

Нина Комолова
Елена Яковлева

Программирование на VBA в Excel

2016

Основы VBA для Excel 2016,
настройки безопасности

Объектная модель Excel: объекты,
свойства, методы, события

Макросы: программирование,
запуск и отладка

Функции, определенные
пользователем

Автоматизация рабочего листа:
элементы управления формы
и ActiveX

Работа с ячейками
и диапазонами ячеек

Создание пользовательских форм,
диалоговых окон, сообщений

Графические элементы
и диаграммы средствами VBA

Работа с датами и временем

Настройки печати



Материалы
на www.bhv.ru

УДК 004.4
ББК 32.973.26-018
К63

Комолова, Н. В.

К63 Программирование на VBA в Excel 2016. Самоучитель / Н. В. Комолова, Е. С. Яковлева. — СПб.: БХВ-Петербург, 2017. — 432 с.: ил. — (Самоучитель)
ISBN 978-5-9775-0884-1

Книга научит самостоятельно создавать приложения для автоматизации работы в программе Microsoft Excel 2016 с использованием макросов и языка программирования Visual Basic for Applications (VBA). Даны теоретические сведения о программировании, элементах объектной модели Excel, запуске и отладке макросов. Рассмотрены вопросы автоматизации рабочего листа при помощи элементов управления Excel. Описаны приемы создания макросов, пользовательских функций и форм в редакторе Visual Basic. Приведены способы взаимодействия при помощи VBA с другими программами пакета Microsoft Office. Для закрепления материала рассмотрены примеры пользовательских приложений с анализом и поясняющими комментариями. Файлы рабочих книг с поддержкой макросов для каждой главы размещены на сайте издательства.

Для широкого круга пользователей

УДК 004.4
ББК 32.973.26-018

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Капальгина</i>
Редактор	<i>Леонид Кочин</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Корректор	<i>Зинаида Дмитриева</i>
Дизайн обложки	<i>Марины Дамбиевой</i>

Подписано в печать 28.02.17.
Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 34,83.
Тираж 1000 экз. Заказ №
"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

ООО "Печатное дело",
142300, МО, г. Чехов, ул. Полиграфистов, д. 1

ISBN 978-5-9775-0884-1

© Комолова Н. В., Яковлева Е. С., 2017
© ООО "БХВ-Петербург", 2017

Оглавление

- Введение 11**
 - Язык VBA 11
 - Объектно-ориентированное программирование..... 12
 - Объект..... 13
 - Объектная модель Microsoft Excel 2016 13
 - Объектная модель VBA..... 13
 - Объект Application 14
 - Классы..... 15
 - Свойства 16
 - Методы 16
 - События 18
- Глава 1. Основные понятия VBA..... 19**
 - Базовые конструкции языка Visual Basic..... 19
 - Типы данных в VBA 20
 - Константы и переменные 22
 - Область действия переменных и констант 22
 - Объявление констант и переменных..... 22
 - Начало работы 23
 - Настройка безопасности 25
 - Запись макроса..... 27
 - Имя макроса 27
 - Разработка проекта..... 28
 - Создание модуля..... 31
 - Создание модуля с помощью контекстно-зависимого меню 31
 - Окно кода..... 31
 - Оператор *Option Explicit* 33
 - Первая процедура 33
 - Объявление переменной в VBA 35
 - Оператор *Debug.Print* 36
 - Автоматический ввод атрибутов команд..... 37
 - Структура кода процедуры 38
 - Метод *Active* 39

Объект <i>ActiveCell</i>	40
Открытие книги с макросом	40
Ввод данных	43
Оператор <i>With</i>	43
Свойство <i>Selection</i>	44
Свойство <i>Orientation</i>	45
Объект <i>Range</i>	46
Кнопка (элемент управления <i>ActiveX</i>).....	47
Свойство <i>Offset</i>	49
Функция <i>Environ</i>	49
Функция <i>MsgBox</i>	50
Константы табуляции <i>Chr(9)</i> и <i>Chr(10)</i>	52
Диалоговое окно <i>InputBox</i>	54
Переменная <i>String</i>	54
Переменная <i>Long</i>	55
Константа с указанием типа <i>Long</i>	55
Переменная <i>Byte</i>	56
Глобальные переменные	56
Методы <i>Protect</i> и <i>Unprotect</i>	56
Удаление модуля	58
Глава 2. Основы программирования в VBA	60
Ячейка и диапазон ячеек	60
Арифметические выражения	61
Правила записи арифметических выражений	62
Арифметические выражения в ячейке	62
Арифметические выражения с ячейками.....	63
Обрамление ячейки — свойство <i>BorderAround</i>	63
Синтаксис оператора <i>With</i>	65
Случайные числа <i>RAND</i>	65
Перевод градусов по Фаренгейту в градусы по Цельсию	66
Замена значений формул числом	67
Функция <i>RGB</i>	70
Свойства <i>Color</i> и <i>ColorIndex</i>	71
Свойство <i>Color</i>	71
Свойство <i>ColorIndex</i>	71
Палитра цветов.....	74
Подсчет цветов в рисунке	75
Замена цвета.....	78
Защита ячеек рабочего листа от форматирования	79
Выделение ячеек	81
Выделение серым цветом каждой второй строки	82
Выделение миганием	83
Календарь	85
Подсветка формул	87
Использование клавиатурного сокращения	88
Подсветка минимальных и максимальных значений	90
Цветовая шкала	91

Глава 3. Логические операторы.....	93
Оператор <i>If...Then...Else</i>	93
Неполная форма оператора <i>If...Then</i>	94
Полная форма оператора <i>If...Then...Else</i>	95
Оператор <i>ElseIf</i>	95
Вложенные логические операторы	96
Операторы сравнения.....	96
Оператор <i>If</i> в названии и в теле процедуры	98
Оператор <i>If</i> в теле процедуры.....	99
Примеры использования логических операторов.....	99
Свойство <i>Name</i>	100
Свойство <i>Value</i>	100
Функции <i>IsEmpty</i> и <i>IsNumeric</i>	101
Свойство <i>Range.HasFormula</i>	101
Переход к ячейке A2016.....	102
Свойство <i>Application.Version</i>	103
Альтернативное использование оператора <i>Case</i>	105
Функция <i>InputBox</i>	106
Оператор <i>GoTo</i>	108
Проверка существования файла	108
Глава 4. Операторы цикла	110
Цикл <i>For...To...Step...Next</i>	110
Цикл <i>For...To...Next</i>	111
Заполнение столбца	112
Заполнение столбца с большим шагом	112
Отрицательный шаг	113
Выход из цикла по условию.....	114
Кнопка для запуска макроса (элемент управления формы).....	115
Вложенный цикл <i>For...To...Next</i>	116
Цикл <i>For...Each</i>	118
Цикл <i>Do...While...Loop</i>	119
Цикл <i>Do...Until...Loop</i>	121
Цикл <i>While...Wend</i>	122
Оператор <i>With...End With</i>	123
Измерение времени работы процедуры.....	125
Глава 5. Функции, определенные пользователем	129
Построение функций	129
Построение графика функции одной переменной	129
Структура кода функции пользователя.....	130
Построение графика функции одной переменной (продолжение)	130
Построение графика кусочно-непрерывной функции с двумя условиями	133
Построение графика кусочно-непрерывной функции с тремя условиями.....	134
Названия формул на английском.....	136
Пользовательская функция с тремя переменными	136
Создание собственной категории.....	138
Функция без параметров	140
Переименование рабочего листа	140

