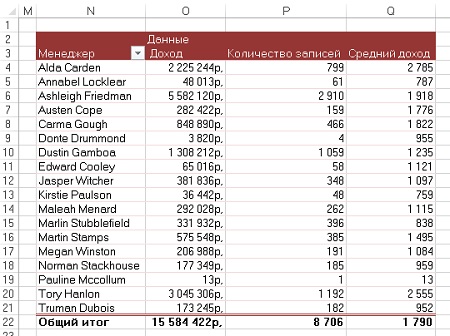


Рис. 16. Используются три функции в области значений:

**Вычисляемые поля данных.** Если ранее вы не сталкивались с такими полями, для начала изучите [Вычисляемые поля и вычисляемые элементы в Excel 2013](http://baguzin.ru/wp/?p=9911). Для настройки вычисляемого поля используйте метод Add по отношению к объекту CalculatedField. Укажите имя поля и формулу вычисления (ВП — валовая прибыль):

PT.CalculatedFields.Add Name:=" ВП ", Formula:="=Доход-Затраты", UseStandardFormula:=True

На основе одного вычисляемого поля могут создаваться другие вычисляемые поля. В нашем примере на основе поля ВП создается поле Рентабельность. С помощью свойства .Caption это поле переименовано на % от валовой прибыли. На рис. 17 показан результат вычислений. Листинг кода см. в Excel-файле, Module1, процедура Sub Figure17().

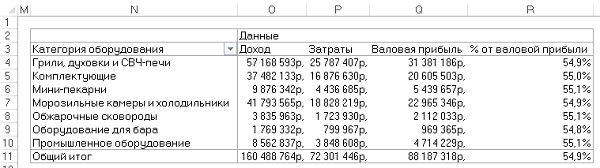


Рис. 17. Добавлено два вычисляемых поля

**Вычисляемые элементы.** При использовании вычисляемых элементов существует риск получения некорректных результатов в сводной таблице. Предположим, вам нужно создать отчет о продажах в девяти штатах. Для четырех штатов нужно отобразить промежуточные итоги. Вычисляемый элемент добавляет девятую позицию в столбец со штатами. И если новый элемент будет задействован в вычислениях, выполняемых в сводной таблице, вычисленный общий итог будет превышать действительное значение.

На рис. 18 показана сводная таблица, включающая сведения о девяти штатах. Общий доход по всем штатам составляет $10 млн. Листинг кода см. в Excel-файле, Module1, процедура Sub Figure18(). Если для четырех штатов (рис. 19) вычисляется промежуточный итог с помощью вычисляемого элемента, общий итог увеличивается до $15 млн. Это означает, что элементы, используемые при формировании вычисляемого элемента, учитываются дважды. Листинг кода см. в Excel-файле, Module1, процедура Sub Figure19().



Рис. 18. Итог по 9 штатам составляет $10М



Рис. 19. После добавления вычисляемого элемента значение общего итога не будет соответствовать действительности

Для обеспечения корректной работы вычисляемого элемента нужно скрыть записи, относящиеся к четырем штатам, которые входят в вычисляемый элемент (рис. 20). Листинг кода см. в Excel-файле, Module1, процедура Sub Figure20().

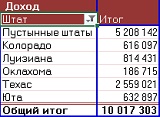


Рис. 20. Чтобы корректно воспользоваться вычисляемым элементом, скройте все входящие в него компоненты

В следующем разделе рассматривается лучшее решение, заключающееся в использовании текстовой группировки вместо вычисляемых элементов.

**Вычисления и группировка.** Если необходимо вычислить промежуточные итоги для отдельных регионов, лучше всего определить группы с помощью текстовой группировки. Если группировка выполняется по четырем штатам, Excel добавит новое поле в область строк сводной таблицы. Несмотря на некоторую сложность процесса группировки, игра стоит свеч, поскольку отчет приобретет профессиональный вид.

