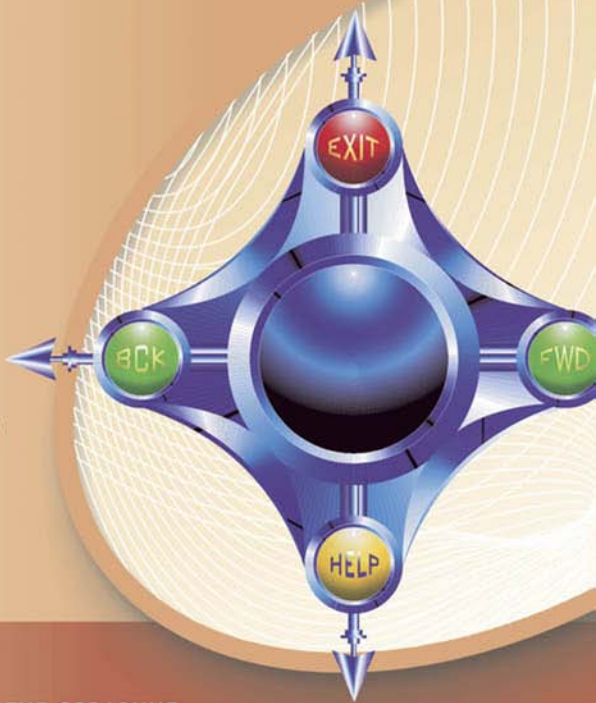


# VBA

## 2-е издание

Андрей Гарнаев

- Автоматизация повседневной работы
- Около 350 примеров разработки приложений
- Основные элементы, объекты и методы
- Использование VBA для взаимодействия с Web и при работе с БД



*Технология создания  
пользовательских приложений на примерах*

**Андрей Гарнаев**

**САМОУЧИТЕЛЬ**

**WBVA**

**2-е издание**

Санкт-Петербург

«БХВ-Петербург»

2004

УДК 681.3.068+800.92VBA  
ББК 32.973.26-018.1  
Г20

**Гарнаев А. Ю.**

Г20 Самоучитель VBA. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2004. — 560 с.: ил.

ISBN 978-5-94157-410-0

В книге изложены базовые сведения о VBA, работе с макросами, технологии ООП, конструировании пользовательского интерфейса и форм, работе с файлами, создании Web-служб и защите приложений. Приведены практические приемы автоматизации работы в офисных приложениях Excel, Word, Access и PowerPoint, обмена XML-документами между приложениями, а также управления работой разных приложений из одного файла, написанного на VBA. Показаны возможности интегрирования офисных приложений в .NET-приложения с использованием Microsoft Visual Studio Tools для Microsoft Office System. Рассмотрены вопросы создания скриптов для Windows, использования компонентов COM и Win32 API, программирования среды разработки кода и др. Книга насыщена большим числом примеров (около 350). Во втором, существенно переработанном и дополненном издании, отражены изменения последних лет в области разработки офисных приложений.

*Для программистов*

УДК 681.3.068+800.92VBA  
ББК 32.973.26-018.1

### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. гл. редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Наталья Бубнова</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Светлана Симуни</i>
Дизайн обложки	<i>Игоря Цырульникова</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 09.06.04.

Формат 70×100<sup>1/16</sup>. Печать офсетная. Усл. печ. л. 45,15.

Тираж 6000 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Гигиеническое заключение на продукцию, товар № 77.99.02.953.Д.001537.03.02 от 13.03.2002 г. выдано Департаментом ГСЭН Минздрава России.

Отпечатано с готовых диапозитивов  
в ГУП "Типография "Наука"  
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-94157-410-0

© Гарнаев А. Ю., 2004  
© Оформление, издательство "БХВ-Петербург", 2004

# Содержание

<b>Предисловие</b> .....	<b>1</b>
О чем данная книга.....	1
Для кого предназначена эта книга.....	3
Об авторе.....	3
<b>Глава 1. Введение</b> .....	<b>5</b>
Что такое VBA.....	5
Макросы.....	6
Запись макроса.....	7
Выполнение макроса.....	10
Редактирование макроса.....	11
Назначение макроса объекту.....	13
Как записывать относительную ссылку на ячейки.....	14
Где хранятся макросы.....	15
Защита от макросов.....	15
Элемент управления <i>Кнопка</i> и создание процедуры обработки события, при которой выполняется макрос.....	16
<b>Глава 2. Основы программирования на VBA</b> .....	<b>19</b>
Типы данных.....	19
Переменные.....	19
Определение времени жизни и области видимости переменных.....	20
Установка необходимости определения переменных с помощью директивы <i>Option Explicit</i> .....	21
Ключевое слово <i>Null</i> .....	21
Комментарии.....	21
Перенос строки кода.....	22
Строки.....	22
Даты.....	22

Массивы .....	23
Динамические массивы.....	24
Определение границ индексов массива .....	25
Константы .....	26
Перечисляемый тип .....	26
Тип данных, определенный пользователем .....	27
Математические операции .....	28
Операции отношения.....	28
Логические операции.....	29
Директива <i>Option Compare</i> .....	29
Математические функции .....	30
Создание последовательности случайных чисел.....	31
Как найти значение числа $\pi$ .....	32
Функции проверки типов.....	32
Преобразование строки в число .....	32
Преобразование числа в строку.....	33
Другие функции преобразования типов .....	33
Форматирование числа функцией <i>Format</i> .....	34
Пользовательские форматы .....	36
Форматирование чисел функцией <i>FormatNumber</i> .....	38
Форматирование процентов функцией <i>FormatPercent</i> .....	38
Задание денежного формата функцией <i>FormatCurrency</i> .....	39
Форматирование даты и времени функцией <i>FormatDateTime</i> .....	39
Встроенные диалоговые окна .....	39
Окно ввода.....	39
Проверка вводимых данных.....	41
Ввод ссылки на диапазон.....	42
Окно сообщения .....	43
Определение нажатой кнопки в окне сообщений .....	46
Оператор присваивания.....	46
Присваивание переменной ссылки на объект .....	47
Оператор <i>With</i> .....	47
Операторы управления .....	47
Оператор условного перехода.....	48
Избегайте оператор условного перехода, если это возможно .....	49
Оператор выбора.....	49
Оператор <i>For Next</i> .....	50
Оператор <i>For Each</i> .....	51
Оператор <i>While</i> .....	51
Оператор <i>Do</i> .....	52
Оператор безусловного перехода <i>GoTo</i> .....	53
Процедура.....	53
Передача параметров по ссылке и значению .....	56
Рекурсивные процедуры .....	57