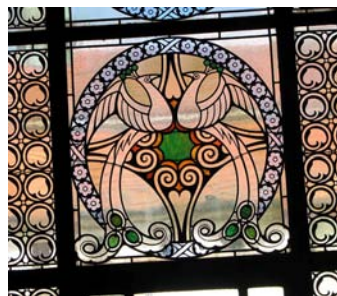
Функция с переменной типа <i>Range</i>	141
Случайное число	142
Функция с массивом	144
Функция с массивом в качестве параметра	145
Вызов функции из процедуры	147
Вызов процедуры, использующей функцию, из процедуры	149
Запись названий формул	150
Вычисление определенного интеграла	152
Метод прямоугольников	153
Метод трапеций	153
Метод Симпсона	153
Переключатели <i>OptionButton</i>	156
Глава 6. Пользовательская форма.....	159
Создание форм средствами VBA	159
Добавление формы <i>UserForm</i> в проект	159
Семейство форм	160
Свойства формы.....	160
Разметочная сетка.....	162
Методы формы.....	163
События формы	163
Командная кнопка для вызова формы	164
Элементы управления.....	166
Префиксы	169
Элемент управления <i>Label</i>	169
Элемент управления <i>CommandButton</i>	172
Элемент управления <i>TextBox</i>	175
Элементы управления <i>OptionButton</i> и <i>Frame</i>	178
Ключевое слово <i>Me</i>	181
Элемент управления <i>ScrollBar</i>	181
Элемент управления <i>ListBox</i>	185
Элемент управления <i>ComboBox</i>	188
Элемент управления <i>Image</i>	190
Элемент управления <i>SpinButton</i>	196
Элемент управления <i>TabStrip</i>	200
Элементы управления <i>CheckBox</i> и <i>MultiPage</i>	203
Элемент управления <i>RefEdit</i>	206
Элемент управления <i>ToggleButton</i>	209
Создание собственного элемента управления	210
Глава 7. Программирование объектов Shape.....	212
Типы объектов семейства Shapes	212
Тип объекта <i>msoShapeRectangle</i> (прямоугольник) с заливкой (<i>Fill</i>)	213
Тип объекта <i>msoConnectorCurve</i> (соединительная линия).....	215
Метод <i>AddConnector</i>	216
Метод <i>Patterned</i>	218
Рисование линии: метод <i>AddLine</i>	219
Тип объекта <i>msoShapeSmileyFace</i>	220
Свойство <i>Name</i>	221

Стрелка	222
Метод <i>FillFormat.OneColorGradient</i>	224
Текстовый фрейм	224
Тип фигуры <i>msoShapeHeart</i> (сердце) с заливкой (<i>Fill</i>)	226
Метод <i>Group</i>	227
Создание выноски с текстовым фреймом	228
Свойство <i>ThreeD</i>	229
Частичное и полное удаление фигур	230
Оператор <i>Set</i>	231
Фракталы	233
Тип данных, определенный пользователем	234
Фракталы из треугольников	234
Фракталы из многоугольников	238
Фракталы из четырехугольников	240
Элементы управления формы	242
Элемент управления <i>Полоса прокрутки</i>	243
Глава 8. Работа с ячейками и областями	246
Объект <i>Application</i>	246
Свойства объекта <i>Application</i>	246
Методы объекта <i>Application</i>	247
Объект <i>Range</i>	248
Адресация ячеек	248
Свойства объекта <i>Range</i>	249
Методы объекта <i>Range</i>	250
Объект <i>Selection</i>	251
Объект <i>Cell</i>	251
Выделение нескольких областей	251
Выделение последней ячейки в диапазоне	252
Нахождение конца диапазона в виде ячейки	254
Выделение ячеек с формулами	255
Выделение рабочей области	256
Форматирование объединенных ячеек	257
Выделение по условию	258
Убираем лишнее	259
Убираем текст	260
Имена и фамилии	262
Создание комментариев	263
Метод <i>Delete</i>	264
Метод <i>Clear</i>	265
Метод <i>Application.Goto</i>	265
Глава 9. Работа с данными	266
Массив из трех элементов	266
Динамический массив данных	268
Сравнение областей на одном листе	270
Сравнение областей на разных листах	271
Сортировка	275
Вертикальная и горизонтальная сортировка	275

Сортировка областей (блоков).....	278
Простая сортировка блоков	278
Сортировка блоков с изменением ее условий	280
Сортировка по цвету.....	281
Контроль автофильтра посредством VBA.....	284
Команда <i>Итоги</i>	285
Глава 10. Автоматизация диаграмм.....	288
Объектная модель диаграмм.....	288
Типы диаграмм	290
Свойства объекта <i>Chart</i>	292
Методы объекта <i>Chart</i>	293
Первая диаграмма.....	293
Создание диаграммы с помощью VBA.....	297
Коническая гистограмма, созданная в VBA.....	299
Печать диаграмм	301
Удаление диаграммы	303
Форматирование параметров диаграммы.....	304
Изменение цвета поверхности.....	306
Добавление линии тренда	308
Форматирование отрицательных значений	311
Убрать/вернуть закрашку	313
Изменение типа диаграммы.....	315
Изменение прозрачности	317
Поясняющие надписи.....	319
Красивые узоры	321
Глава 11. Программирование объектов и событий.....	323
Где и как создаются процедуры обработки событий?.....	324
Процедура для объекта <i>ЭтаКнига</i>	325
События, связанные с кнопками мыши	326
Процедура в модуле.....	326
Событие для объекта <i>Лист1</i>	327
Ключевое слово <i>ByVal</i> и параметр <i>Target</i>	328
Очистка ячейки	329
Свойства <i>ScrollRow</i> и <i>ScrollColumn</i>	330
События активации и деактивации	330
Свойство приложения <i>ActiveWindow</i>	330
Активный лист	331
Число обращений к макросу	332
Управление выделением области.....	333
События <i>Activate</i> и <i>Deactivate</i> рабочего листа	334
Двойной щелчок левой кнопкой мыши	335
Щелчок правой кнопкой мыши	336
Введите пароль.....	336
Событие закрытия книги.....	337
Событие сохранения книги	338

