

Работа с процедурами VBA

Процедура — это последовательность операторов VBA, расположенная в модуле VBA, доступ к которому можно получить с помощью VBE. Модуль может включать любое количество процедур.¹ Некоторые процедуры получают аргументы. Аргумент — это информация, используемая процедурой в процессе выполнения. Аргументы процедуры во многом подобны аргументам, используемым функциями Excel.

При объявлении процедуры с использованием ключевого слова Sub применяется следующий синтаксис.

```
[Private | Public][Static] Sub имя( [список_аргументов] )  
    [инструкции]  
    [Exit Sub]  
    [инструкции]  
End Sub
```

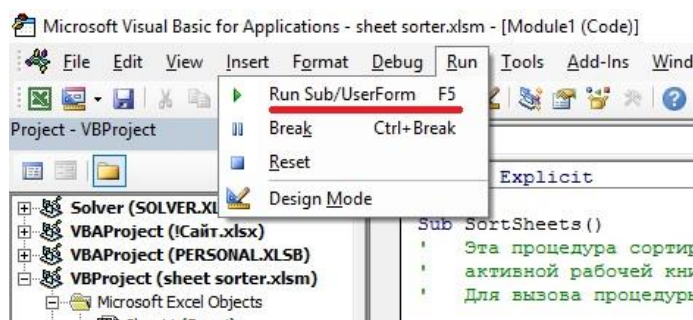


Рис. 1. Запуск процедуры из Visual Basic Editor

Private (необязательное ключевое слово). Указывает на то, что процедура доступна только для других процедур в том же модуле.

Public (необязательное ключевое слово). Указывает на то, что процедура доступна для всех остальных процедур во всех модулях рабочей книги. При использовании в модуле, содержащем оператор `Option Private Module`, процедура будет недоступна за пределами проекта.

Static (необязательное ключевое слово). Указывает на то, что переменные процедуры сохраняются после окончания процедуры.

Sub (обязательное ключевое слово). Обозначает начало процедуры.

Имя. Любое корректное название процедуры.

Список_аргументов. Представляет заключенный в скобки список переменных, содержащих аргументы, которые передаются в процедуру. Для разделения аргументов используется запятая. Если процедура не использует аргументы, то необходимо включить в объявление процедуры пустые скобки.

Инструкции (необязательные). Корректные инструкции VBA.

Exit Sub (необязательный оператор). Вызывает немедленный выход из процедуры до ее формального завершения.

End Sub (обязательный оператор). Указывает на завершение процедуры.

Выполнение процедуры

Основные способы выполнения, или вызова, процедуры VBA.

1-й способ. С помощью команды `Run` → `Run Sub/UserForm` (Выполнить → Выполнить процедуру/пользовательскую форму, рис. 1) в VBE. Альтернатива — нажать <F5> либо воспользоваться кнопкой `Run Sub/UserForm` панели инструментов Standard (Стандартная, рис. 2).

¹ По материалам книги [Джон Уокенбах. Excel 2010. Профессиональное программирование на VBA](#). – М: Диалектика, 2013. – С. 253–273.

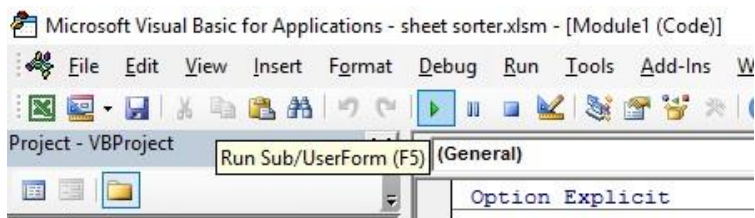


Рис. 2. Кнопка Run Sub/UserForm на панели Standard VBE

2-й способ. Из диалогового окна *Макрос* в Excel (рис. 3). Чтобы вызвать окно пройдите по меню *Разработчик* → *Макрос* или нажмите Alt+F8.

