

УДК 004.67
ББК 32.973.26-018.2
М15

Mike McGrath
Excel VBA in Easy Steps

Copyright © 2021 by Mike McGrath

Translated and reprinted under a license agreement from the Publisher: In Easy Steps, 16
Hamilton Terrace, Holly Walk, Leamington Spa, Warwickshire, U.K. CV32 4LY



МакГрат, Майк.
М15 Excel VBA. Стань продвинутым пользователем за неделю / Майк МакГрат ; [перевод с английского М.А. Райтмана]. — Москва : Эксмо, 2023. — 240 с. : ил. — (Excel для всех).

ISBN 978-5-04-121944-4

Пошаговый самоучитель по языку VBA, при помощи которого создаются макросы для Excel, поможет вам стать продвинутым пользователем и повысить свою эффективность работы в этой программе в несколько раз. Книга снабжена множеством иллюстраций, а вся теория объясняется на доступных даже для полных новичков примерах. Внутри вы найдете полезные советы, предостережения и сможете скачать архив с бесплатными примерами для работы с ними на компьютере.

УДК 004.67
ББК 32.973.26-018.2

ISBN 978-5-04-121944-4

© Райтман М.А., перевод на русский язык, 2022
© Оформление. ООО «Издательство «Эксмо», 2023

1

Первые шаги

9

Знакомство с Excel VBA	10
Запись макроса	12
Просмотр кода макроса	15
Тестирование макроса	17
Изменение кода макроса	18
Ссылки в Excel	19
Сохранение макросов	22
Безопасность при работе с макросами	25
Заключение	27

2

Написание макросов

29

Обзор редактора	
Visual Basic	30
Создание макроса	32
Настройка панели быстрого доступа	35
Добавление элементов управления формы	37
Определение иерархии	40
Определение диапазона	42
Адресация ячеек	44
Заключение	47

3

Хранение значений

49

Создание переменных	50
Определение типов данных	52
Управление строками	54
Работа с массивами	56
Описание измерений	59
Представление объектов	61
Объявление констант	64
Заключение	67

4**Выполнение операций****69**

Арифметические операторы	70
Операторы сравнения	72
Логические операторы	74
Объединение строк	76
Приоритеты операций	78
Заключение	81

5**Создание инструкций****83**

Управление ветвями	84
Альтернативное ветвление	86
Выбор ветвей	89
Управление циклами	92
Выполнение циклов	94
Прерывание циклов	97
Итерирование циклов	99
Оператор with	101
Заключение	104

6**Выполнение процедур****107**

Вызов подпрограмм	108
Изменение области видимости	110
Передача аргументов	112
Добавление модулей	114
Сохранение значений	117
Отладка кода	119
Обработка ошибок	122
Заключение	125

7

Использование функций**127**

Определение функции	128
Вызов функции	130
Область видимости функции	132
Передача массива аргументов	135
Определение параметров	137
Возвращение ошибок	139
Отладка функций	142
Описание функций	144
Заключение	146

8

Распознавание событий**149**

Создание обработчиков событий	150
События открытия книги	152
События изменения книги	155
События закрытия книги	157
Выявление изменений книги	159
Обработка изменений книги	162
Перехват нажатий клавиш	164
Отслеживание времени	166
Заключение	169

9

Отображение диалоговых окон**171**

Запрос ввода	172
Отображение сообщений	174
Импортирование файлов	176
Сохранение файлов	178
Создание форм	180
Выполнение команд на ленте	183
Заключение	185

10**Добавление пользовательских форм 187**

Вставка пользовательских форм	188
Добавление элементов управления формы	190
Сравнение элементов формы	192
Изменение свойств	195
Присваивание имен элементам формы	197
Отображение форм	199
Обработка событий формы	202
Использование списков	204
Заключение	207

11**Разработка приложений 209**

Игнорирование режимов	210
Индикация прогресса	212
Управление элементами MultiPage	215
Создание вкладок с данными	217
Отображение диаграмм	220
Создание надстроек	222
Установка надстроек	225
Добавление кнопок на ленту программы	227
Заключение	229

Алфавитный указатель 231

1

Первые шаги

Добро пожаловать в увлекательный мир Excel VBA (Visual Basic for Applications). В этой главе вы узнаете, как создать макрос VBA для книг Excel.