Глава 12. Операторы даты и времени	340
Вывод даты и времени в окно <i>Immediate</i> оператором <i>Debug.Print</i>	340
Печать даты и времени с помощью функции <i>CDate</i>	341
Функции <i>DateSerial</i> и <i>TimeSerial</i>	341
Текущие дата и время.....	342
Текущие дата и время с учетом минут и секунд	342
Функция <i>Weekday</i> — день недели	343
Функция <i>Format</i>	344
Функция <i>DateDiff</i>	346
Функция <i>DatePart</i>	347
Функция <i>WeekdayName</i>	347
Вывод сообщения на 3 секунды	348
Метод <i>Application.OnTime</i>	349
Автоматическое заполнение ячеек датами методом <i>AutoFill</i>	350
Подсветка даты	352
Поиск даты	353
Календарь	355
Календарь по месяцам	358
Календарь по неделям	361
Определение возраста	364
 Глава 13. Работа с рабочей книгой	 366
Свойства объекта <i>Workbook</i>	366
Методы объекта <i>Workbook</i>	367
Синтаксис открытия рабочей книги.....	367
Открытие рабочей книги.....	367
Вызов диалогового окна <i>Открытие документа</i>	369
Открытие рабочей книги в диалоговом окне	369
Открытие Блокнота	371
Свойство <i>Workbook.Name</i>	373
Создание рабочей книги.....	373
Имя приложения	374
Сохранение рабочей книги	376
Метод <i>Save</i>	376
Метод <i>Save As</i>	376
Сохранение копированием.....	377
Сохранение всех книг и выход	377
Сохранение всех книг и выход по запросу	378
Объект <i>Worksheet</i>	379
Свойства объекта <i>Worksheet</i>	379
Методы объекта <i>Worksheet</i> и семейства <i>Worksheets</i>	379
Деление рабочего листа на страницы для печати	380
 Глава 14. Файловые операции.....	 382
Формат файлов.....	382
Метод <i>CreateTextFile</i>	383
Список файлов указанной папки	384
Режим доступа <i>Input/Output</i>	385
Файлы из <i>Application.AddIns</i>	387

Объект <i>FileDialog</i>	388
Функция <i>GetAttr</i>	389
Документирование информации о файле	390
Проверка существования файла	392
Команда <i>Kill</i> для удаления файла	393
Команда <i>FileCopy</i> для копирования файла	394
Переименование файла	395
Перемещение файла	395
Глава 15. Отладка программ и сообщения об ошибках	397
Возникновение ошибок	397
Выявление и исправление ошибок	397
Три окна для просмотра ошибок	399
Окно просмотра	399
Локальное окно	401
Окно наблюдения	402
Объект <i>Err</i>	405
Оператор <i>On Error</i>	405
Оператор <i>On Error Resume Next</i>	406
Оператор <i>On Error GoTo</i> : вариант 1	407
Оператор <i>On Error GoTo</i> : вариант 2	407
Константы <i>xlDisabled</i> и <i>xlInterrupt</i>	409
Массив листов	409
Команда меню <i>Debug</i>	410
Глава 16. Программирование связей	412
Гиперссылки	412
Кнопка гиперссылки	414
Передача данных из Excel в Word	415
Передача данных из Word в Excel	416
Передача данных из Excel в PowerPoint	417
Передача данных из PowerPoint в Excel	419
Приложение. Описание электронного архива	423
Предметный указатель	425



ГЛАВА 1

Основные понятия VBA

Базовые конструкции языка Visual Basic

Начиная изучение языка программирования *Microsoft Visual Basic for Applications*, рассмотрим сначала такое понятие, как язык. Язык — это знаковая система, предназначенная для хранения и передачи информации. Причем в нашем случае, не умаляя общности, можно говорить о том, что не имеет значения, естественный это язык или язык программирования (а может, математический или язык жестов).

Безусловно, языки различаются между собой, однако большинству языков присущи общие закономерности и системная организация. Для каждого языка определены следующие понятия:

- ◆ *алфавит*, определяющий допустимые символы языка;
- ◆ *лексика* — словарь языка и способы образования слов из символов;
- ◆ *синтаксис* — способы и правила записи и соединения слов в предложения. В VBA это правила записи лексем языка: констант, имен переменных, массивов, ключевых слов, строк и др., а также способы соединения слов пользователя, знаков операций и ключевых слов VBA в различные программные конструкции;
- ◆ *семантика* — значения отдельных слов, словосочетаний и предложений. В VBA этот термин определяет значение и логический смысл выражений и записываемых пользователем программных конструкций;
- ◆ *прагматика* — назначение и область применения языка.

Как и все языки программирования высокого уровня, Visual Basic имеет свой набор допустимых для использования символов — алфавит. Алфавит VBA содержит следующие группы символов:

- ◆ *буквы*: заглавные и строчные буквы латинского алфавита и кириллицы: A, B, ..., Z, a, b, ... , z, А, Б, ..., Я, а, б, ... , я;
- ◆ *арабские цифры*: 0, 1, ... , 9;

◆ *специальные символы:*

+|-|*|/|\|'|"|.|,|=|>|<|
|[|]|(|)|:|{|}|&|@|%|\$|!|;

◆ *разделители:* пробел; символ продолжения строки (_); символ табуляции; символ новой строки.

Из символов алфавита конструируются все структуры языка. В том числе составные символы (<= | <> | >= |); ключевые слова (Dim, As, Binary и т. п.); имена констант, переменных, процедур, функций и т. п.

Заглавные и строчные буквы в записи имен VBA не различаются. Транслятор запоминает формат первой записи идентификатора в программе и при повторном использовании идентификатора в тексте программы он автоматически преобразуется к первоначальному формату.

Типы данных в VBA

Тип данных программного элемента определяет данные, которые могут содержаться в нем, и способы их хранения. Типы данных применяются ко всем значениям, которые могут храниться в памяти компьютера или участвовать в вычислении выражения.

Каждая переменная, литерал, константа, перечисление, свойство, параметр процедуры, аргумент процедуры и возвращаемое значение процедуры имеют тип данных.

Переменные подразделяются на *простые* и *индексированные* (переменные с индексом). Индексированными переменными являются элементы массивов. Все подробности о массивах будут рассмотрены позднее.

Все слова, задействованные в конструкциях языка, являются зарезервированными словами и не могут применяться для других целей.

В языке программирования Visual Basic версии для Microsoft Excel 2016 определены различные типы данных (табл. 1.1), используемые как напрямую, так и в универсальных типах данных (например Decimal в Variant): Boolean, Byte, Currency, Date, Double, Decimal, Integer, Long\LongLong, LongPtr, Object, Single, String, Variant, а также типы данных, определяемые пользователем так же, как некоторые типы объектов.

Таблица 1.1. Типы данных, используемые в программах на Visual Basic для Excel 2016

Тип	Занимаемый объем памяти	Диапазон значений
Boolean	2 байта	Логическая величина, имеющая значения True (Истина) или False (Ложь)
Byte	1 байт	Целое число без знака в диапазоне от 0 до 255
Currency (денежный)	8 байтов	Число с фиксированной десятичной точкой от -922 337 203 685 477,5808 до 922 337 203 685 477,5807

Константы и переменные

В VBA, как и в других языках программирования высокого уровня, для хранения значений предусмотрены переменные и константы.

