**Расширяем возможности макросов в Excel за счет средств Visual Basic**

*Заметка написана Андреем Макаренко*

Всем, кому приходится использовать функции Excel, на определенном этапе сталкиваются с проблемой ограниченности предлагаемого функционала. Первый выход из этой ситуации — все более глубокое изучение встроенных возможностей (знакомясь с этой темой, не устаешь поражаться как богатству предлагаемых Excel возможностей, так и неординарности решений их использования). Однако нужно признать, что имеющаяся ограниченность объективна, т.к. «нельзя объять необъятное» с одной стороны, и изучение специфических функций Excel нигде кроме Excel не потребуется с другой.

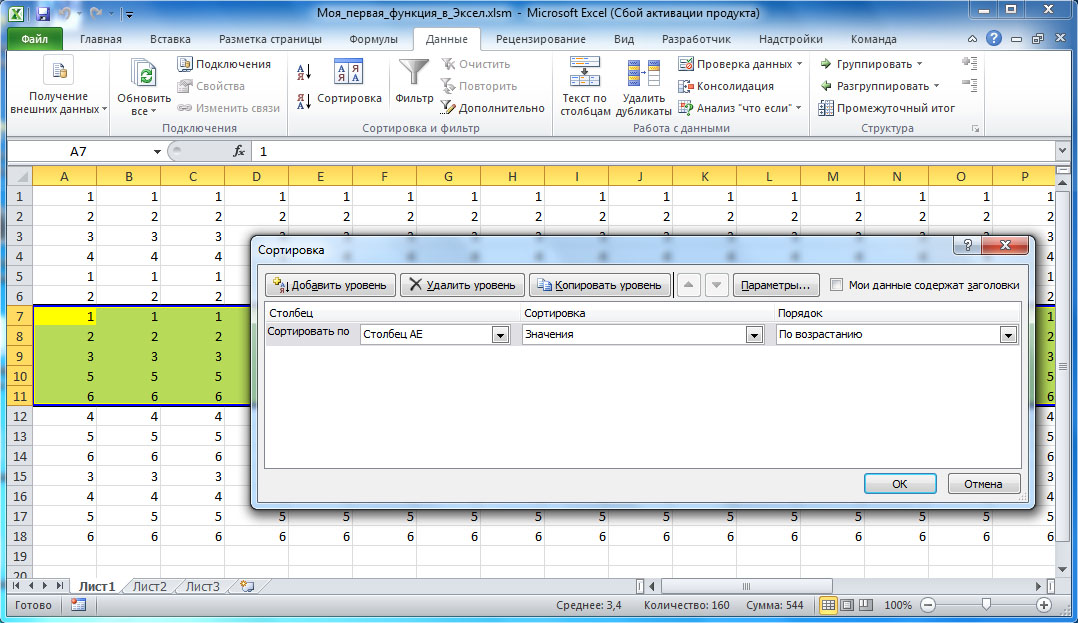


Рис. 1. Диалоговое окно сортировки предоставляет большой выбор параметров

Поэтому следующий уровень развития — это написание своих собственных функций. Выходя на этот уровень, получаешь в качестве инструментов всё богатство управляющих конструкций, регулярные выражения, доступ к уже созданным библиотекам и «прочая, прочая, прочая», и всё это на основе знаний, применяемых в том или ином виде в любом языке программирования. Первому шагу на этом пути посвящена данная статья, целью которой является показать, что программирование — это просто!

Автоматизировать можно и нужно любое часто повторяемое действие. Возьмем для примера сортировку по заданному столбцу — «AE» (мы выбрали удаленный столбец преднамеренно, т.к. его ручное выделение затрудненно). Для ее выполнения проводятся следующие действия (используем Excel2010):

1. Выделяем подлежащие сортировке строки
2. Вызываем диалоговое окно *Данные* –> *Сортировка*
3. В поле «Сортировать по» выбираем колонку «AE»
4. Нажимаем Ok

В среднем операция занимает 15 секунд. За день у нас она повторяется раз 200. Итого 3000 секунд, или практически час тупого повторения. Т.к. операции эти делать нам, мы хотим от них избавиться! Чтобы не копаться в объектной модели Excel (у нас же нет цели стать великими программистами) можно здорово упростить задачу используя механизм макросов. Мы запишем последовательность необходимых действий, а потом доработаем их под наши задачи. Макросы в Excel по умолчанию выключены (точнее, записать вы их сможете, но вот сохранить или открыть файл с макросами, нет). Чтобы включить возможность работать с макросами, пройдите по меню *Файл* –> *Параметры* –> *Центр управления безопасностью* –> *Параметры центра управления безопасностью* –> *Параметры макросов* –> *Включить все макросы*. Теперь можно сохранить исходный файл с расширением \*.xlsm — файл Excel с поддержкой макросов.