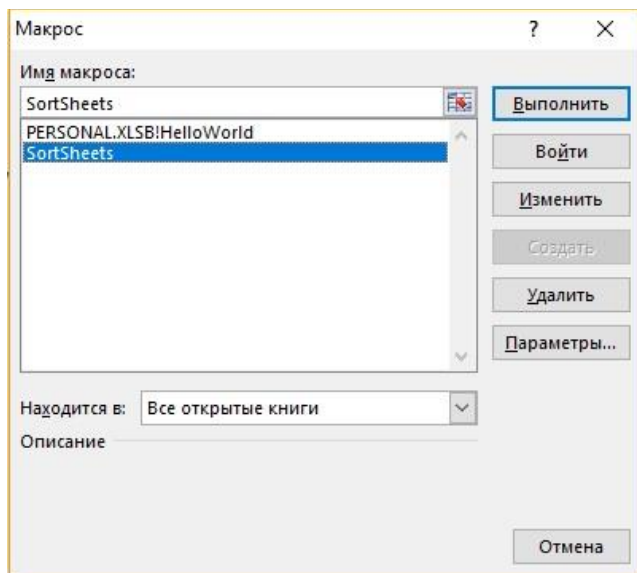


Рис. 3. Диалоговое окно *Макрос* в Excel

3-й способ. С помощью комбинации клавиши <Ctrl> и присвоенной процедуре клавиши (если процедуре присвоена комбинация клавиш). Если в момент создания процедуры ей не была присвоена клавиша, сделать это никогда не поздно. Откройте окно *Макрос*, как описано выше, выделите процедуру в окне *Имя макроса*, кликните *Параметры*, и введите букву в окне *Сочетание клавиш* (рис. 4).

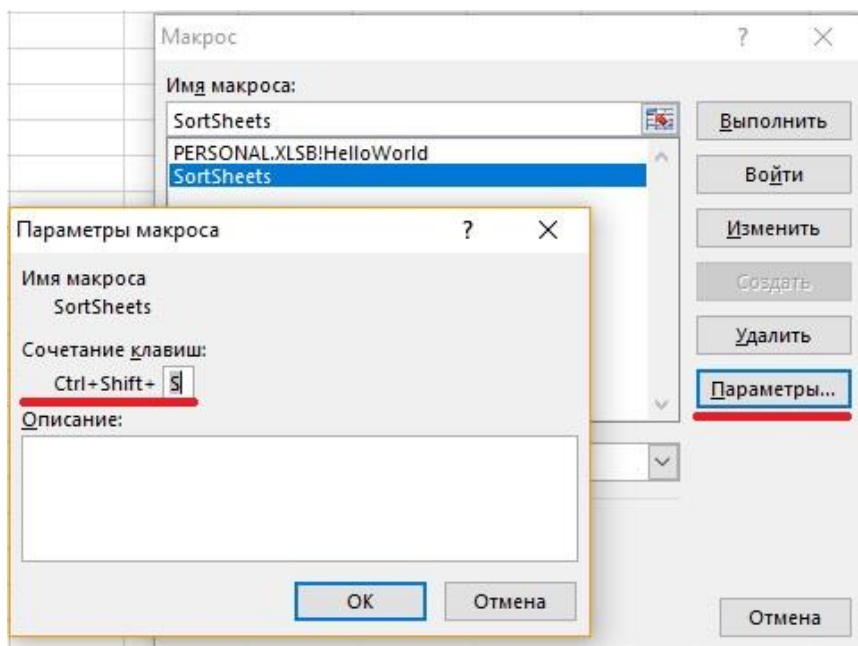


Рис. 4. Присвоение процедуре комбинации клавиш

4-й способ. Щелкнув на кнопке или любой фигуре рабочего листа. Для этого кнопке или фигуре должна быть присвоена процедура (рис. 5).