- ♦ *Константа* — именованная область памяти, используемая для хранения фиксированного значения, не изменяемого при выполнении программы.
- ♦ *Переменная* — именованная область памяти, используемая для хранения значения, которое можно изменить при выполнении программы.

Область действия переменных и констант

Область действия переменных и констант определяется с помощью ключевых слов: `Private`, `Public` и `Static`.

- ♦ `Private` — определяет область действия в пределах конкретного модуля программы. После завершения выполнения модуля память, отведенная под эти переменные, освобождается. Переменные и константы, объявленные как `Private`, сохраняют свои значения только во время выполнения блока кода, в котором они определены.
- ♦ `Public` — область действия в пределах приложения. Переменные и константы, объявленные как `Public`, сохраняют свои значения до конца выполнения программы.
- ♦ `Static` — область действия в пределах конкретного модуля и внешних процедур, задействованных в этом модуле. После завершения выполнения модуля значения этих переменных сохраняются между вызовами процедур и могут быть использованы при повторном выполнении модуля.

Переменные и константы, объявленные без ключевых слов `Public`, `Private`, `Static`, сохраняют свое значение согласно месту их объявления (описания).

Переменные и константы, в зависимости от области действия, подразделяются на глобальные и локальные.

- ♦ Если переменная или константа объявлена внутри процедуры, то она является *локальной*, т. е. она определена и доступна только в пределах данной процедуры.
- ♦ Если переменная или константа объявлена вне процедуры, то она будет *глобальной*. Такая переменная или константа может быть доступна в нескольких процедурах.

Объявление констант и переменных

В VBA допускаются два способа включения идентификаторов в текст программы.

- ♦ Во-первых, это неявное объявление нового объекта при записи ранее не использовавшегося в тексте идентификатора в каком-либо выражении.
- ♦ Во-вторых, явное объявление, в котором все новые идентификаторы объявляются (описываются) явно с помощью ключевых слов `Private`, `Public`, `Static`, `Dim`, а для того, чтобы явно указать тип переменной, либо указывается ключевое

слово `As`, либо в имя переменной включается символ объявления типа (см. табл. 1.1).

Примеры описания простых переменных:

```
Private X# 'тип переменной X# объявляется как Double
Public i As Integer, r As Long, c As Date
Static Строка As String
Dim Y 'переменная Y объявлена, ее тип по умолчанию Variant
```

Константы объявляются с помощью ключевого слова `Const`. При этом используемой константе можно указать тип, область действия и присвоить значение.

Синтаксис объявления констант:

```
[{Private|Public|Static}] Const <имя> [As <тип>] = <значение>
```

Если в константе явно не указан тип данных, то VBA назначает ей тип, соответствующий значению выражения.

Примеры:

```
Private Const q = 44,55
Public Const pi = 3,1459
Static Const QWER = 2,54
Const Y = 34
Const Con As Byte = 34
Const Z As Single = -3,8374E-22
```

В объявлениях строковых констант их значения указываются в двойных кавычках, например:

```
Const prv As String = "Язык программирования VBA".
Public Const prv = "Язык программирования VBA".
```

Начало работы

Запустите программу Microsoft Excel 2016 — самую мощную электронную таблицу, разработанную для использования в среде Windows (рис. 1.1).

Откройте знакомую вам книгу или создайте пустую книгу. Вы попадете в привычный мир столбцов, строк и ячеек (рис. 1.2). Из рисунка сразу понятно, как переходить в среду VBA, — видна кнопка **Visual Basic**.

Если же вы всерьез решили заниматься программированием в среде VBA, то вызовите на ленту вкладку **Разработчик**. Обычно при запуске программы Microsoft Excel 2016 этой вкладки на ленте нет.

1. Подведите курсор к команде **Файл** — появятся все команды, доступные при работе с файлами. Среди команд выберите команду **Файл | Параметры** — откроется диалоговое окно **Параметры Excel**.
2. В диалоговом окне **Параметры Excel** необходимо выбрать раздел **Настроить ленту** и активизировать вкладку **Разработчик** (рис. 1.3). Теперь эта вкладка всегда будет на экране.