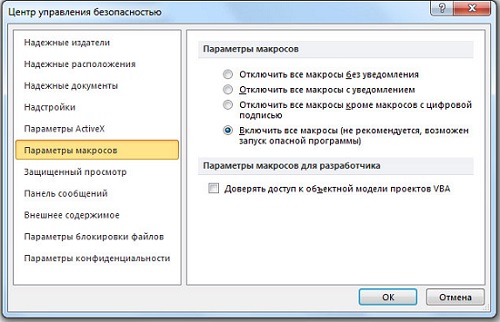


Рис. 2. Включая возможность выполнения макросов вы переключаете на себя контроль над защитой от вирусов

Обеспечиваем себе удобный доступ к средствам программирования: *Файл* –> *Параметры* –> *Панель быстрого доступа*. Ставим «галку» в правой части (*Настройка ленты*) напротив поля *Разработчик*.

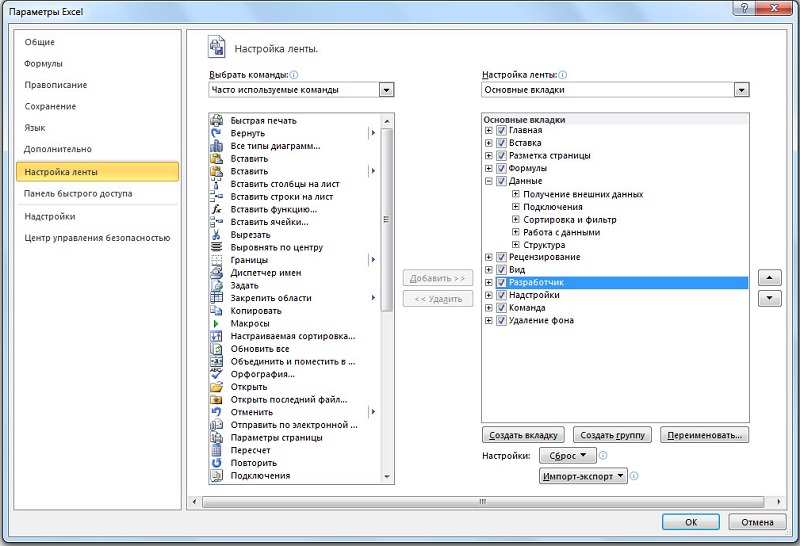


Рис. 3. Выводим вкладку *Разработчик* на ленту

В ленте появилась вкладка *Разработчик*. Выбираем её и в разделе *Код* нажимаем *Запись макроса.* Макрос можно также запустить, нажав кнопку в *Строке состояния* Excel в нижнем левом углу окна (рис. 4). После начала записи макроса эта кнопка меняет вид, и теперь с ее помощью можно остановить запись макроса.

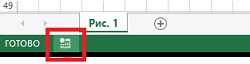


Рис. 4. Кнопка в Строке состояния, активирующая начало записи макроса

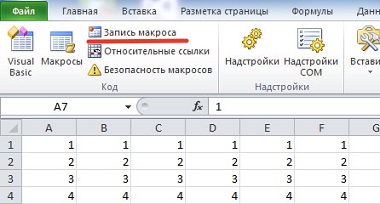


Рис. 5. После нажатия на кнопку *Запись макроса*, ее название меняется на *Остановить*

Даем имя макросу, присваиваем клавиатурное сокращение Ctrl+q (рис. 6), и начинаем запись последовательности выполняемых действий.

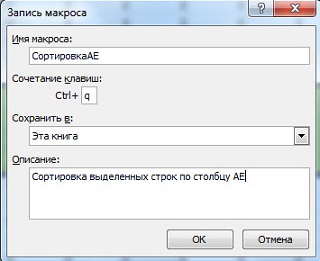


Рис. 6. Параметры макроса можно установить и изменить в дальнейшем в диалоговом окне *Макросы*

Выполняем описанную выше последовательность из четырех пунктов, выделяя в данном примере строки с 1-й по 6-ю (см. рис. 1), и нажимаем кнопку *Остановить запись*. Чтобы увидеть и отредактировать получившийся код, нажимаем кнопку *Макросы*, выбираем наш макрос (рис. 7), нажимаем кнопку *Изменить* и попадаем в редактор VisualBasic. Мы увидим код, который на рис. 8 приведен с комментариями.