<b>Глава 3. Интегрированная среда разработки .....</b>	<b>59</b>
Где набирается код VBA .....	59
Структура редактора VBA .....	60
Окно <i>Project — VBAProject</i> .....	60
Копирование модулей и форм из одного проекта в другой .....	61
Окно редактирования кода .....	61
Интеллектуальные возможности редактора кода .....	62
Окно <i>UserForm</i> (Редактирование форм) .....	64
Окно <i>Properties</i> (Свойства) .....	67
Окно <i>Object Browser</i> (Просмотр объектов) .....	68
Программирование среды разработки .....	69
Экспортирование модуля .....	70
Удаление модуля .....	71
Программа, которая выполняется только один раз .....	71
Импорт данных в модуль из текстового файла .....	71
Очистка содержания модуля .....	72
Как проверить, существует ли модуль либо процедура .....	72
Автоматическое создание кода в модуле .....	73
Удаление кода процедуры из модуля .....	74
Список всех процедур модуля .....	75
Создание пользовательской формы .....	75
Построение формы с элементом управления, реагирующим на события .....	76
<b>Глава 4. Элементы ООП в VBA .....</b>	<b>79</b>
Классы и их экземпляры .....	79
Объявление класса .....	80
Создание экземпляра класса .....	80
Инициализация значений полей .....	81
Методы и свойства .....	82
События .....	84
Таймер, как пример класса, генерирующего события .....	87
Объект <i>Collection</i> .....	89
<b>Глава 5. Работа со строками, временем и датами .....</b>	<b>93</b>
Функции обработки строк .....	93
Нахождение ASCII-кода по литералу и литерала по ASCII-коду .....	93
Преобразование строки к нижнему или верхнему регистру .....	94
Как сделать так, чтобы каждое слово в предложении начиналось с заглавной буквы .....	94
Возвращение подстроки по указанному местоположению .....	94

Определение длины строки .....	95
Копия строки без начальных и конечных пробелов .....	95
Строка, состоящая из указанного числа пробелов .....	95
Строка, состоящая из указанного числа повторяющихся символов .....	96
Преобразование строки в массив .....	96
Преобразование массива в строку .....	97
Сравнение строк .....	97
Создание зеркальной строки по отношению к данной .....	97
Нахождение вхождения в строку подстроки .....	98
Замена в строке подстроки .....	98
Замена всех специфицированных подстрок в рабочем листе .....	99
<b>Функции времени и даты .....</b>	<b>99</b>
Определение текущей системной даты .....	99
Определение текущего системного времени .....	99
Определение текущей даты и системного времени .....	100
Извлечение из времени часового, минутного и секундного компонентов .....	100
Извлечение из даты годового, месячного и дневного компонентов .....	100
Определение дня недели .....	101
Определение числа секунд, прошедших с полуночи .....	101
Определение количества лет, кварталов, месяцев, недель и дней, прошедших между двумя датами .....	102
Определение компонента даты .....	103
Добавление к дате указанного временного интервала .....	103
Преобразование часов, минут и секунд в формат времени .....	104
Преобразование года, месяца и дня в формат даты .....	104
Преобразование строки в формат времени .....	104
Проверка, является ли год високосным .....	105

## **Глава 6. Пользовательские функции..... 107**

Где пишется код функции пользователя? .....	107
Ваша первая функция пользователя .....	107
Математические функции .....	109
Как найти значение числа $\pi$ ? .....	110
Определение числового формата ячейки .....	111
Расчет комиссионных .....	111
Функция, имеющая несколько параметров .....	112
Функция, проверяющая вхождение данной строки в другую по образцу .....	113
Функция, возвращающая Web-адрес из ячейки с гиперссылкой .....	113
Расчет сложных комиссионных .....	114
Функция, параметрами которой является диапазон .....	115
Подсчет числа ячеек со значениями из указанного диапазона .....	115

Функция с необязательными параметрами .....	116
Использование неопределенного числа параметров.....	117
Использование массива в качестве параметра функции .....	118
Функция, возвращающая массив значений .....	119
Функция, вид которой зависит от параметра .....	120
Расчет следующего понедельника .....	121
Определение возраста человека .....	121
Надстройки.....	122

## **Глава 7. VBA и Excel .....** **123**

Объектная модель.....	123
Полная и неявная ссылка на объект .....	124
Объект <i>Application</i> .....	125
Свойства объекта <i>Application</i> .....	125
Ссылка на активную рабочую книгу, лист, ячейку, диаграмму и принтер .....	126
Загрузка инсталлированной надстройки .....	127
Управление уровнем безопасности .....	128
Рабочая книга, в которой выполняется данный макрос .....	129
Установка заголовка окна MS Excel .....	129
Работа со строкой состояния.....	130
Установка выполнения специфицированной процедуры при нажатии заданной комбинации клавиш .....	130
Как переопределить горячие клавиши приложения.....	132
Установка курсора.....	132
Семейство встроенных диалоговых окон .....	134
Объект <i>FileDialog</i> .....	134
Отображение встроенных предупреждений о работе программы .....	136
Поиск файлов .....	136
Печать активного листа .....	138
Предварительный просмотр книги .....	138
Доступ из кода к функциям рабочего листа .....	138
Номер версии <i>MS Excel</i> .....	139
Определение локальной версии Excel и установок Windows.....	139
Методы объекта <i>Application</i> .....	140
Проверка правописания отдельного слова.....	140
Преобразование имени MS Excel в объект или значение .....	141
Симулирование вычисления арифметических выражений.....	141
Простейший функциональный калькулятор .....	141
Симулирование ячеек рабочего листа .....	142
Применение квадратных скобок при симулировании операций на рабочем листе .....	142
Назначение выполнения процедуры на определенное время .....	143