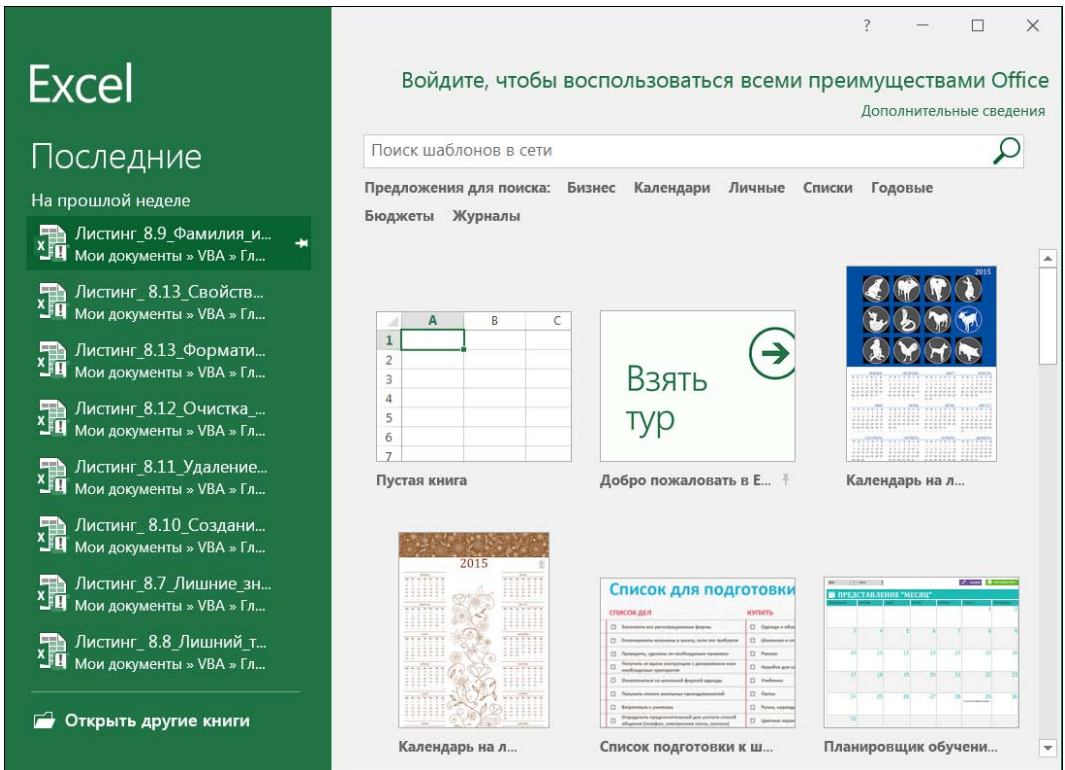


Рис. 1.1. Начало работы в Microsoft Excel 2016

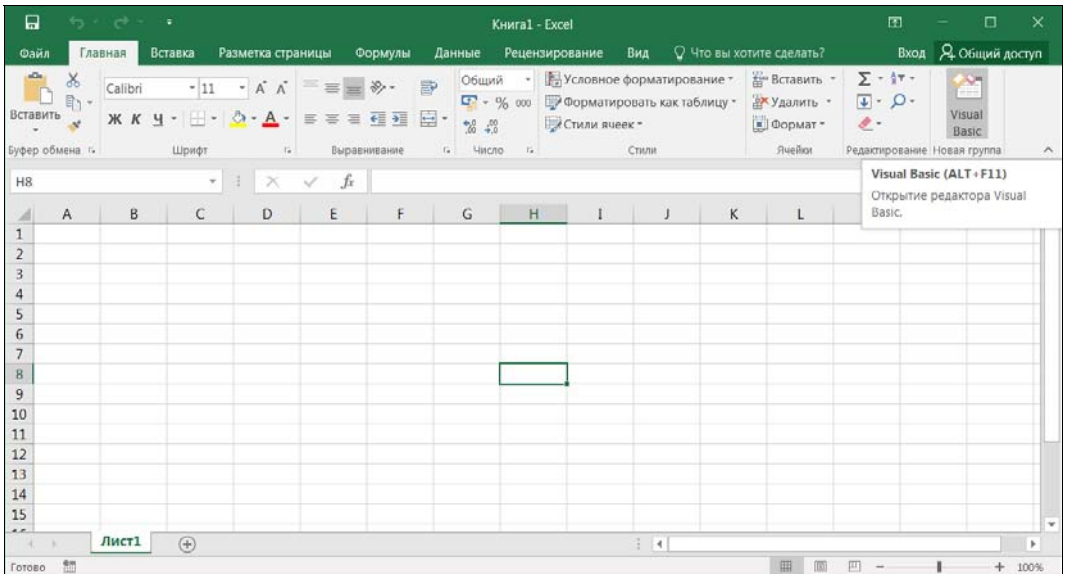


Рис. 1.2. Интерфейс программы Microsoft Excel 2016

- **Отключить все макросы без уведомления;**
- **Отключить все макросы с уведомлением;**
- **Отключить все макросы без цифровых подписей;**
- **Включить все макросы (не рекомендуется, возможен запуск опасной программы).**

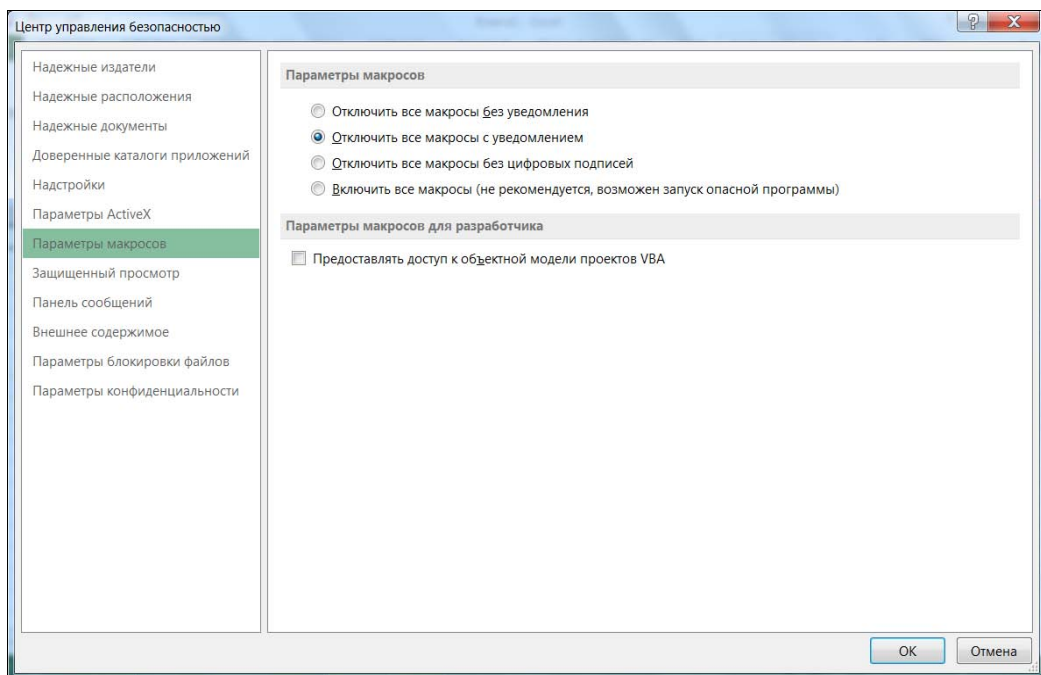


Рис. 1.5. Диалоговое окно Центр управления безопасностью

По умолчанию выбран первый вариант. Если файл сохранен с поддержкой макросов, то выберите второй вариант. В этом случае на экране под лентой появляется предупреждение Системы безопасности (рис. 1.6), предлагающее включить отключенное содержимое, если в этом есть необходимость.

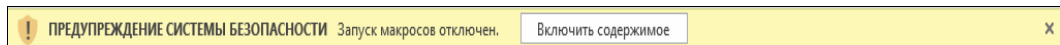


Рис. 1.6. Предупреждение Системы безопасности

В диалоговом окне **Центр управления безопасностью** в разделе **Параметры макросов** есть еще раздел **Параметры макросов для разработчика**, в котором можно включить или отключить флажок **Предоставлять доступ к объектной модели проектов VBA**.

Чтобы получить свою цифровую подпись (для приобретения коммерческого сертификата), необходимо обратиться к соответствующим коммерческим организациям.

Как можно видеть, в диалоговом окне **Параметры Excel**, кроме прочих, имеется также вкладка **Центр управления безопасностью** (рис. 1.7).

С помощью этого диалогового окна можно решить вопросы обеспечения безопасности документа, а также защиты и работоспособности компьютера.

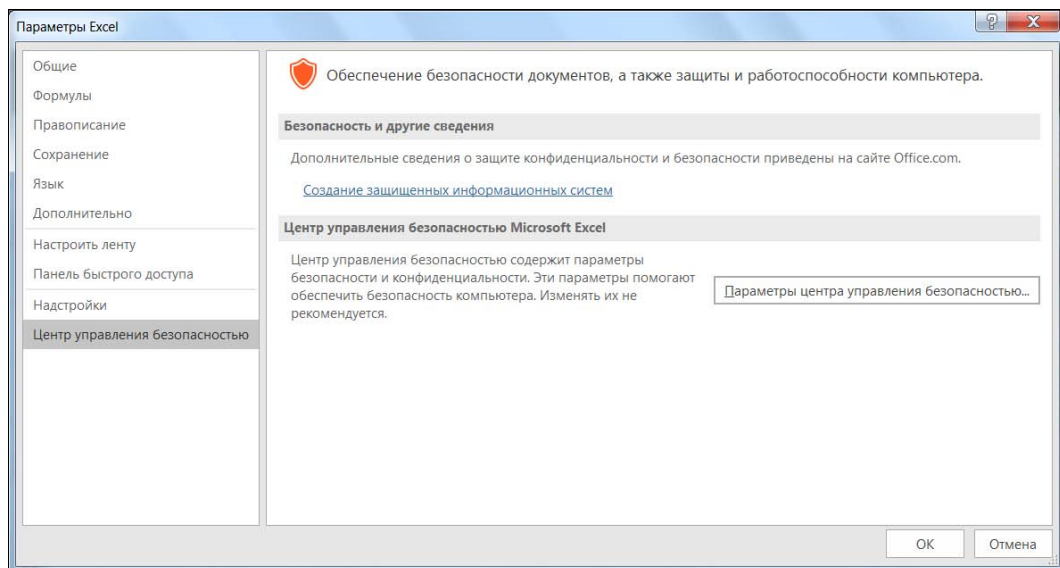


Рис. 1.7. Диалоговое окно **Параметры Excel**, вкладка **Центр управления безопасностью**