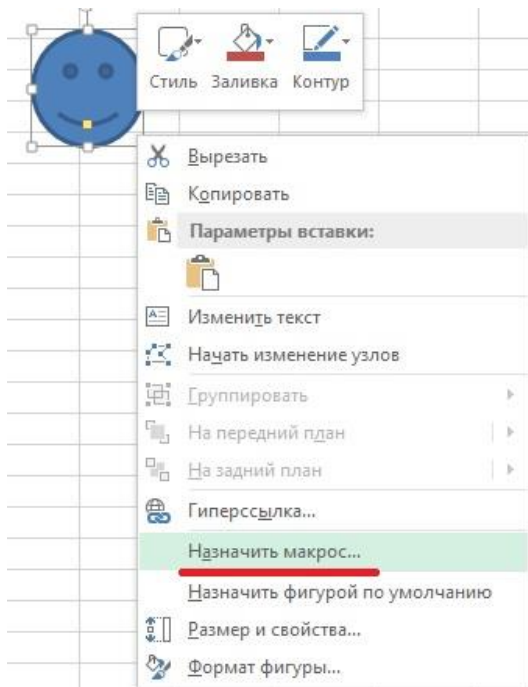


Рис. 5. Назначение макроса фигуре

5-й способ. Из другой процедуры. Процедуры `Sub` и `Function` могут вызывать другие процедуры.

6-й способ. С помощью пользовательского элемента управления, находящегося на ленте. Кроме того, встроенные элементы управления ленты могут быть «перенастроены» для вызова макроса на выполнение.

7-й способ. Из пользовательского контекстного меню.

8-й способ. После выполнения определенного события. Такими событиями могут выступать открытие рабочей книги, сохранение рабочей книги, закрытие рабочей книги, изменение ячейки, переход на другой рабочий лист и многие другие.

9-й способ. Из окна отладки (*Immediate*) в VBE. Просто введите название процедуры, укажите все необходимые аргументы и нажмите клавишу <Enter>.

Передача аргументов процедурам

Аргументы обеспечивают процедуру данными, используемыми в ее инструкциях. Аргумент может передавать следующие данные: переменная, константа, массив, объект.

Используются два способа передачи аргументов процедуре.

- *По ссылке.* При передаче аргумента по ссылке (метод, применяемый по умолчанию) процедуре передается адрес ячейки памяти, в которой хранится переменная. Поэтому изменение аргумента в процедуре приводит к изменению исходной переменной.
- *По значению.* Передача аргумента по значению фактически означает передачу процедуре копии исходной переменной. Следовательно, изменение аргумента при выполнении процедуры не отражается на исходной переменной.

В следующем примере аргумент процедуры `Process` передается по ссылке (по умолчанию). После того как процедура `Main` присваивает переменной `MyValue` значение 10, она вызывает процедуру `Process` и передает `MyValue` в качестве аргумента. Процедура `Process` умножает значение своего аргумента (с названием `YourValue`) на 10. По окончании процедуры `Process` возобновляется выполнение процедуры `Main`, а функция `MsgBox` отображает строку `MyValue: 100`.

```
Sub Main ()
    Dim MyValue As Integer
    MyValue = 10
    Call Process(MyValue)
```

```

    MsgBox MyValue
End Sub

Sub Process (YourValue)
    YourValue = YourValue * 10
End Sub

```

Если требуется, чтобы вызываемая процедура не изменяла переменные, полученные как аргументы, измените список аргументов вызываемой процедуры так, чтобы аргументы передавались по значению, а не по ссылке. Для этого добавьте перед аргументом ключевое слово `ByVal`. Тогда вызываемая процедура будет управлять копией переданных данных, а не самими данными. В следующей процедуре, например, изменения, которые происходят с `YourValue` в процедуре `Process`, не влияют на значение переменной `MyValue` в процедуре `Main`. В результате функция `MsgBox` отображает 10, а не 100.

```

Sub Process(ByVal YourValue)
    YourValue = YourValue * 10
End Sub

```

Обработка ошибок

Чтобы указать программе, что должно произойти при возникновении ошибки, используется оператор `On Error`. Вы вправе выбрать один из двух вариантов.