- 10 Знакомство с Excel VBA
- 12 Запись макроса
- 15 Просмотр кода макроса
- 17 Тестирование макроса
- 18 Изменение кода макроса
- 19 Ссылки в Excel
- 22 Сохранение макросов
- 25 Безопасность при работе с макросами
- 27 Заключение

Знакомство с Excel VBA

Visual Basic for Applications (VBA) — это язык программирования, встроенный в электронные таблицы Excel и остальные продукты Microsoft Office. С помощью VBA можно решить множество задач, с которыми не справятся стандартные инструменты Excel. Также VBA автоматизирует многие рутинные процессы.



Если вы только начинаете работу с Excel, ознакомьтесь с нашей книгой «Excel 2019 in easy steps».

Мы рекомендуем заниматься по этой книге продвинутым пользователям Excel, которые умеют:

- создавать книги и добавлять листы;
- искать материалы в книге и на листе;
- использовать ленточный интерфейс;
- именовать ячейки и диапазоны;
- использовать функции рабочего листа.

Для демонстрации примеров использовалась версия программы Excel 2019. Многие из них также актуальны и для более ранних версий Excel.

Включение VBA

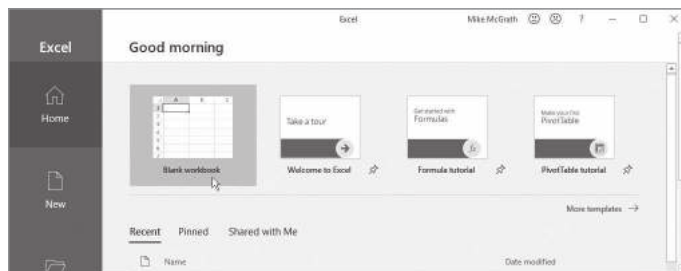
Перед началом работы вам нужно включить VBA в настройках Excel:



Все примеры из этой книги вы можете бесплатно скачать по ссылке: http://addons.eksmo.ru/it/excelvba_examples.zip.

1

Откройте Excel и выберите пункт Blank workbook (Пустая книга).



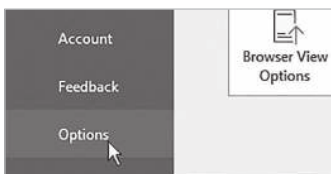
2

Когда книга откроется, нажмите вкладку **File** (Файл) на ленточном интерфейсе.



3

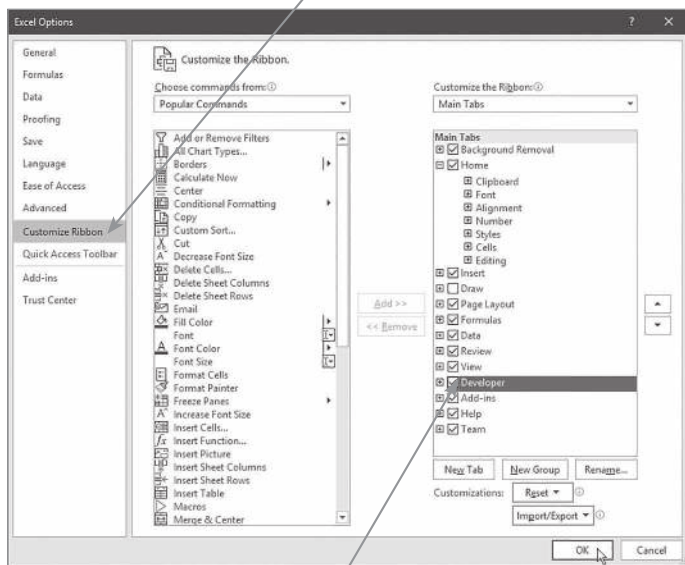
Выберите строку **Options** (Параметры). Откроется диалоговое окно **Excel Options** (Параметры Excel).



Вы можете также открыть диалоговое окно, нажав сочетание клавиш **Alt+F+T**.

4

Выберите пункт **Customize Ribbon** (Настроить ленту) в левом столбце.



В диалоговом окне параметров напротив строки **Developer** (Разработчик) вы можете щелкнуть мышью по значку + и раскрыть группу элементов. Если вы щелкнете правой кнопкой мыши по любой из групп элементов, контекстное меню предложит вам параметры для изменения групп, которые будут отображаться на панели **Developer** (Разработчик).

5

Проверьте, включено ли отображение вкладки **Developer** (Разработчик) в правом столбце.

6

Нажмите кнопку **ОК**, чтобы принять изменения и закрыть диалоговое окно.





В диалоговом окне **Record Macro** (Запись макроса) вы можете добавить описание макроса.



Макрос — это набор инструкций программирования, которые хранятся в коде VBA.