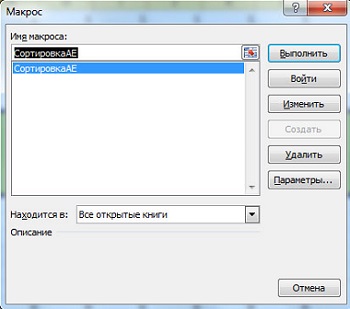


Рис. 7. Окно *Макрос* позволяет выбрать и отредактировать код макроса

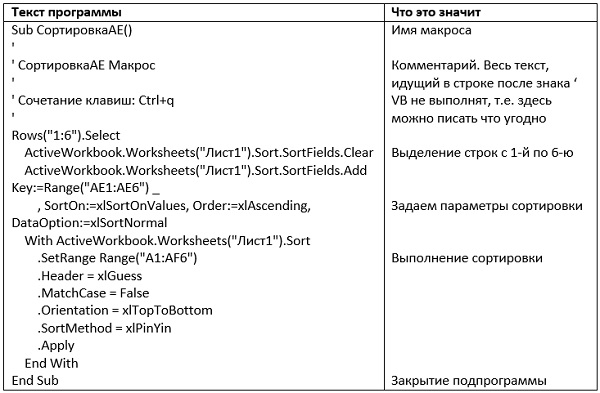


Рис. 8. Код макроса

Как я смог написать комментарии справа? Очень просто — один абзац, как правило, соответствует одной операции, совершенной при записывании макроса в той же последовательности, как мы выполняли операции. Нужно ли разбираться в сгенерированном коде более подробно? Все покажет тестирование написанной программы. Если она будет выполнять наши ожидания, то не надо, если нет — то придется вникать глубже. Таким образом, мы получили последовательность действий, которую можно повторять, вызывая макрос заданным клавиатурным сокращением Ctrl+q. Недостаток этой программы очевиден — мы можем применить ее только к жестко заданным строкам. Вот с этого места и начинается программирование.

Очевидно, чтобы программа работала так, как нам нужно, мы должны подставить в неё вместо строк с первой по шестую номера фактически выделенных. Куда подставлять? Обратим внимание на две строки, в которых упоминаются эти номера:

указание диапазона, по которому производится сортировка  
Key:=Range("AE1:AE6"), SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal

указание диапазона, в котором производится сортировка  
.SetRange Range("A1:AF6")

Т.е., подставлять нужно в выражения в кавычках: "AE1:AE6" и "A1:AF6". Что это за текст? A1:AF6 – это имена ячеек левого верхнего и правого нижнего углов выделенного диапазона данных. AE1:AE6 – это имена верхней и нижней ячеек, по которым ведется сортировка. Имена столбцов (A, AE и AF) будут оставаться неизменными, какие бы строки мы не выделили, значит нужно определить только номера первой и последней выделенных строк и подставить их вместо 1 и 6. За дело!

Для начала объявляем переменные, которым в дальнейшем присвоим искомые значения. Делается это следующими строками:

Dim strLineBegin As String 'первая строка сортируемого диапазона  
Dim strLineEnd As String 'последняя строка сортируемого диапазона

Придерживаемся при этом следующих несложных правил (необязательных, просто их применение облегчит нам жизнь в дальнейшем):

1. Присвоение переменных производим сразу после имени макроса, т.е. строки   
   Sub СортировкаАЕ()
2. Имя переменной включает:
   1. первые три символа строчными буквами сокращенное обозначение типа переменной; в нашем случае str, т.е. String — текст; в практической жизни еще применяются типы Integer — целое число, Single — число с плавающей запятой, Boolean — да/нет (подробнее см., например, [здесь](http://on-line-teaching.com/vba/lsn007.html))
   2. далее с большой буквы латинскими буквами собственно имя

Таким образом, начало нашей программы приобретает вид:

Sub СортировкаАЕ()   
Dim strLineBegin As String 'первая строка сортируемого диапазона  
Dim strLineEnd As String 'последняя строка сортируемого диапазона  
…

Далее Интернет нам в помощь и запросами типа «vba номер выделенной строки excel» узнаем, что получить номер строки выделенной ячейки можно простой командой:

strLineBegin = ActiveCell.Row

Однако такой вариант не годится при одновременном выделении нескольких строк. Модифицируем запрос на «vbа номер первой и последней строки выделенного диапазона excel». Находим следующую конструкцию:

strLineBegin = Selection.Row   
strLineEnd = strLineBegin + Selection.Rows.Count – 1

Подставляем их в наш код:

Sub СортировкаАЕ()   
Dim strLineBegin As String 'первая строка сортируемого диапазона  
Dim strLineEnd As String 'последняя строка сортируемого диапазона

strLineBegin = Selection.Row  
strLineEnd = strLineBegin + Selection.Rows.Count – 1  
…

В этом месте хорошо бы проверить, что у нас получается. Для этого, находясь в редакторе VB, нажимая клавишу F8, пошагово выполняем написанный код. Объявления переменных при этом пропускаются, а после каждой выполненной команды дальнейшее выполнение приостанавливается до следующего нажатия. Таким образом, пройдя написанные выше строки, нажимая клавишу F8, мы можем посмотреть, какие значения присвоились переменным, наведя на них курсор мыши.