Электронные часы в ячейке рабочего листа.....	143
Электронный будильник.....	144
Отмена задания у часов.....	144
Закрытие приложения.....	145
Сохранение изменений во всех рабочих книгах.....	145
Приостановка работы приложения до указанного времени.....	145
События объекта <i>Application</i> .....	146
Отключение генерации событий.....	148
Семейство <i>Workbooks</i> .....	149
Создание новой рабочей книги.....	149
Открытие рабочей книги.....	149
Объект <i>Workbook</i> .....	150
Свойства объекта <i>Workbook</i> .....	150
Закрытие книги без сохранения изменений.....	154
Определение объекта <i>Workbook</i> по имени рабочей книги.....	155
Объект <i>Name</i> и простой способ удаления из рабочей книги ненужных имен.....	155
Методы объекта <i>Workbook</i> .....	156
Установка и снятие защиты книги.....	158
Как сохранить рабочую книгу, чтобы ее именем была текущая дата.....	159
Как определить, была ли сохранена открытая рабочая книга.....	159
События объекта <i>Workbook</i> .....	160
Запрет закрытия рабочей книги.....	161
Семейство <i>Worksheets</i> .....	161
Вставка нового листа с именем, отличным от существующих.....	164
Объект <i>Worksheet</i> .....	164
Свойства объекта <i>Worksheet</i> .....	164
Проверка, установлена ли защита на содержании рабочего листа.....	168
А как проверить, существует ли лист.....	168
Методы объекта <i>Worksheet</i> .....	169
Удаление рабочего листа без предупреждения пользователя.....	170
Установка и снятие защиты с рабочего листа.....	170
Объект <i>Protection</i> (как определить, какая защита установлена на рабочем листе).....	171
Объект <i>PageSetup</i> , и как вывести в колонтитул имя книги и рабочего листа и текущую дату.....	172
События объекта <i>Worksheet</i> .....	173
Блокировка действий, связанных с событием по умолчанию.....	173
Автоматическое переформление таблицы при изменении в ней значений.....	176
Автоматический ввод данных в верхнем регистре данного диапазона.....	177

Вывод суммы значений из выделенного диапазона в строку состояния.....	178
Объекты <i>Range</i> и <i>Selection</i> .....	178
Адресация ячеек.....	179
Задание групп строк и столбцов.....	180
Связь объекта <i>Range</i> и свойства <i>Cells</i> объекта <i>Worksheet</i> .....	180
Свойства объекта <i>Range</i> .....	181
Ввод или считывание значения из диапазона .....	181
Поиск по шаблону подобных значений в диапазоне .....	182
Ввод или считывание формулы в ячейку в формате A1.....	182
Ввод или считывание формулы в ячейку в формате R1C1.....	182
Ввод или считывание формулы локальной версии в ячейку в формате A1 .....	183
Ввод или считывание формулы локальной версии в ячейку в формате R1C1 .....	183
Ввод формулы массива в диапазон .....	183
Ввод формулы массива локальной версии в диапазон.....	183
Ввод формулы массива в диапазон с относительными ссылками на ячейки.....	183
Определение адреса ячейки .....	184
Управление стилем границы диапазона и объектами <i>Border</i> .....	185
Функции <i>RGB</i> и <i>QBColor</i> .....	186
Доступ к отдельным ячейкам диапазона.....	187
Выбор элементов на рабочем листе или в книге .....	188
Как проверить, является ли выбранный объект диапазоном .....	190
Объект <i>Characters</i> (как форматировать часть содержимого ячейки).....	191
Объект <i>Comment</i> (создание комментариев к диапазону) .....	193
Определение текущего диапазона .....	194
Нахождение крайней ячейки диапазона в указанном направлении.....	195
Нахождение строки и столбца, содержащих данную ячейку.....	195
Объект <i>Hyperlink</i> (задание гиперссылки).....	195
Объект <i>Font</i> (задание шрифта).....	197
Объект <i>Interior</i> (заливка диапазона) .....	198
Разрешение редактирования содержимого ячеек на защищенном рабочем листе .....	200
Установка числового формата .....	201
Нахождение диапазона, сдвинутого относительно данного на указанное число строк и столбцов.....	201
Задание угла, под которым выводится текст в диапазоне.....	202
Переопределение размеров диапазона.....	203
Методы объекта <i>Range</i> .....	203
Активизация и выбор диапазона .....	203

Вставка и удаление комментариев в диапазон .....	204
Заполнение диапазона прогрессией.....	204
Автозаполнение ячеек диапазона элементами последовательности .....	206
Табуляция функции .....	208
Заполнение диапазона по одному значению .....	209
Очистка ячейки .....	209
Копирование, вырезание и удаление данных из диапазона .....	209
Специальная вставка.....	210
Вставка диапазона с транспонированием .....	211
Копирование диапазона в буфер обмена как растровое изображение.....	211
Бегущая картинка.....	212
Поиск значения в диапазоне .....	213
Повторный поиск и поиск всех значений .....	215
Замена значений.....	215
Подбор параметра и решение уравнения с одной неизвестной.....	216
Ввод в диапазон неповторяющихся значений .....	218
Отсылка электронной почты.....	219
Озвучивание текста .....	220
Условное форматирование .....	221
Отображение результата только при условии ввода всех данных .....	225
Проверка вводимых значений.....	225

## **Глава 8. Формы и элементы управления .....**

Первая форма.....	228
Форма как объект .....	229
Отображение формы при выборе ячейки рабочего листа .....	230
Как запустить проект на исполнение.....	231
Ключевое слово <i>Me</i> .....	231
Создание формы в коде .....	231
Предотвращение закрытия окна с помощью кнопки <i>Close</i> .....	233
Надпись .....	233
Всплывающая форма.....	234
Кнопка .....	236
Событие <i>Click</i> .....	238
Поле .....	238
Список .....	239
Выбор нескольких элементов из списка.....	240
Заполнение списка названиями месяцев.....	242
Поле со списком.....	243
Рисунок.....	245
Вставка в рисунок изображения выделенного диапазона.....	245

Флажок .....	247
Использование списка вместо группы флажков.....	248
Изменение цвета выделенного диапазона .....	250
Рамка.....	250
Переключатель.....	250
Запуск и остановка часов при помощи переключателя.....	252
Выключатель .....	253
Полоса прокрутки .....	255
Счетчик.....	256
Как можно перебирать элементы управления.....	259
Обеспечение функционирования кнопки только в случае заполненных полей .....	260
Обмен сообщениями между формами .....	261
Объект <i>DataObject</i> .....	263

## **Глава 9. Создание меню, контекстного меню и панели инструментов ..... 265**

Создание строки меню .....	265
Типичные ошибки, возникающие при создании пользовательских панелей инструментов.....	272
Пункты меню со встроенными функциями и картинками .....	273
Добавление нового элемента в существующее меню .....	275
Создание панели инструментов.....	277
Расположение нескольких панелей инструментов в один ряд .....	284
Добавление раскрывающегося списка в панель инструментов.....	285
Конструирование контекстного меню .....	287