Для выполнения группировки по четырем штатам с помощью команд интерфейса Excel выделите ячейки, включающие эти штаты, выберите контекстную вкладку *Анализ*, входящую в группу контекстных вкладок *Работа со сводными таблицами*, и щелкните на кнопке *Группировка по выделенному*. В процессе группирования выполняются следующие действия:

* Все элементы группы перемещаются в область строк.
* Слева от поля, содержащего данные о штате, появляется новое поле. Если исходное поле называлось Штат, новое поле получает название Штат2.
* Отключается свойство отображения промежуточных итогов для поля Штат2, что не всегда приемлемо.
* Под именем Группа 1 добавляется промежуточный итог для выделенных элементов.
* Для всех элементов, не входящих в состав группы, в поле Штат2 появляется новый промежуточный итог с повторяющимся именем штата.

Если захотите с помощью кода VBA выделить ячейки, содержащие названия требуемых штатов, вас ожидают определенные трудности. Листинг кода см. в Excel-файле, Module1, процедура Sub Figure21(). В результате выполнения этого кода в сводной таблицы появляются новые виртуальные группы (рис. 21). К сожалению, мне не удалось полностью выполнить эту процедуру.



Рис. 21. Благодаря группировке по текстовым полям можно отобразить в отчете сведения, отсутствующие в исходной таблице

**Дополнительные функции вычисления итогов.** В раскрывающемся списке *Итоги по*, находящемся на контекстной вкладке *Анализ* в группе контекстных вкладок *Работа со сводными таблицами*, доступно 15 функций, применяемых для вычисления итогов. Эти функции позволяют вычислить количество позиций в заказе, нарастающий итог, долю и ряд других статистических показателей. Для изменения функции вычисления итогов воспользуйтесь свойством .Calculation для поля сводной таблицы.

Некоторые вычисления, такие как *% от суммы по столбцу* либо *% от суммы по строке*, не требуют дополнительных определений (не нужно указывать базовое поле). Другие вычисления, например, % от родительской суммы, требуют указания базового поля. Таким образом, свойство .Calculation может использоваться вместе со свойствами .BaseField и .BaseItem. В зависимости от выбранной функции вычисления итогов могут потребоваться либо оба свойства — .BaseField и .Baseltem, либо только свойство .Baseltem, либо ни одно из них. Примеры блоков, задающих функции вычисления итогов вы найдете в листинге процедуры Sub Figure22(). К сожалению, поскольку код этой процедуры является расширением кода предыдущей процедуры, у меня от также не запустился. Но если бы он мог быть выполнен, то получилась бы сводная, изображенная на рис. 22. Полный перечень значений свойства .Calculation см., например, [здесь](http://msdn.microsoft.com/en-us/office/ff836192(v=office.12).aspx).

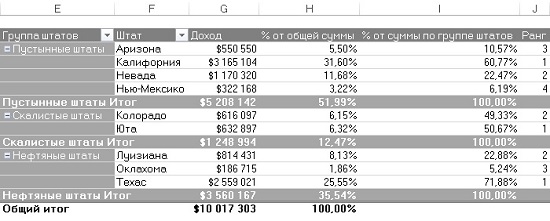


Рис. 22. Доход представлен в четырех вариантах: столбец G – без дополнительных итогов; H – % от суммы по столбцу, I – % от родительской суммы с базовым полем Группа штатов; J – сортировака от максимального к минимальному с базовым полем Штат

**Специальные методики управления сводными таблицами**

**Метод .AutoShow и создание обзоров рынков.** Менеджеры зачастую нуждаются в отчетах-обзорах, фокусирующихся на лучших рынках сбыта. Можно создать подобный отчет, позволяющий выбрать *n* верхних или нижних записей на основе любого поля данных.