Запись макроса

Что же такое макрос? Самый простой ответ — это запись некоторых команд, т. е. небольшая программа, которая выполняется при ее вызове или нажатии соответствующего ей значка. Иными словами, макрос — программный алгоритм действий, записанный пользователем. Программа Microsoft Excel позволяет записывать макросы. В буквальном смысле слова, как при работе с магнитофоном, — нажали кнопку "Запись" и говорите в микрофон, произносите речь — она записывается. Нажали кнопку "Стоп" — запись прекратилась. А теперь можете слушать свою запись сколько угодно раз. И так же, как и к записи на магнитофоне, в программе Microsoft Excel к записанному макросу можно обращаться неоднократно.

Все ваши действия при записи макроса запишутся в виде последовательности команд, а по завершении записи ее можно будет просмотреть в модуле.

Имя макроса

Имя (идентификатор) константы или переменной — это произвольная последовательность символов, начинающаяся с буквы и удовлетворяющая следующим правилам:

- ◆ в составе имени нельзя использовать символы: . (точка), !, @, &, \$, # , пробел;
- ◆ в качестве имен переменных и констант недопустимы ключевые (зарезервированные) слова, входящие в конструкции языка VBA;
- ◆ длина имени не может быть более 255 символов;
- ◆ имя нельзя повторно объявлять в пределах области его видимости (действия);
- ◆ в имени может присутствовать символ подчеркивания;
- ◆ в конце имени можно указать один из символов объявления типа (табл. 1.2). Включение такого символа в имя заменяет в тексте программы запись оператора объявления типа.

Таблица 1.2. Символы, определяющие тип переменных

Тип данных	Символ объявления типа
Integer	%
Long	&
Single	!
Double	#
Currency	@
String	\$

Разработка проекта

Начните знакомиться со средствами визуализации разработки проектов в VBA.

1. Запустите программу Microsoft Excel 2016.
2. Сохраните вновь созданную книгу с поддержкой макросов под именем **Старт.xlsm**. Для этого необходимо выбрать команду **Файл | Сохранить как** и в диалоговом окне **Сохранение документа** в раскрывающемся списке **Тип файла** выбрать вариант сохранения файла: **Книга Excel с поддержкой макросов** (рис. 1.8).
3. Перейдите в среду VBA, выбрав на вкладке ленты **Разработчик** раздел **Visual Basic**. На экране откроется окно интегрированной среды разработки приложений **Microsoft Visual Basic for Applications – Название книги – Активная компонента**. Можно для этого воспользоваться и клавиатурным сокращением, одновременно нажав клавиши <Alt>+<F11>.

ПРИМЕЧАНИЕ

Самый быстрый переход в среду Visual Basic for Applications — нажатие комбинации клавиш <Alt>+<F11>.

4. Для знакомства с приложением и объектами VBA нажмите клавишу <Alt> или воспользуйтесь командой **View | Object Browser** (Вид | Просмотр объектов).