- 7 Теперь панель **Developer** (Разработчик) отображается на ленточной панели.



- 8 Перейдите на вкладку **Developer** (Разработчик). В группе элементов **Code** (Код) вы увидите кнопку **Visual Basic** — теперь VBA активирован.

Запись макроса

Активировав VBA способом, указанным в предыдущем разделе, вы можете создать простое приложение посредством записи «макроса» для сохранения действий:

- 1 В Excel откройте пустую книгу, затем выберите ячейку A1.

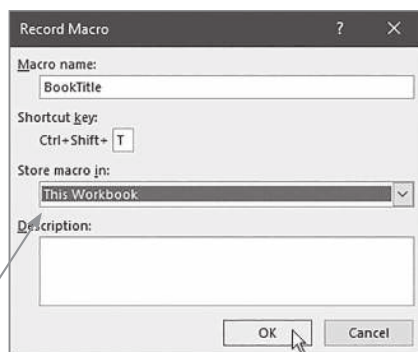


- 2 На панели **Developer** (Разработчик) нажмите кнопку **Record Macro** (Запись макроса) в группе элементов **Code** (Код). Откроется диалоговое окно **Record Macro** (Запись макроса).

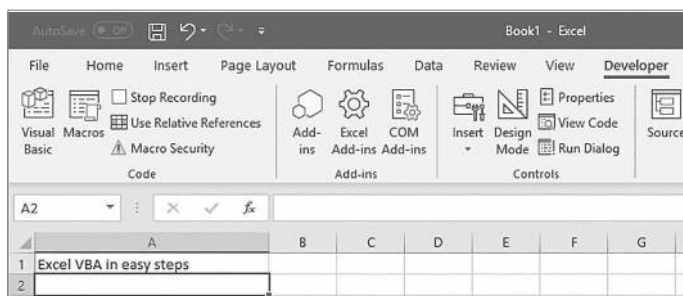


- 3 Напишите любое название в поле **Macro name** (Имя макроса), например, **BookTitle**.

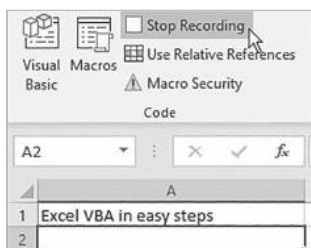
- 4 Затем напишите букву в поле **Shortcut key** (Сочетание клавиш), например, **T**. Таким образом вы назначите макросу сочетание клавиш **Ctrl + Shift + T**.



- 5 В раскрывающемся списке ниже выберите пункт **This Workbook** (Эта книга).
- 6 Нажмите клавишу **ОК**, чтобы закрыть диалоговое окно, и начните запись действий.
- 7 Напечатайте название этой книги в ранее выбранную ячейку **A1**, затем нажмите клавишу **Enter**, чтобы сохранить название в ячейке.



- 8 Обратите внимание, что фокус сместился, — после того, как вы нажали клавишу **Enter**, ячейка **A2** была выбрана автоматически.



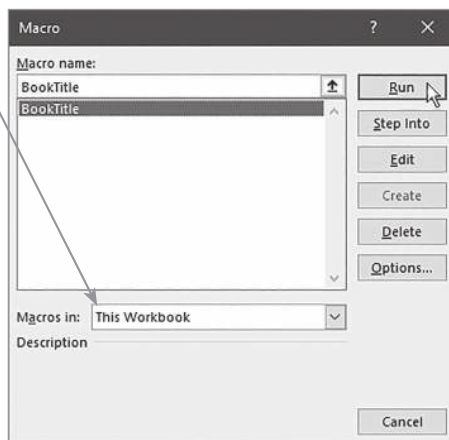
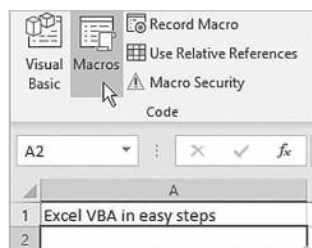
Когда вы начнете запись, кнопка **Record Macro** (Запись макроса) изменится на **Stop Recording** (Остановить запись).



Чтобы открыть диалоговое окно макроса, вы можете использовать сочетание клавиш **Alt + F8**.

9 Теперь нажмите кнопку **Stop Recording** (Остановить запись), чтобы прекратить запись действий.

10 Нажмите кнопку **Macros** (Макросы) в группе элементов **Code** (Код). Откроется диалоговое окно **Macro** (Макросы), в котором перечислены макросы, сохраненные в **This Workbook** (Эта книга).