При этом используется код VBA, созданный на основе метода .Autoshow:

' Отображение 5 лучших рынков сбыта

PT.PivotFields("Рынок сбыта").AutoShow Type:=xlAutomatic, Range:=xlTop, \_

Count:=5, field:="Суммарный доход"

При создании отчета с использованием метода .AutoShow часто имеет смысл скопировать данные, а затем вернуться к исходному отчету для получения итогов по всем рынкам сбыта. Эта идея реализована в процедуре Sub Top5Markets(), а получающийся отчет показан на рис. 23.



Рис. 23. Отчет *Лучшие 5 рынков* не является сводной таблицей

Отчет по первым пяти рынкам содержит два снимка моментального состояния сводной таблицы. После вызова метода .AutoShow для выборки пяти лучших рынков макрос возвращается к сводной таблице, отменяет действие метода .AutoShow и обрабатывает данные всех рынков для создания строки *Итого по компании*.

**Использование свойства ShowDetails для фильтрации набора записей.** Откройте любую сводную таблицу в окне Excel и дважды щелкните на любом числе в таблице. Программа вставит в рабочую книгу новый рабочий лист и скопирует в него все исходные записи, представляющие данное число. В окне Excel это прекрасный способ быстрого запроса набора данных.

В VBA эквивалентным свойством является .ShowDetail. Присвоив этому свойству любой ячейки в сводной таблице значение True, вы сгенерируете новый рабочий лист со всеми записями, составляющими данную ячейку: PT.TableRange2.Offset(2, 1).Resize(1, 1).ShowDetail = True

В листинге Sub RetrieveTop3CustomerDetail() создается сводная таблица с общим объемом продаж для трех лучших магазинов с использованием свойства .ShowDetails для каждого магазина. Это свойство является альтернативой команде *Расширенный фильтр*. В результате выполнения соответствующего макроса вы получите три новых листа. На рис. 24 показан первый из них.



Рис. 24. Свойство .ShowDetail для выборки записей, соответствующих одному из заказчиков

**Создание отчетов по каждому региону или названию оборудования.** Сводная таблица может содержать одно или несколько полей фильтров отчета. Поле фильтра располагается в отдельном наборе строк над самим отчетом сводной таблицы. Оно может служить для фильтрации отчета по определенному региону, названию оборудования или комбинации региона и названия оборудования. В VBA поле фильтра отчета называется полем страницы. Можно создать сводную таблицу, включающую несколько полей фильтра. Поля фильтра также уместно использовать для создания отчета по каждому региону.

Чтобы настроить поле страницы в VBA, используйте параметр PageFields метода AddFields. Следующая строка кода создает сводную таблицу с полем *Регион* в качестве поля страницы.

PT.AddFields RowFields:="Заказчик", ColumnFields:="Данные", \_

PageFields:="Регион"

Эта строка кода устанавливает для поля страницы *Регион* значение (все), которое определяет вывод всех регионов. Чтобы ограничить отчет лишь северным регионом, используйте свойство CurrentPage: PT.PivotFields("Регион").CurrentPage = "Север"

Один из способов эффективного использования полей страниц заключается в создании пользовательской формы, в которой свойство CurrentPage устанавливает пользователь.

Можно использовать свойство ShowPages для создания копии сводной таблицы, предназначенной для каждого элемента раскрывающегося поля фильтра. Если в набор данных включены сведения о восьми регионах, то после выполнения следующей строки кода в книгу будут вставлены восемь новых листов (по одному для каждого региона). Сводная таблица отображается на каждом листе вместе с соответствующим регионом, выбранным в раскрывающемся списке. PT.ShowPages PageField:=Регион

Будьте осторожны при использовании свойства ShowPages. Если попытаться воспользоваться этим свойством при выбранном поле Заказчик, содержащем 1000 записей, Excel попытается вставить в книгу 1000 листов и на каждом из них создать сводную таблицу. Все эти сводные таблицы используют один и тот же кеш для минимизации используемой оперативной памяти. Это может привести к нехватке памяти, что вызовет аварийное завершение выполняемого кода.