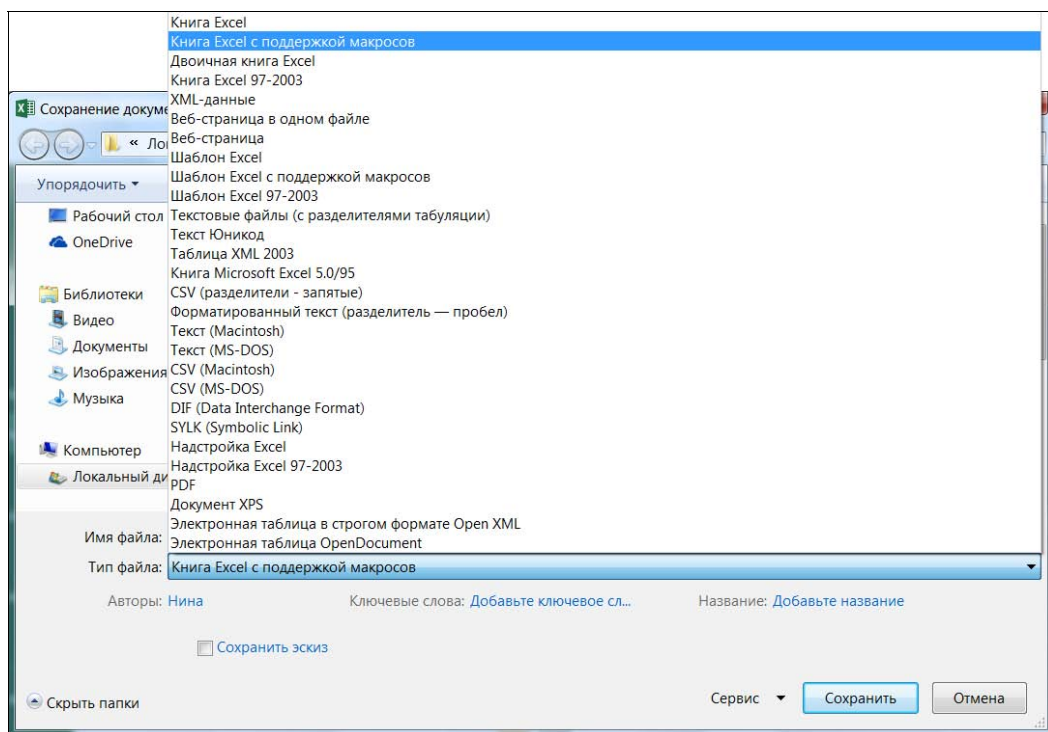


Рис. 1.8. Выбор варианта сохранения книги

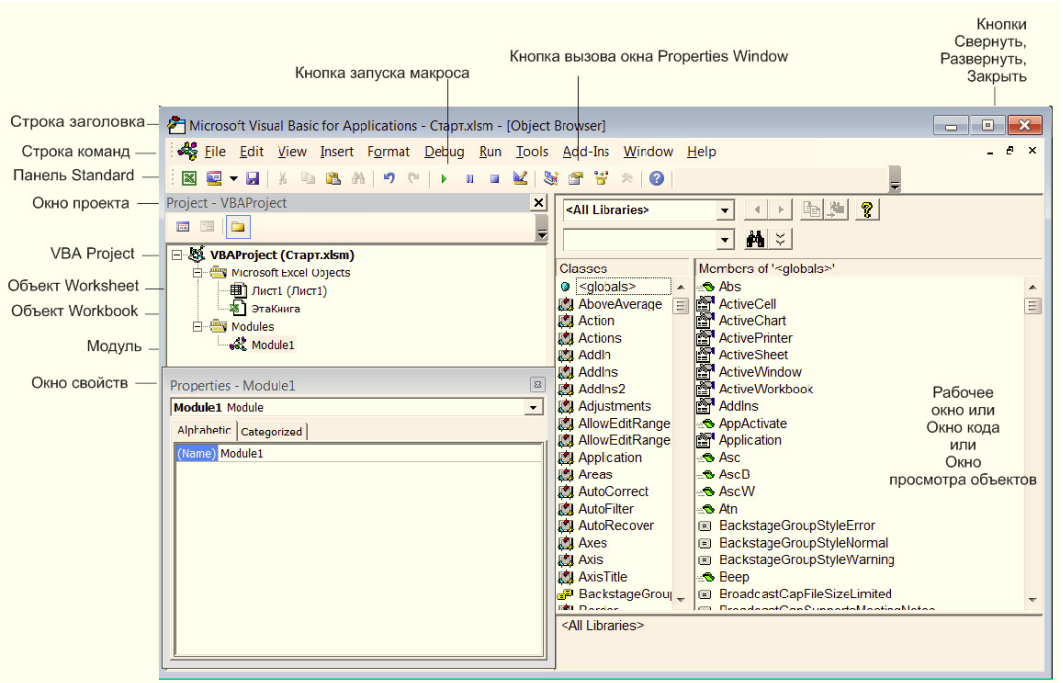
5. В раскрывающемся списке **Project/Library** выберите библиотеку **Excel**.
6. В списке **Classes** выберите пункт **Application**. На экране откроется окно **Application** с перечислением объектов (рис. 1.9, а). Если вставить модуль при помощи команды меню **Insert | Module** (Вставить | Модуль), то он будет называться **Module1**. Переименовать модуль в **Мой_МАКРОС** (рис. 1.9, б) можно, изменив его название в поле **(Name)** (Имя) в окне **Properties** (Свойства).

На рис. 1.9, а можно увидеть и другие компоненты редактора VBA:

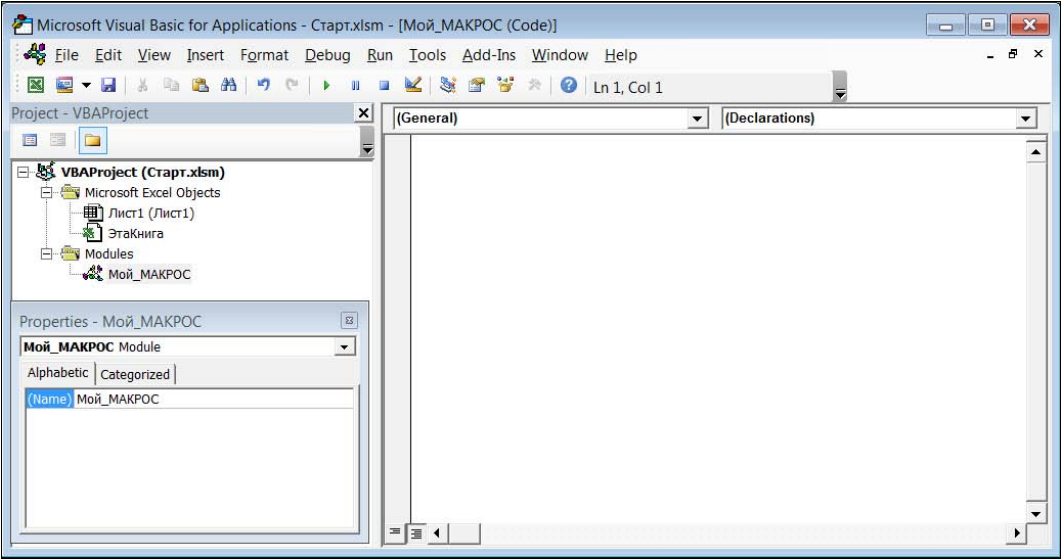
- ◆ окно проекта **Project - VBAProject**;
- ◆ окно свойств **Properties**;
- ◆ рабочее окно (окно кода, окно форм **UserForm**, окно просмотра объектов);
- ◆ панели инструментов.

В нашем случае в окне просмотра объектов видны все объекты класса **Application** (Приложение): **ActiveWorkbook** (Активная рабочая книга), **ActiveSheet** (Активный лист), **ActiveCell** (Активная ячейка), **ActiveChart** (Активная диаграмма), **ActivePrinter** (Активный принтер) и др.

Кроме того, можно увидеть иерархически построенную систему объектов. Эти объекты составляют как собственно Microsoft Excel, так и задают методы и свойства.




а




б

Рис. 1.9. а — окно проекта; б — пример переименования модуля

Окно проекта **Project - VBAProject** показывает структуру проекта (файла). Это окно активизируется в редакторе VBA выбором команды **View | Project Explorer**, или кнопкой  **Project Explorer**, или нажатием комбинации клавиш <Ctrl>+<R>.

Окно кода предназначено для хранения кода, связанного с объектом. У каждого объекта есть свой код, расположенный в окне кода, — так, каждый рабочий лист **Worksheet** (Лист1) имеет свой код, и рабочая книга **Workbook** (ЭтаКнига) — свой код.

Если что-то не ясно, всегда можно воспользоваться справкой **Help** или кнопкой  — как непосредственно в программе, так и в режиме онлайн.

Создание модуля

Макросы или пользовательские функции создаются в виде *модулей*. Вообще, модуль — это часть программы, оформленная в виде, допускающем ее независимую трансляцию.

Модуль добавляется к проекту (рис. 1.10) с помощью команды **Insert | Module** (Вставить | Модуль), и в окне проекта он отобразится на уровне вашего приложения.

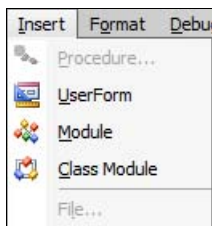


Рис. 1.10. Выбор команды **Insert | Module** из меню

Создание модуля с помощью контекстно-зависимого меню

Создать модуль можно и с помощью команды **Insert | Module** (Вставить | Модуль), выбрав ее из контекстно-зависимого меню, появляющегося при щелчке правой кнопкой мыши по объекту **ЭтаКнига** в окне проекта **Project - VBAProject** (рис. 1.11).

Окно кода

Как только модуль создан, появляется окно кода модуля (рис. 1.12).

Чтобы посмотреть в этом окне свойства модуля или рабочего листа, следует нажать клавишу <F4>. На рис. 1.12, кроме того, можно увидеть иерархическое строение объектов в окне проекта **Project - VBAProject** и окно свойств объекта **Properties - Module1**.