## **Глава 10. Диаграммы ..... 291**

Построение диаграммы.....	291
Как получить ссылку на активную диаграмму.....	292
Как методом <i>ChartWizard</i> изменить параметры диаграммы.....	293
Изменение типа диаграммы с помощью метода <i>ChartWizard</i> .....	293
Построение внедренной диаграммы .....	295
Установка размеров и местоположения внедренной диаграммы .....	296
Построение диаграммы на основе массива данных .....	297
Удаление диаграммы .....	297
Сохранение диаграммы в виде графического файла.....	299
Задание цветов серий.....	301
Рисунок вместо заливки серий.....	302
Установка градиентной заливки серий.....	303
Изменение параметров диаграммы .....	305
Печать и предварительный просмотр печати диаграммы .....	308
Построение диаграммы на основе данных из нескольких листов .....	309

Защита диаграммы .....	311
Изменение диапазона, на основе которого конструируется диаграмма .....	311
Анимация диаграммы .....	313
События и диаграммы .....	315
Привязка событий к вложенным в рабочий лист диаграммам .....	317
Изменение типа диаграммы при помощи контекстного меню .....	318

## **Глава 11. Помощник MS Office ..... 321**

Объект <i>Assistant</i> .....	321
Объект <i>Balloon</i> .....	323
Помощник с надписями .....	326
Помощник с флажками .....	327
Немодальное окно помощника .....	329
Добавление значка в надписи и флажки помощника .....	330

## **Глава 12. Работа с файлами..... 333**

Объект <i>FileSystemObject</i> .....	333
Создание объекта <i>FileSystemObject</i> с явной ссылкой на <i>Microsoft Scripting Runtime</i> .....	333
Создание объекта <i>FileSystemObject</i> с неявной ссылкой на <i>Microsoft Scripting Runtime</i> .....	334
Как проверить, существует ли диск .....	334
Получение информации о диске .....	335
Как проверить, существует ли каталог или файл .....	336
Создание и удаление каталога .....	337
Получение информации о каталоге .....	338
Копирование и перемещение файла .....	339
Удаление файла .....	339
Информация о файле .....	340
Список всех файлов данного каталога .....	341
Открытие файла и создание <i>TextStream</i> объекта .....	342
Метод <i>OpenAsTextStream</i> .....	342
Метод <i>CreateTextFile</i> .....	343
Метод <i>OpenTextStream</i> .....	344
Запись, присоединение и считывание данных из файла .....	345
Функции по работе с файлами .....	346
Просмотр файлов в каталоге .....	347
Поиск файла и объект <i>FileSearch</i> .....	347
Как определить, существует ли рабочая книга .....	349

<b>Глава 13. Обработка ошибок и отладка программ .....</b>	<b>351</b>
Разработка процедур, предотвращающих появление ошибок .....	351
Контроль вводимых значений с помощью обработки события <i>KeyPress</i> .....	354
Перехват и обработка ошибок .....	355
Оператор <i>On Error</i> .....	356
Процедура обработки ошибки .....	356
Оператор <i>Resume</i> .....	356
Объект <i>Err</i> .....	356
Создание пользовательских ошибок для тонкой настройки обработчика ошибок .....	360
Отладка программ .....	362
Ошибки компиляции .....	362
Ошибки выполнения .....	363
Логические ошибки .....	365
Директива <i>Option Explicit</i> .....	365
Пошаговое выполнение программ .....	366
Точка прерывания .....	367
Вывод значений свойств и переменных .....	368
Окно <i>Watches</i> .....	368
Окно <i>Locals</i> .....	368
Окно <i>Immediate</i> .....	369
Программный способ вывода значений в окно <i>Immediate</i> .....	369
 <b>Глава 14. VBA и Win32 API .....</b>	 <b>371</b>
Первый пример .....	371
Отсылка электронной почты и загрузка Web-страницы .....	372
Определение и изменение разрешения экрана .....	373
Как открыть окно Проводника .....	374
Как отобразить окно свойств заданного файла .....	374
Как загрузить Web-страницу, адрес которой набран в окне ввода .....	376
Полупрозрачная форма .....	376
Форма произвольной формы .....	378
Форма без заголовка .....	381
Заголовок активного окна .....	382
 <b>Глава 15. VBA и Word .....</b>	 <b>383</b>
Создание стилей .....	383
Назначение стиля кнопке пользовательского меню .....	384
Добавление в пользовательское меню кнопки, которой назначен макрос .....	386

Программное задание стиля.....	387
Создание шаблонов.....	388
Создание документа по шаблону.....	388
Присоединение шаблона к существующему документу.....	388
Вставка символа.....	389
Заполнение письма.....	389
Полезные макросы.....	391
Печать текущей страницы.....	391
Получение статистических сведений о документе или выделенном фрагменте.....	392
Вставка фонового рисунка.....	393
Двухстраничный и одностраничный просмотр документа.....	393
Поиск текста в выделенном фрагменте.....	394
Окраска всех вхождений данного слова в документ в заданный цвет.....	394
Отображение в строке состояния информации об относительном объеме просмотренного документа.....	395

## **Глава 16. VBA и Automation .....397**

Средство <i>Automation</i> .....	397
Программные идентификаторы приложений-серверов <i>Automation</i> .....	397
Функции доступа к объектам <i>Automation</i> .....	398
Позднее и раннее связывание.....	399
Открытие документа Word.....	399
Создание документа Word.....	400
Создание отчета Word на основе данных, хранимых на рабочем листе.....	401
Создание презентации на основе данных, хранимых на рабочем листе <i>MS Excel</i> .....	404

## **Глава 17. ADO (ActiveX Data Objects) .....409**

Создание ссылки на библиотеку ADO.....	410
Объект <i>Connection</i> и установка подключения к базе данных.....	411
Объект <i>Recordset</i> и его создание.....	411
Пример использования объекта <i>Recordset</i> для просмотра базы данных.....	412
Методы, свойства и события объекта <i>Recordset</i> .....	414
Последовательный просмотр записей базы данных.....	418
Доступ к данным, расположенным на рабочих листах закрытой книги.....	421
Работа с курсорами.....	422
Объект <i>Field</i> .....	423