Еще одна проблема, связанная с использованием свойства ShowPages, заключается в том, что отчеты создаются в виде отдельных книг. На практике обычно требуется создавать отдельные книги для каждого региона, чтобы пересылать отчеты по электронной почте заинтересованным лицам. Эта проблема решается циклическим перебором объектов Pivotltems, выводимых поочередно в поле страницы. Используя этот метод, можно быстро создать отчет по пяти лучшим магазинам для каждого региона.

До сих пор для копирования данных из сводной таблицы применялось свойство PT.TableRange2. Оно включает все строки сводной таблицы, в том числе и поля страниц. Применяется также свойство .TableRange1, которое исключает поля страниц. Чтобы отобразить строки из области данных, воспользуйтесь одним из следующих двух выражений:

PT.TableRange2.Offset(3, 0)

PT.TableRange1.Offset(1, 0)

Если использовать свойство TableRange2, то проблем при попытке удаления сводной таблицы с помощью метода PT. TableRange2.Clear не возникнет. Если вы случайно попытаетесь очистить TableRange1 с полями страниц, то получите сообщение о невозможности перемещения или удаления части сводной таблицы.

В результате выполнения кода из листинга Sub Top5ByRegionReport() создается 7 рабочих книг для каждого региона. Книги содержат отчеты (но не сводные таблицы) о продажах по 5 первым заказчикам.

**Фильтрация нескольких элементов в сводной таблице вручную.** Помимо настройки вычисляемого сводного элемента для отображения итога по нескольким товарам, можно вручную фильтровать отдельные поля. Например, клиент, покупающий обувь, запросил отчет по продажам сандалий только в штатах с теплым климатом. Код сокрытия отдельного магазина:

РТ.PivotFields("Магазин").Pivotltems("Миннеаполис").Visible = False

Проявляйте осторожность и никогда не устанавливайте значение False для всех элементов. В противном случае макрос вызовет ошибку. Это происходит чаще, чем можно ожидать. Приложение может вначале вывести товары А и В, а затем в следующем цикле — товары С и D. Если вы попытаетесь скрыть товары А и В перед выводом товаров С и D, то попадете в ситуацию, когда в поле вообще не будет отображаемых товаров, что равнозначно ошибке. Чтобы решить эту проблему, циклически обрабатывайте все элементы Pivotltems и устанавливайте для них значения Visible перед повторным запуском цикла.

Эта проблема легко решается в VBA. После создания таблицы с полем *Название оборудования* в поле страниц выполните цикл изменения свойства Visible и вывода итогов по определенным товарам. Для этого используется следующий код:

' Все элементы Pivotltems видимы

For Each Pivltem In PT.PivotFields("Название оборудования").Pivotltems

Pivltem.Visible = True

Next Pivltem

' Циклический просмотр и отображение выбранных элементов

For Each Pivltem In

PT.PivotFields("Название оборудования").Pivotltems

Select Case Pivltem.Name

Case "Landscaping/Grounds Care", "Green Plants and Foliage Care"

Pivltem.Visible = True

Case Else

Pivltem.Visible = False

End Select

Next Pivltem

**Использование концептуальных фильтров.** Начиная с версии Excel 2007 в программе появились концептуальные фильтры для полей данных, числовых и текстовых полей. В списке полей сводной таблицы установите указатель мыши над произвольным активным полем. В появившемся раскрывающемся списке можно выбрать один из трех видов фильтра: фильтры по дате, фильтры по значению, фильтры по подписи. Для применения фильтра по подписи в VBA воспользуйтесь методом PivotFilters.Add. Следующий код фильтрует список заказчиков, извлекая элементы, которые начинаются с 1.