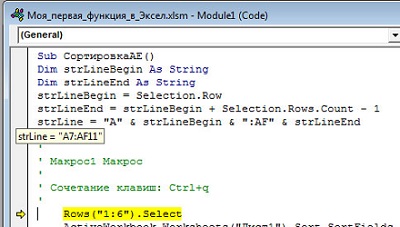


Рис. 9. Узнать текущее значение переменной можно, наведя на нее курсор мыши

Делаем это и убеждаемся, что все работает! Дальше проходить код бессмысленно, т.к. мы его еще не поправили. Прерываем выполнение программы, нажав на панели инструментов кнопку *Reset* — синий квадратик.

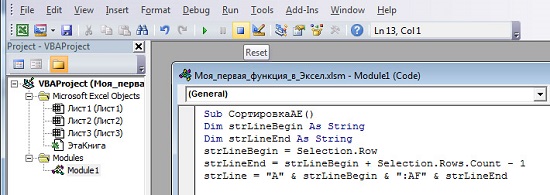


Рис. 10. Управление выполнением макроса производится с помощью кнопок: выполнить, пауза, стоп

Остается подставить полученные значения в команду сортировки. Делаем это следующим образом:

Объявляем переменные с результирующим текстом

Dim strCol As String 'колонка, по которой производится сортировка  
Dim strLine As String 'диапазон, который сортируем

Формируем ее значение, используя не фактические значения, а переменные (напомню, что должен получиться текст "AE1:AE6" и "A1:AF6"):

strCol = "AE" & strLineBegin & ":AE" & strLineEnd  
strLine = "A" & strLineBegin & ":AF" & strLineEnd

Здесь имена столбцов AE, A и AF — константы, нам их менять не нужно, а знаки амперсанда (&) служат операторами, склеивающими части текста. Обращаем внимание на то, что чистый текст берется в английские кавычки, в то время как названия переменных идет без них.

Подставляем полученное выражение в условие сортировки:

Key:=Range(strCol) , SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal

и

.SetRange Range(strLine)

Опять обращаем внимание, что, т.к. текст в строке заменен на имя переменной, то кавычки не ставим.

Команда – Rows("1:6").Select – из записанного макроса, выделяющая строки нам не нужна, мы будем подставлять вместо фиксированных значений, фактические номера выделенных строк, убираем эту строку из макроса (точнее, сохраняем ее в виде примечания, неисполняемой части кода). Итого остается (рис. 11).

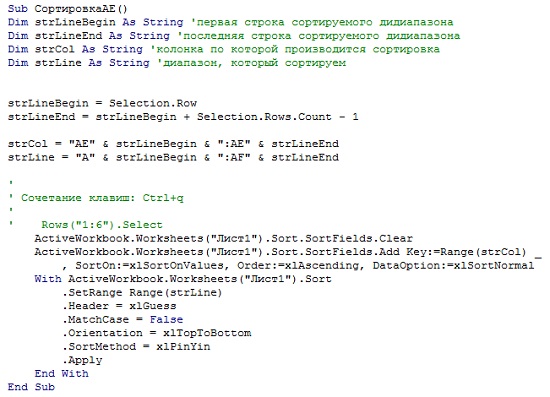


Рис. 11. Итоговый код VBA, позволяющий сортировать выделенный диапазон по столбцу АЕ по возрастанию

Идем на страницу Excel, выделяем желаемый диапазон и нажимаем Ctrl+q (помните, мы с самого начала привязали выполнение данного макроса к этому клавиатурному сочетанию) и убеждаемся, что все работает. Если не работает, переключаемся в редактор VB и проходим тест программы в режиме отладки (Debug) нажимая клавишу F8, проверяя значения переменных или определяя строку, на которых наша программа «сломается» и устраняя возникшие проблемы. У меня все прошло гладко. Вуаля!

См. также

[Написание пользовательской функции в Excel](http://baguzin.ru/wp/?p=158)

[Сумма по цвету ячеек в Excel](http://baguzin.ru/wp/?p=317)