Создание браузера базы данных .....	424
Браузер просмотра нескольких таблиц базы данных .....	426
Редактирование, создание, обновление и удаление записей .....	429
Вывод набора записей на рабочий лист .....	435
Использование объекта <i>Command</i> .....	436
Вставка данных в рабочий лист без ADO.....	436
<b>Глава 18. VBA и Access .....</b>	<b>439</b>
Макросы в Access .....	439
Применение объекта <i>DoCmd</i> .....	440
Объектная модель <i>MS Access</i> .....	440
Открытие и закрытие <i>MS Access</i> из других приложений .....	440
Ваша первая форма в <i>MS Access</i> .....	441
Создание всплывающей формы.....	444
Обращение к объекту по имени .....	446
Источники данных .....	446
Извлечение данных из списка в несвязанный элемент управления.....	448
Программная навигация по записям .....	449
Создание документов в Word на основе данных, хранящихся в базе данных .....	451
Построение отчета.....	454
<b>Глава 19. VBA и анализ данных .....</b>	<b>457</b>
Обработка данных: сортировка, фильтрация, сводные таблицы, списки.....	457
Сортировка данных .....	457
Сортировка по выделенному полю.....	457
Метод <i>Sort</i> .....	458
Пример приложения, сортирующего данные.....	459
Фильтрация .....	462
Использование автофильтра.....	462
Описание процесса автофильтрации .....	463
Метод <i>AutoFilter</i> .....	464
Объекты <i>AutoFilter</i> и <i>Filter</i> и семейство <i>Filters</i> .....	465
Пример приложения, фильтрующего данные .....	466
Как сделать, чтобы в фильтре отображались только списки выбора указанных столбцов .....	467
Как определить число отфильтрованных записей.....	469
Сводная таблица .....	469
Подведение итогов по странам .....	471
Обновление данных .....	477
Удаление сводной таблицы.....	477



Объекты, связанные со сводной таблицей .....	477
Метод <i>PivotTableWizard</i> .....	478
Объект <i>PivotTable</i> .....	479
Объект <i>PivotCache</i> .....	480
Объект <i>PivotField</i> .....	481
Пример приложения, помогающего построить и обновлять сводную таблицу .....	482
Списки .....	486

## **Глава 20. VBA и Web ..... 491**

Отсылка сообщений по электронной почте.....	491
Задание параметров Web-страницы .....	492
Предварительный просмотр документа как Web-страницы .....	492
Сохранение документа как Web-страницы .....	493
Объект <i>PublishObject</i> .....	493
Гиперссылки .....	494
Отсылка сообщения с помощью гиперссылки .....	495
Вставка рисунка в рабочий лист из Web.....	495
Создание web-запроса .....	496
Получение данных из Интернета .....	498
Получение данных из Интернета по запросу из указанной таблицы .....	499
Web-компоненты .....	500
Компонента <i>Microsoft Office Spreadsheet</i> .....	501
Компонента <i>Microsoft Office Chart</i> .....	503
Web-службы.....	504
Создание Web-службы с помощью Visual Studio .NET .....	505
Применение Web-службы в MS Office.....	507
Конвертация валют .....	510

## **Глава 21. VBA и Windows-сценарии ..... 513**

Первый Windows-сценарий .....	513
Доступ к файловой системе .....	514
Создание документа Word и его печать.....	515
Доступ к открытому активному документу Word .....	516
Как определить, запущено ли приложение.....	516
Перевод HTML документа в RTF формат.....	517
Доступ к программам и передача им данных с клавиатуры.....	517

---

<b>Глава 22. Microsoft Visual Studio Tools для MS Office.....</b>	<b>519</b>
Ваш первый проект в Microsoft Visual Studio Tools.....	519
Перехват событий рабочего листа.....	521
Загрузка данных из базы данных на рабочий лист MS Excel.....	522
<b>Глава 23. VBA и XML.....</b>	<b>525</b>
Синтаксис XML-документа.....	525
Синтаксис XSD-документа.....	527
Открытие XML-документа в Excel без использования схемы.....	531
Открытие XML-документа в Excel с использованием схемы.....	531
Импорт XML-карты в коде.....	532
Создание связанных с картой столбцов и импорт в них данных.....	533
<b>Приложение 1. VBA и игра <i>Сапер</i>.....</b>	<b>535</b>
<b>Предметный указатель.....</b>	<b>539</b>



# Предисловие

## О чем данная книга

Операционная система Windows (особенно Windows XP и выше) корпорации Microsoft обладает изумительно удобным интерфейсом и множеством интеллектуальных средств, которые освобождают пользователя от рутинной работы, присущей другим операционным системам. Естественным и незаменимым дополнением этой операционной системы являются программные продукты Microsoft Office, которые даже начинающему пользователю позволяют с легкостью создавать идеальную по своему оформлению документацию (например, эта книга была написана и сверстана с помощью Microsoft Word), производить финансовые, инженерные и прочие расчеты при помощи Microsoft Excel, создавать и работать с базами данных средствами Microsoft Access, конструировать презентации и слайды с помощью Microsoft PowerPoint, Microsoft Outlook, создавать собственные Web-сайты средствами Microsoft FrontPage. Вместе, операционная система Windows и программные продукты Microsoft Office, покорили весь мир. Сейчас буквально нет ни одного офиса, ни одного дома, где бы не было компьютера с этими продуктами. Но это еще не все. Корпорация Microsoft интегрировала в свои офисные продукты, в саму операционную среду изумительный по простоте и необычайный по своей эффективности язык программирования Visual Basic for Applications или сокращенно VBA. С помощью этого языка теперь каждый пользователь может автоматизировать работу приложения и максимально приспособить его работу для решения текущих задач, не только добавив интерфейсу новую функциональность, но и удалив из него ненужные для данного приложения элементы, тем самым обеспечив ему дополнительную стабильность, своеобразную защиту от дурака. Например, если вам поручено в фирме ежедневно по результатам представлять сводную таблицу и диаграмму ее работы, наверное, уже через пару дней вам придет идея, почему бы эту процедуру не автоматизировать, т. к. программный инструментарий используется один и тот же, только источники данных разные. Здесь как раз вам на помощь придет VBA. А результат от созданного продукта будет двоякий — во-первых,

вы освободитесь от ежедневной работы, а во-вторых, получите шанс, чтобы босс вас заметил и понял, что вы действительно незаменимый сотрудник. VBA позволяет не только автоматизировать работу одного приложения, но и интегрировать работу нескольких в единое целое. Например, в нашем примере данные по работе фирмы пусть хранятся в базе данных Microsoft Access, расчеты производятся в Microsoft Excel, а отчет распечатывается в формате документа Microsoft Word. Кроме того, отчет надо представить в виде слайдов Microsoft Power на ежедневной планерке, а сводку о них опубликовать на Web-сайте. Оказывается всю эту комплексную задачу можно автоматизировать, причем, благодаря интегрированию. Управление можно производить из одного приложения, или даже из одного исполняемого файла, созданного на VB или VB .NET. В последнем случае пользователь даже не будет знать, что весь этот большой объем работы выполняется не данным приложением, а программными продуктами Microsoft Office.