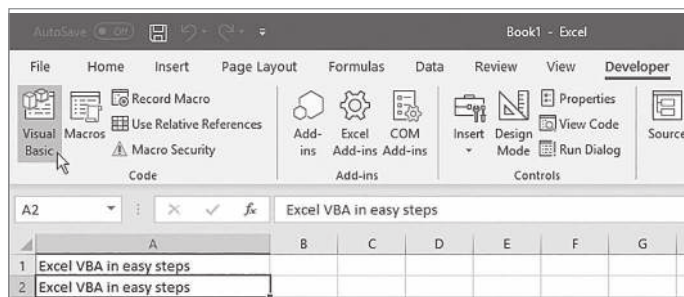


11 Выберите макрос **BookTitle**, затем нажмите кнопку **Run** (Выполнить) для выполнения макроса. Название книги автоматически появится в ячейке **A2**.

Просмотр кода макроса

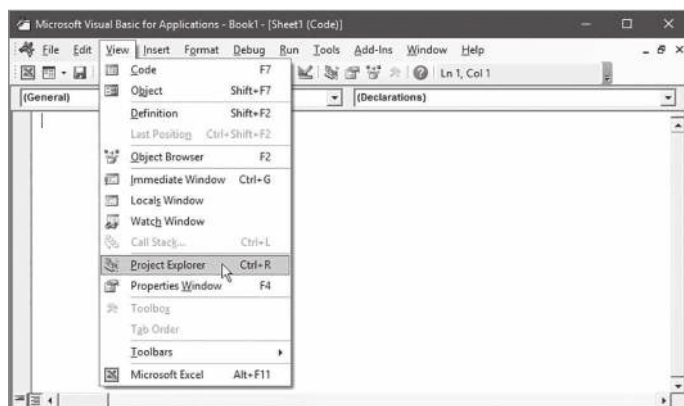
Создав макрос способом, описанным ранее, вы можете просмотреть инструкции программирования VBA в редакторе Visual Basic:

- 1 Чтобы запустить редактор Visual Basic, нажмите кнопку **Visual Basic** на панели **Developer** (Разработчик).



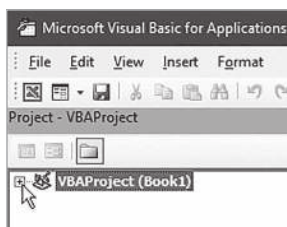
Для открытия редактора Visual Basic вы можете воспользоваться сочетанием клавиш **Alt + F11**.

- 2 В редакторе Visual Basic выберите команду меню **View** ⇒ **Project Explorer** (Вид ⇒ Обозреватель проектов), чтобы открыть соответствующую панель.



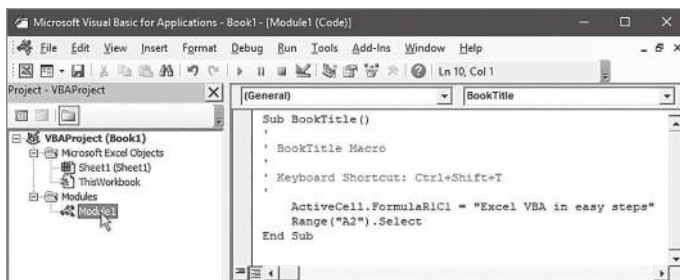
При открытии редактора Visual Basic панель **Project Explorer** (Обозреватель проектов) уже может быть открыта, однако пользователю полезно самому научиться открывать и закрывать эти элементы, чтобы полностью ознакомиться с интерфейсом редактора Visual Basic.

- 3 Для просмотра группы элементов на панели **Project Explorer** (Обозреватель проектов) нажмите кнопку **+** возле строки **Book1**.



4

Чтобы увидеть код макроса, на панели **Project Explorer** (Обозреватель проектов) в папке **Modules** дважды щелкните мышью по пункту **Module1**.



В круглых скобках () в первой строке кода должны быть написаны параметры. Больше информации см. в главе 6.

Анализ программного кода

- **Sub BookTitle()** указывает на начало подпрограммы (**Sub**) с именем как у макроса (**BookTitle**), который вы именовали перед началом записи.
- **BookTitle Macro** — примечание, которое означает, что эта подпрограмма была создана для макроса с указанным названием.
- **Keyboard Shortcut: Ctrl+Shift+T** — еще одно примечание, описывающее выбранное вами сочетание клавиш для выполнения макроса.
- **ActiveCell.FormulaR1C1 = "Excel VBA in easy steps"** — эта инструкция была написана в то время, когда вы вводили название книги в ячейку и нажимали клавишу **Enter**.
- **Range("A2").Select** — эта инструкция была написана в тот момент, когда фокус сместился на ячейку **A2**.
- **End Sub** означает конец подпрограммы. Эта строка была записана в тот момент, когда вы закончили запись макроса.