РТ.PivotFields("Заказчик").PivotFilters.Add \_

Type:=xlCaptionBeginsWith, Valuel:="1"

Чтобы очистить поле Заказчик от имеющихся фильтров, используйте метод ClearAllFilters:

PT.PivotFields("Заказчик").ClearAllFilters

Чтобы применить фильтр по дате к полю и отобразить записи, соответствующие текущей неделе:

PT.PivotFields("Дата").PivotFilters.Add Type:=xlThisWeek

Фильтры по значению позволяют фильтровать данные одного поля по значению в другом поле. Например, чтобы найти все рынки сбыта, в которых общий доход превышает 100 тыс. долларов, выполните следующий код:

PT.PivotFields("Рынок сбыта").PivotFilters.Add \_

Type:=xlValueIsGreaterThan, \_

DataField:=PT.PivotFields("Сумма по полю Доход"), \_

Value1:=100000

Еще один фильтр по значению позволяет определить рынки сбыта, доходы по которым находятся в диапазоне от 50 до 100 тыс. долларов. В этом случае применяются два граничных значения: Value1 и Value2:

PT.PivotFields("Рынок сбыта").PivotFilters.Add \_

Type:=xlValueIsBetween, \_

DataField:=PT.PivotFields("Сумма по полю Доход "), \_

Value1:=50000, Value2:=100000

Все возможные типы фильтров, применяемые в сводных таблицах см., например, [здесь](http://msdn.microsoft.com/en-us/library/office/ff193663(v=office.15).aspx).

**Использование поля поиска фильтра.** В Excel 2010 в раскрывающемся списке фильтра появилось поле Поиск. К сожалению, в VBA отсутствует эквивалент этого приятного дополнения интерфейса Excel. На рис. 25 показан флажок *Выделить все результаты поиска*, VBA-эквивалент которого просто выводит все элементы, которые соответствуют выделенным элементам. Если воспользоваться средством записи макросов, будет создан макрос, состоящий из 5 876 строк, который выполняет просмотр всех заказчиков, скрывая тех, в названии которых отсутствует слог "be".

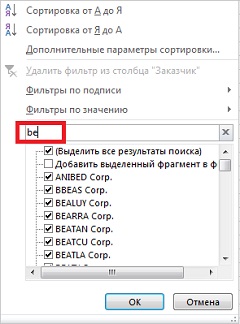


Рис. 25. В Excel 2010 появилось поле поиска в раскрывающемся списке фильтра. В VBA можно имитировать это свойство с помощью устаревшего фильтра xlCaptionContains

В версии Excel 2013 VBA отсутствуют новые средства, позволяющие имитировать действие этого флажка. Поэтому для выполнения этой задачи используется прежнее средство — фильтр xlCaptionContains, описанный в предыдущем разделе.

**Фильтрация данных в сводной таблице с помощью срезов.** В Excel 2010 появились срезы, позволяющие выполнять фильтрацию данных с помощью визуальных фильтров (подробнее см. [Excel 2013. Срезы сводных таблиц; создание временной шкалы](http://baguzin.ru/wp/?p=9538)). Можно изменять размеры и положение среза. Можно также изменять цвет среза и количество отображаемых столбцов с данными. С помощью VBA можно выбирать либо удалять отдельные элементы среза. На рис. 26 показана сводная таблица, включающая два среза. Для среза *Штат* количество столбцов было увеличено до пяти. Срез *Территория* основан на поле *Регион*.

Для создания среза используются объекты SlicerCache и Slicer. Для определения кеша среза следует указать сводную таблицу в качестве источника данных и имя поля в качестве SourceField. Объект SlicerCache определяется на уровне рабочей книги. Благодаря этому можно определять объект Slicer на листе, который не содержит сводную таблицу.