Данная книга как раз и посвящена изучению этого замечательного языка программирования — VBA. Она является переработанным вторым изданием книги "Самоучитель VBA" издательства BHV-Петербург. С ее помощью вы получите базовые сведения о VBA, научитесь работать с макросами, освоите технологию объектно-ориентированного программирования, научитесь конструировать пользовательский интерфейс и формы, создавать защиту от дурака, работать с файлами, использовать и конструировать Web-службы. В Excel вы сумеете автоматизировать работу с рабочим листом, диаграммами и обработку данных. В Word вы научитесь создавать макросы, пользовательские стили и шаблоны. В Access узнаете, как создаются базы данных и производится работа с ними, как конструируется пользовательский интерфейс для доступа и обработки данных. В PowerPoint научитесь автоматизировать процесс создания презентаций. Узнаете, как в офисных проектах можно обрабатывать XML документы. Узнаете, как, используя Microsoft Visual Studio Tools для Microsoft Office System, в .NET приложения можно интегрировать офисные приложения. Научитесь создавать скрипты для Windows, а также применять в офисных приложениях COM компоненты и Win32 API (например, узнаете, как создать прозрачную форму и форму произвольного формата). VBA позволяет не только автоматизировать приложения, но и программировать саму среду разработки кода. Об этом также пойдет разговор в данной книге. Она научит вас программно экспортировать и удалять модули, импортировать в них текстовые файлы, как удалять, так и создавать в модулях код, создавать формы. Будет приведен пример программы, которая выполняется только один раз.

Книга насыщена большим числом примеров (около 350), которые выполняют двоякую функцию: во-первых, это готовый код, который вы сможете использовать в своих приложениях, а во-вторых, изучив эти примеры, вы с легкостью сможете самостоятельно создавать любые офисные приложения.

## Для кого предназначена эта книга

Книга рассчитана на широкий круг читателей: от начинающего пользователя, который научится самостоятельно создавать офисные приложения, до эксперта, который найдет в ней обширные справочные сведения по технологии разработки подобных приложений. Благодаря тщательной подборке обширной коллекции примеров, она может также быть полезна как студентам, изучающим офисное программирование, так и преподавателям, ведущим занятия по офисному программированию.

## Об авторе

Андрей Юрьевич Гарнаев, доктор физико-математических наук, профессор Санкт-Петербургского государственного университета.

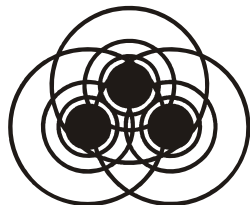
Область интересов: программное обеспечение, математическая кибернетика. Является автором десяти книг по программированию и программным средствам, двух — по теории игр, а также книг по офисному программированию: "Excel 2002: разработка приложений", "Excel, VBA и Internet в экономике и финансах", "Visual Basic .NET: разработка приложений", вышедших в издательстве БНВ-Санкт-Петербург. Более подробную информацию об авторе можно найти на сайте <http://www.apmath.spbu.ru/~kmms/garnaev/default.htm>.

Данная книга написана на основе многолетнего опыта разработки офисных приложений и курса лекций, читаемого в Санкт-Петербургском государственном университете, а также с учетом многочисленных комментариев и пожеланий читателей, которым автор искренне признателен.

Благодарю вас за выбор этой книги. Надеюсь, ее чтение для вас будет не только полезным, но и увлекательным.



# Глава 1



## Введение

В данной главе пойдет разговор о том, что такое VBA, а также будет продемонстрировано, как при помощи встроенного средства макрорекордер можно автоматизировать и визуализировать процесс создания кода. Компьютеру достаточно показать, что вы хотите получить в конечном счете, а макрорекордер сам переведет эти действия в программный код. В главе также объясняется, как макросы можно назначать различным объектам и элементам управления, тем самым создавая индивидуальный интерфейс вашего приложения, описывается, где хранятся макросы и как сделать, чтобы они были достижимы любой рабочей книге.

## Что такое VBA

Существует целый ряд систем программирования, позволяющих в той или иной степени реализовать концепцию объектно-ориентированного подхода при разработке программных средств. К ним относятся C++, Java, Visual Basic. В отличие от VB, VBA не является языком объектно-ориентированного программирования в строгом смысле этого слова, но в нем широко используются элементы объектно-ориентированного подхода и связанные с ним понятия.

Язык программирования Visual Basic for Application вначале стал применяться как средство, которое позволило Excel, а затем и другим приложениям Microsoft Office программно управлять их собственной средой. Оно взаимодействовало с другими приложениями, используя механизм автоматизации OLE.

Язык VBA — это подмножество VB, которое включает почти все его средства создания приложений, структуры данных и управляющие структуры, возможность создания пользовательских типов данных.

Visual Basic for Application, как и Visual Basic, является языком визуального и событийно управляемого программирования. Он реализует следующие возможности: создание нестандартного окна диалога в виде формы с базовым набором элементов управления, создание формы на рабочем листе, написание



процедур, обрабатывающих события, которые возникают при тех или иных действиях системы и конечного пользователя.

Язык VBA не обладает всеми возможностями VB, но позволяет работать с огромным набором объектов — по существу в нем определены все объекты приложений MS Office.

Отметим одну, может быть главную особенность программирования на VBA: создание проекта на каком-либо языке программирования осуществляется в соответствующей среде, где язык — главное средство, а создание кода (последовательности операторов) — главная цель действия программиста. А при работе на VBA целью является создание документа в широком смысле (документ Word, рабочая книга Excel, презентация, база данных Access). Проект (программа) на VBA — результат побочной деятельности по созданию документа. Более того, проект на VBA нельзя создать независимого от какого-либо документа, даже если никакие свойства этого документа не используются.