- Прогнорировать ошибку и позволить VBA продолжить выполнение программы. После этого можно проанализировать объект `Err`, чтобы узнать, какая ошибка произошла, и при необходимости принять меры для ее предотвращения.
- Перейти к специальному разделу кода для обработки ошибок, чтобы выполнить необходимые действия. Этот раздел вводится в конце процедуры и обозначается специальной меткой.

Чтобы программа продолжала выполняться после возникновения ошибки, необходимо вставить в начало процедуры оператор `On Error Resume Next`. При возникновении ошибки можно использовать объект `Err` для определения ее номера. Например, на рис. 6 представлена процедура, присваивающая Листу2 имя Исходные данные. Однако, в книге может не быть Листа2. В этом случае появится сообщение об ошибке.

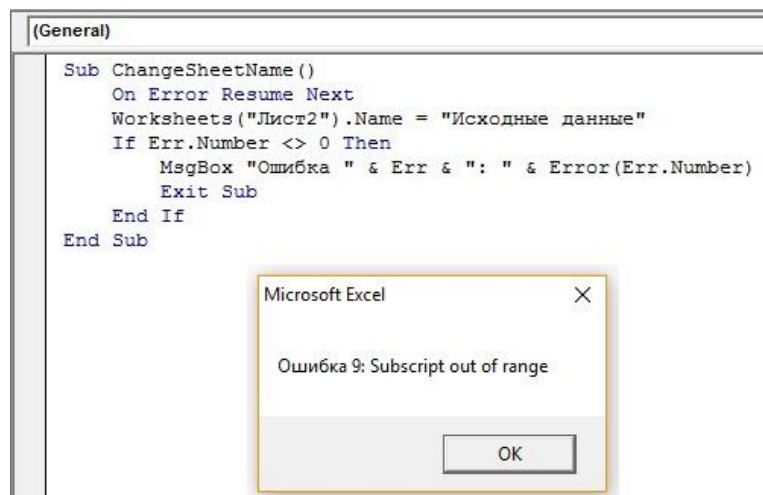


Рис. 6. Процедура присвоения имени Листу Excel, обрабатывающая ошибку

Ссылка на `Err` эквивалентна обращению к свойству `Number` объекта `Err`. Следовательно, два приведенных ниже оператора идентичны:

```

MsgBox Err
MsgBox Err.Number

```

Оператор `On Error` также применяется для определения места в процедуре, к которому должна перейти программа в случае ошибки. Чтобы обозначить это место, используется метка.

```

On Error GoTo ErrorHandler

```

Следующая процедура выделяет все ячейки в текущем диапазоне, содержащие формулы, возвращающие число. Процедура также использует оператор `If` для определения результата: произошла ли ошибка. Оператор `On Error GoTo 0` восстанавливает нормальную обработку ошибок перед выходом из процедуры.

```
Sub SelectFormulas2()  
    On Error Resume Next  
    Selection.SpecialCells(xlFormulas, xlNumbers).Select  
    If Err.Number = 1004 Then MsgBox "Не найдены ячейки с формулами."  
    On Error GoTo 0  
    ` ...[код]  
End Sub
```

Если свойство `Number` объекта `Err` не равно 0, происходит ошибка. С помощью оператора `If` проверяется, не равно ли свойство `Err.Number` 1004, и, если это так, отображается окно сообщения. В рассмотренном примере осуществляется проверка кода на предмет обнаружения ошибки с указанным номером.

В следующем примере кода демонстрируется обработка ошибок путем перехода по метке.

```
Sub ErrorDemo()  
    On Error GoTo Handler  
    Selection.Value = 123  
Exit Sub  
Handler:  
    MsgBox "Невозможно присвоить значение выделенному диапазону."  
End Sub
```

В процедуре предпринимается попытка присвоить значение текущему выделенному объекту. Если происходит ошибка (например, не выделен диапазон ячеек или лист защищен), то оператор присваивания выдает ошибку. Оператор `On Error` задает переход к метке `Handler` в случае ошибки. Обратите внимание, что перед меткой используется оператор `Exit Sub`. Программа обработки не выполняется, если ошибок не было.