Dim SCS As SlicerCache

Dim SCI As SlicerCache

Set SCS = ActiveWorkbook.SlicerCaches.Add(Source:=PT, \_

SourceField:="Штат")

Set SCI = ActiveWorkbook.SlicerCaches.Add(Source:=PT, \_

SourceField:="Регион")

После определения кеша среза (объект SlicerCache) можно приступать к созданию самого среза. При этом срез определяется в виде объекта кеша среза. Задайте рабочий лист, на который будет помещен срез. С помощью аргумента Caption создается заголовок среза. Укажите размеры среза, задавая значения ширины и высоты (в пунктах). Укажите местоположение среза посредством координат верхнего левого угла (в пунктах). В следующем примере значения высоты, ширины и координаты верхнего левого угла среза устанавливаются в соответствии с местоположением определенного диапазона ячеек:

Dim SLS As Slicer

Set SLS = SCS.Slicers.Add(SlicerDestination:=ActiveSheet, \_

Name:="Штат", Caption:="Штат", \_

Top:=WSD.Range("O2").Top, \_

Left:=WSD.Range("O2").Left, \_

Width:=WSR.Range("O2:U2").Width, \_

Height:=WSD.Range("O2:O17").Height)

Любой созданный срез содержит один столбец. Стиль и количество столбцов среза можно изменить с помощью следующего кода:

With SLS

.Style = "SlicerStyleLight6"

.NumberOfColumns = 5

End With

Сразу же после создания среза можно воспользоваться кодом VBA для выбора требуемых элементов, входящих в его состав. Для выбора элемента среза потребуется изменить свойство Slicerltem, которое относится к объекту SlicerCache, а не к объекту Slicer:

With SCR

.Slicerltems("Средний Запад").Selected = True .Slicerltems("Север").Selected = True

.SlicerItems("СевероВосток").Selected = True .SlicerItems("Юг").Selected = True

.SlicerItems("ЮгоВосток").Selected = True .SlicerItems("ЮгоЗапад").Selected = True

.SlicerItems("Запад").Selected = True

End With

Для форматирования срезов, представленных на рис. 26, использовалась процедура Sub AddSlicers (к сожалению, у меня она также отработала некорректно).

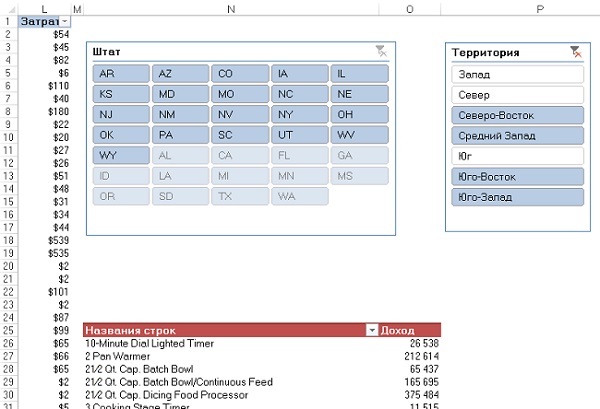


Рис. 26. С помощью срезов можно построить визуальные фильтры по полям *Штат* и *Регион*

**Использование модели данных в Excel 2013**

В Excel 2013 реализованы функции PowerPivot. Объекты, доступ к которым открывается с ленты Excel, включены в модель данных (подробнее см. [Обработка данных с помощью модуля PowerPivot](http://baguzin.ru/wp/?p=10449)). Можно добавить две таблицы в модель данных, создать связь между ними, а затем создать сводную таблицу на основе модели данных. Чтобы выполнить примеры, рассматриваемые в этом и следующих разделах, откройте файл VBA в модели данных.xlsm. Модель данных этой рабочей книги включает две таблицы: Sales и Sector (не перепутайте с листами книги Excel). Таблица Sector — это таблица подстановки, которая связана с таблицей Sales с помощью поля Заказчик. Чтобы построить сводную таблицу, создайте макрос, который будет выполнять следующие действия:

1. Добавление главной таблицы в модель данных.
2. Добавление таблицы подстановки в модель данных.
3. Создание связи между таблицами.
4. Создание кеша сводной таблицы с помощью объекта ThisWorkbookDataModel.
5. Создание сводной таблицы на основе кеша.
6. Добавление полей в область строк.
7. Создание меры и ее добавление в сводную таблицу.