Приступая к очередному сеансу работы, программист открывает одно из приложений MS Office, и в этот момент в языке VBA автоматически становится доступным объект `Application`, определяющий это приложение, а также все встроенные в него объекты.

Итак, VBA отличается от VB и прочих языков программирования тем, что он предоставляет возможность непосредственной работы с объектами MS Office. Это позволяет эффективно его использовать для автоматизации деятельности, связанной с обработкой различных типов документов.

VBA позволяет существенно расширить вычислительные средства MS Office. С помощью VBA можно легко и быстро создавать различные приложения, даже не являясь программистом.

VBA имеет графическую инструментальную среду, позволяющую создавать экранные формы и управляющие элементы. С помощью VBA можно создавать собственные функции для Excel, вызываемые мастером функций, разрабатывать макросы, создавать собственные меню и многое другое. VBA позволяет с легкостью решать задачи, которые средствами Excel практически невозможно решить.

VBA реализует концепцию визуального программирования, управляемого событиями.

## Макросы

Если вы осуществляете многократно повторяющиеся действия, то этот процесс можно автоматизировать при помощи макросов. *Макрос (macro)* — это последовательность команд, которые написаны на VBA и которые хранятся в стандартном модуле среды разработки VBA приложений. Когда возникает необходимость выполнить данную последовательность действий, пользова-

тель может запустить на выполнение соответствующий макрос. Макросы могут осуществлять широкий спектр задач от простых вычислений до создания пользовательского интерфейса вашего приложения. Их возможности ограничены только вашей фантазией и уровнем знания VBA. Нет необходимости писать макросы вручную. В MS Office имеется встроенное средство — *макрорекордер (macro recorder)*, позволяющее преобразить все ваши действия в макросы. Их потом остается только немного отредактировать для придания большей функциональности и гибкости. Использование макрорекордера является замечательным подспорьем при изучении VBA.

## Запись макроса

Прежде чем начать запись макроса, надо тщательно спланировать ваши действия. Давайте, для примера, запишем макрос, который создает новую рабочую книгу, состоящую из единственного рабочего листа **Отчет** с шаблоном отчетной таблицы, которую остается только заполнить (рис. 1.1).

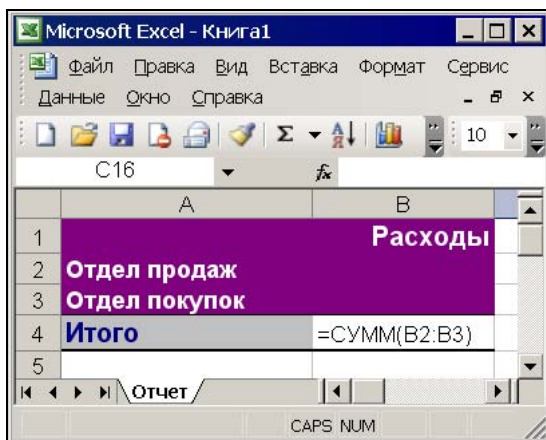


Рис. 1.1. Создаваемый шаблон таблицы

Для активизации макрорекордера выберите команду **Сервис | Макрос | Начать запись**. Появится диалоговое окно **Запись макроса** (рис. 1.2). Это диалоговое окно позволяет задать параметры макроса.

Поля **Имя макроса** и **Описание** используются для задания имени макроса и его описания. Описание важно для многократно используемых макросов. Наша память не долговечна, и, не имея подсказки в виде описания, через некоторое время будет трудно вспомнить, для чего тот или иной макрос создавался. По умолчанию макросам присваиваются имена *Макрос1*, *Макрос2* и т. д. С целью облегчения узнаваемости макроса лучше не использовать стандартное имя, а присвоить ему какое-нибудь уникальное имя, поясняющее,

для чего он используется. В нашем случае, например, присвоим макросу имя `СоздатьОтчет`. Поле **Сочетание клавиш** позволяет назначить макросу комбинацию клавиш, т. е. указать символ, который в комбинации с клавишей <Ctrl> позволить его выполнить. Назначать комбинацию клавиш макросу совсем не обязательно. Это стоит делать только для постоянно используемых макросов, для быстрого доступа к ним. Не используя комбинации клавиш, макрос можно всегда вызвать командой **Сервис | Макрос | Макросы**. Раскрывающийся список **Сохранить в** предназначен для выбора книги, в которой будет сохранен макрос. Если выбрать **Личная книга макросов**, то макрос будет сохранен в специальную скрытую книгу, в которой хранятся макросы. Эта книга всегда существует, хотя и скрыта, и записанные в ней макросы доступны для других рабочих книг. Команда **Окно | Отобразить** позволяет отобразить личную книгу макросов. Если в раскрывающемся списке выбрать **Эта книга** (т. е. выбор, который по умолчанию предлагает компьютер), то макрос сохранится на новом листе модуля в активной рабочей книге, а если выбрать **Новая книга**, то он сохранится в новой рабочей книге.

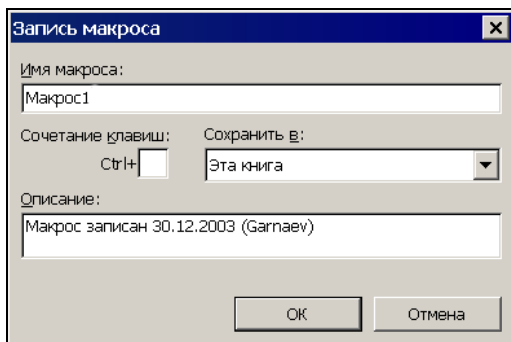


Рис. 1.2. Диалоговое окно **Запись макроса**

Итак, зададим параметры макроса в диалоговом окне **Запись макроса**:

- в поле **Имя макроса** введите `СоздатьОтчет`;
- в поле **Описание** введите `Создание рабочей книги с отчетной таблицей`;
- в поле **Сочетание клавиш** введите `a`. Нажмите кнопку **ОК**.