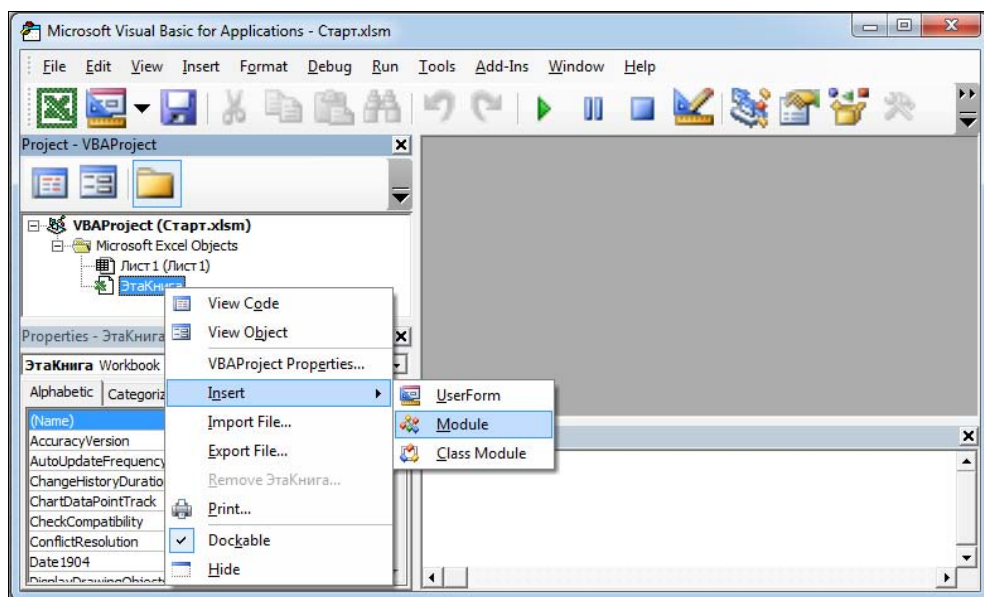


Рис. 1.11. Выбор команды **Insert | Module** из контекстно-зависимого меню

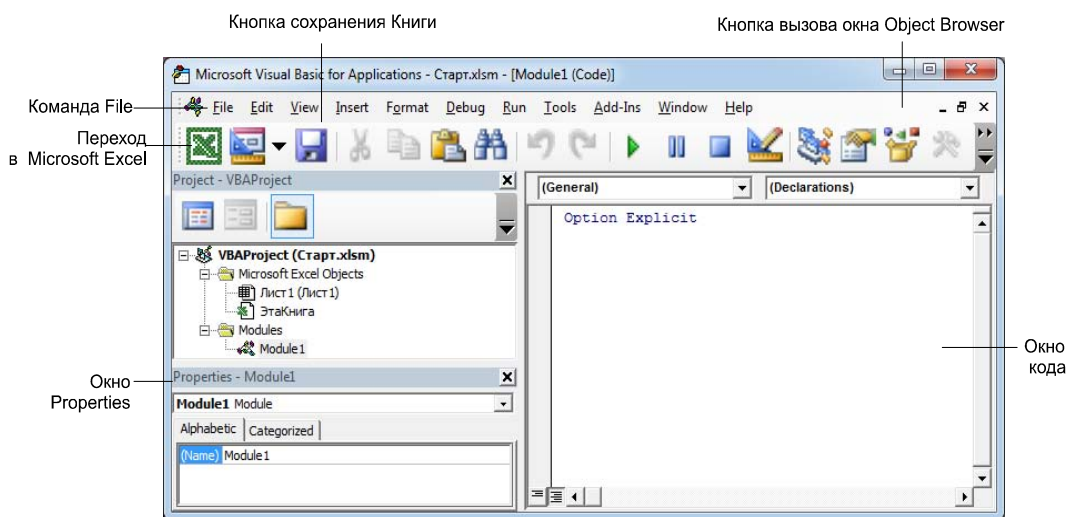


Рис. 1.12. Окно кода модуля

В окне кода в верхней строке расположены раскрывающиеся списки: справа — **General**, слева — раздел объявлений **Declarations**. В списке **General** перечислены текущие объекты в зависимости от того, какой элемент выбран в окне рабочего проекта **Project - VBAProject**. В списке **Declarations** перечислены доступные для данного объекта события, при выборе которых в окне кода автоматически появятся заготовки данных процедур-обработчиков событий с параметрами по умолчанию.

Оператор *Option Explicit*

Оператор `Option Explicit` необходимо поместить в окне кода в самом начале перед последующими процедурами и функциями для обязательного объявления всех переменных. Использование этого оператора не допускает возможности неправильного ввода имени переменной, которая задействована в одной или нескольких процедурах модуля. Например, если переменная была объявлена как `НАЛОГ`, а в коде при наборе вместо русской буквы `о` была набрана латинская буква `o`, то это приведет к ошибке. Без оператора `Option Explicit` подобную ошибку было бы трудно отследить.

ПРИМЕЧАНИЕ

Если после создания модуля оператор `Option Explicit` в самом его начале не появляется, то, чтобы не дописывать его самостоятельно каждый раз при создании модуля, вызовите диалоговое окно параметров VBA при помощи команды меню **Tools | Options** (Сервис | Параметры) и на вкладке **Editor** (Редактор) установите флажок **Require Variable Declaration** (Требовать объявление переменных).

Первая процедура

Процедура — это минимальная семантически законченная программная конструкция, допускающая выполнение. В конце концов, просто так операторы не выполняются и не пишутся, они находятся в описании процедур и функций. Функция отличается от процедуры тем, что помимо выполнения операторов она возвращает некоторое значение.

1. Запустите программу Microsoft Excel 2016.
2. Перейдите в среду VBA, нажав на пиктограмму **Visual Basic**, расположенную в самом конце справа на ленте **Главная** либо в самом начале слева на ленте **Разработчик**. На экране откроется окно интегрированной среды разработки приложений **Microsoft Visual Basic. Книга 1**. Можно также воспользоваться и сочетанием клавиш `<Alt>+<F11>`.
3. Добавьте к проекту модуль с помощью команды **Insert | Module** (Вставить | Модуль).
4. Создайте первую процедуру. Для этого выполните команду **Insert | Procedure** (Вставить | Процедура) — откроется диалоговое окно **Add Procedure** (Добавить процедуру), показанное на рис. 1.13.

В поле **Name** (Имя) введите имя создаваемой процедуры, например, `Первая_процедура`. Имя процедуры может быть написано как русскими, так и латинскими буквами, но должно состоять из одного слова, поэтому в нашем примере два слова соединены подчеркиванием. Очень важно, чтобы имя функции не совпадало с номером какой-либо ячейки на рабочем листе. Нарушение этого условия приводит к появлению ошибок при выполнении процедур и функций. В настоящее время в последних версиях программы номера ячеек меняются от `A1` до `XFD1048576`.