**Добавление двух таблиц в модель данных.** Для дальнейшей работы потребуется набор данных в рабочей книге на листе *Продажи*, который был преобразован в таблицу путем нажатия комбинации клавиш Ctrl+T. С помощью контекстной вкладки *Конструктор*, относящейся к набору контекстных вкладок *Работа с таблицами*, в области *Свойства* измените имя таблицы на Sales. Чтобы добавить таблицу в модель данных, воспользуйтесь соответствующим фрагментом кода процедуры Sub BuildModelPivotTable (этот фрагмент выделен примечаниями внутри кода).

Несколько переменных, применяемых в коде, используют имя таблицы, путь к рабочей книге и/или название рабочей книги. В переменной, находящейся в верхней части кода, хранится имя таблицы. На основе значения этой переменной можно создать название подключения, строку подключения и текст команды.

Чтобы создать связь с таблицей подстановки, потребуется лишь изменить значение переменной TableName.

**Создание связи между двумя таблицами.** Для создания связи между таблицами программным образом используется довольно простой код VBA. Условие для корректного выполнения этого кода — наличие единственной модели данных в каждой рабочей книге. Присвойте объектной переменной МО ссылку на модель данных, находящуюся в рабочей книге. Воспользуйтесь методом ModelRelationships.Add, задающим два связанных поля (см. фрагмент кода после комментария *Связывание двух таблиц*).

**Задание кеша и создание сводной таблицы.** Код, задающий кеш сводной таблицы, определяет, что данные являются внешними. Даже если связанные таблицы находятся в рабочей книге и модель данных хранится в виде большого двоичного объекта книги, все равно используется подключение внешних данных. Это подключение называется ThisWorkbookDataModel (см. два фрагмента кода, начинающиеся комментариями *Задание кеша сводной таблицы* и *Создание сводной таблицы на основе кеша*).

**Добавление полей из модели данных в сводную таблицу.** В сводной таблице будут использоваться два типа полей. Текстовые поля, такие как Заказчик, Сектор и Товар, представляют собой обычные поля, добавляемые в область столбцов сводной таблицы. По отношению к этим полям не выполняются какие-либо вычисления. Чтобы добавить числовое поле в область значений понадобится явно создать поле и добавить его в таблицу (см. фрагмент кода, начинающийся с комментария *Добавление поля Сектор из таблицы Сектор в область строк*).

**Добавление числовых полей в область значений сводной таблицы.** В модуле PowerPivot for Excel 2010 вычисляемые поля назывались мерами. В Excel 2013 эти же поля называются вычислениями. Но в исходном коде VBA они все равно называются мерами. Если в сводной таблице, включенной в модель данных, установить флажок возле поля Доход, это поле переместится в область ЗНАЧЕНИЯ. При этом Excel неявным образом создает новую меру *Сумма по полю Доход* (для просмотра неявно заданных мер перейдите в окно PowerPivot). В VBA следует начать с создания новой меры для поля *Сумма по полю Доход*. Чтобы облегчить ссылку на эту меру в дальнейшем, присвойте её объектной переменной (см. фрагмент кода, начинающийся с комментария *Перед добавлением поля Доход в сводную таблицу нужно создать меру*). Все функции, доступные в версии Office 2013 Standard, могут быть запрограммированы с помощью кода VBA. К сожалению, это не касается вычисляемых полей, созданных на основе языка формул DAX.

На рис. 27 показана сводная таблица модели данных, созданная в результате выполнения кода процедуры Sub BuildModelPivotTable (к сожалению, и этот код у меня не заработал).

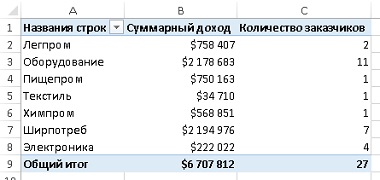


Рис. 27. С помощью макроса задана связь между двумя таблицами и созданы две меры

1. Заметка написана на основе книги Джелен, Александер. [Сводные таблицы в Microsoft Excel 2013](http://baguzin.ru/wp/?p=9613). Глава 13. [↑](#footnote-ref-1)