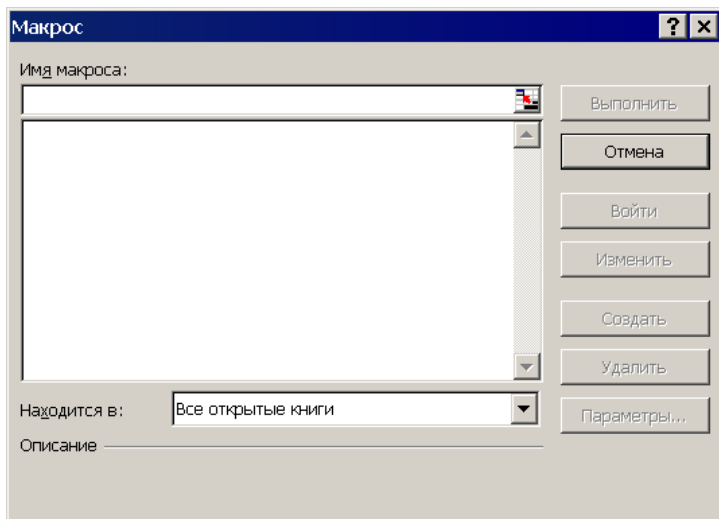
Появится плавающая панель инструментов с кнопкой  **Остановить запись**.

Теперь все производимые действия будут записываться до тех пор, пока не будет нажата эта кнопка или не будет выбрана команда **Сервис | Макрос | Остановить запись**. Итак, запишем макрос:

- выберите команду **Сервис | Параметры**. На вкладке **Общие** окна **Параметры** в поле **Листов в новой книге** введите `1` и нажмите кнопку **ОК**;
- выберите команду **Файл | Создать**. В окне **Создание книги** нажмите кнопку **Чистая книга**. В результате в проект будет добавлена новая книга;

- выберите ярлык листа **Лист1** этой книги и в контекстном меню выберите команду **Переименовать**. Измените имя листа на **Отчет**;
- выберите ячейку **A2** и введите в нее Отдел продаж;
- выберите ячейку **A3** и введите в нее Отдел покупок;
- выберите ячейку **A4** и введите в нее Итого;
- выберите столбец **A**, а затем команду **Формат | Столбец | Автоподбор ширины**;
- выберите ячейку **B1** и введите в нее Расходы;
- выберите ячейку **B4** и введите в нее при помощи кнопки **Автосумма** формулу  
`=СУММ(B2:B3)` ;
- для форматирования таблицы выделите диапазон **A1:B4**, а затем выберите команду **Формат | Автоформат**, и в окне **Автоформат** выберите **Классический 2**;
- Выберите команду **Сервис | Макрос | Остановить запись** для остановки записи.

Для просмотра только что записанной процедуры выберите команду **Сервис | Макрос | Макросы**, что приведет к отображению диалогового окна **Макрос** (рис. 1.3).



**Рис. 1.3.** Диалоговое окно **Макрос**

В диалоговом окне **Макрос** в списке выделите макрос и нажмите кнопку **Изменить**. На экране отобразится окно редактора VBA с активизированным стандартным модулем, в котором будет код (листинг 1.1) только что записанного макроса.

**Листинг 1.1. Макрос, активизирующий рабочий лист Лист2. Стандартный модуль**

```
Sub СоздатьОтчет()  
,  
' СоздатьОтчет Macro  
' Создание рабочей книги с отчетной таблицей  
,  
  
Application.SheetsInNewWorkbook = 1  
Workbooks.Add  
Sheets("Лист1").Select  
Sheets("Лист1").Name = "Отчет"  
Range("A2").Select  
ActiveCell.FormulaR1C1 = "Отдел продаж"  
Range("A3").Select  
ActiveCell.FormulaR1C1 = "Отдел покупок"  
Range("A4").Select  
ActiveCell.FormulaR1C1 = "Итого"  
Columns("A:A").Select  
Selection.Columns.AutoFit  
Range("B1").Select  
ActiveCell.FormulaR1C1 = "Расходы"  
Range("B4").Select  
ActiveCell.FormulaR1C1 = "=SUM(R[-2]C:R[-1]C)"  
Range("A1:B4").Select  
Selection.AutoFormat Format:=xlRangeAutoFormatClassic2, _  
    Number:=True, Font:=True, Alignment:=True, _  
    Border:=True, Pattern:=True, Width:=True  
End Sub
```

## Выполнение макроса

Для выполнения макроса:

выберите команду **Сервис | Макрос | Макросы**;

в окне **Макрос** выберите ссылку на макрос и нажмите кнопку **Выполнить**.

Макрос можно выполнить и при помощи горячих клавиш. В нашем случае такой комбинацией была <Ctrl>+<a>. Просто нажмите ее, и книга с шаблоном отчета будет создана.

## Редактирование макроса

Созданный макрос можно редактировать, упрощая его код и придавая ему большую общность, и, если необходимо, интерактивность. В качестве примера давайте проанализируем созданный макрорекордером код и упростим его.

Первая инструкция говорит о том, что новая книга будет состоять из одного рабочего листа. Эту инструкцию оставим без изменения.

```
Application.SheetsInNewWorkbook = 1
```

Вторая инструкция создает новую рабочую книгу. Ее тоже оставим без изменения.

```
Workbooks.Add
```

Первая из следующих двух инструкция выбирает рабочий лист **Лист1**, а затем переименовывает его в **Отчет**.

```
Sheets("Лист1").Select
```

```
Sheets("Лист1").Name = "Отчет"
```

Эти две инструкции можно заменить одной. Во-первых, код будет короче, а во-вторых, он не будет зависеть от локальной версии MS Office, т. к. доступ к листу будет производиться не по его имени, а по номеру.

```
Sheets(1).Name = "Отчет"
```

Первая из следующих двух инструкций выбирает ячейку **A2**, а вторая вводит в выбранную ячейку значение **Отдел продаж**.

```
Range("A2").Select
```

```
ActiveCell.FormulaR1C1 = "Отдел продаж"
```

Эти две инструкции также лучше заменить одной, которая вводит значение непосредственно в ячейку **A2**.

```
Range("A2").Value = "Отдел продаж"
```

Следующие четыре инструкции вводят значения в ячейки **A3** и **A4**.

```
Range("A3").Select
```

```
ActiveCell.FormulaR1C1 = "Отдел покупок"
```

```
Range("A4").Select
```

```
ActiveCell.FormulaR1C1 = "Итого"
```

Их можно заменить двумя инструкциями:

```
Range("A3").Value = "Отдел покупок"
```

```
Range("A4").Value = "Итого"
```

Первая из следующих двух инструкций выбирает столбец **A**, а вторая автоматически подбирает у выбранного столбца ширину.

```
Columns("A:A").Select
```

```
Selection.Columns.AutoFit
```