Все строки, начинающиеся с апострофа, при выполнении макроса игнорируются.

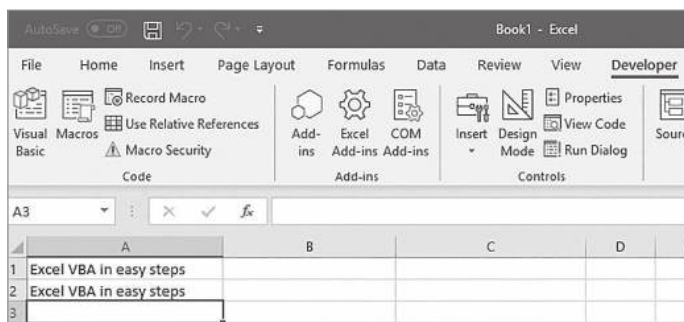
Использование различных цветов в коде применяется для выделения синтаксических элементов и облегчения чтения кода. Редактор Visual Basic автоматически активирует эту функцию. Синим цветом выделяются «ключевые слова», которые имеют важное значение в коде, а зеленым — комментарии, описывающие код. Для удобства читателя эти цвета используются во всех примерах в этой книге.

Тестирование макроса

Прежде чем записывать макрос способом, описанным ранее, пользователь в диалоговом окне **Record Macro** (Запись макроса) указывает сочетания клавиш. Давайте проверим, как таким образом можно запустить макрос:

1 Открыв редактор Visual Basic, выберите команду меню **View** ⇒ **Microsoft Excel** (Вид ⇒ Microsoft Excel) или щелкните мышью по значку , чтобы вернуться к интерфейсу Excel.

2 Выберите пустую ячейку **A3**.



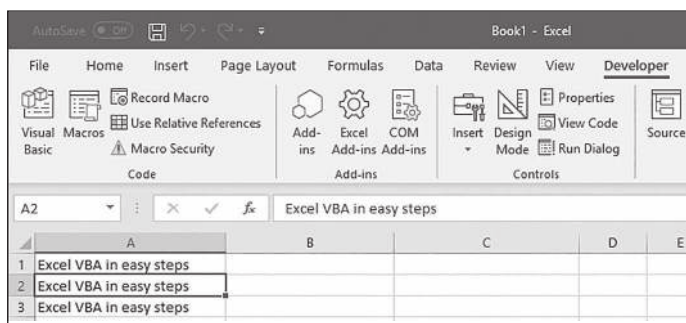
3 Теперь нажмите сочетание клавиш **Ctrl + Shift + T**, чтобы попробовать выполнить макрос. В выбранной ячейке должно появиться название книги, а фокус вернется на ячейку **A2**.



Для закрытия редактора Visual Basic вы можете использовать сочетание клавиш **Alt + F11**.



Если вы попытаетесь использовать сочетание клавиш, которое уже используется для другого макроса, то появится диалоговое окно с просьбой указать другое сочетание клавиш, чтобы не было совпадений.



Важно помнить, что ячейка **A1** была выбрана до записи макроса. Если вы выберете нужную ячейку уже после начала записи, то программа примет это действие за «инструкцию», и при каждом запуске макроса название книги будет записываться только в ячейку **A1**.

Изменение кода макроса

Теперь вы знаете, что можете запустить макрос и с помощью кнопки **Run** (Выполнить) из диалогового окна, и с помощью сочетания клавиш **Ctrl + Shift + T**. Но, скорее всего, вам не нужно, чтобы после каждого выполненного макроса фокус возвращался на ячейку **A2**. Вы можете изменить код, чтобы удалить инструкции для фокуса, а заодно изменить стиль шрифта:



При написании инструкции в случае ввода знака разделителя или точки будет появляться всплывающее окно со списком предложений, где вы можете выбрать нужный элемент.

- 1 На панели **Developer** (Разработчик) в группе элементов **Code** (Код) нажмите кнопку **Visual Basic**. Появится окно редактора Visual Basic.
- 2 На панели **Project Explorer** (Обозреватель проектов) дважды щелкните мышью по пункту **Module1**, чтобы увидеть код макроса VBA.
- 3 Удалите строку, которая отвечает за возвращение фокуса.
Range("A2").Select