Лекция 8.
Объекты VBA

РХТУ им. Д.И. Менделеева
Каф. ИКТ
Курс создал: ст. преп. А.М. Васецкий

2013 г

Объект Application

Объект Application (приложение) является главным в иерархии объектов Excel и представляет само приложение Excel. Он имеет более 120 свойств и 40 методов. Эти свойства и методы предназначены для установки общих параметров приложения Excel.

Application находится на вершине объектной модели Excel и содержит все остальные объекты. Кроме этого, объект Application выступает хранилищем для свойств и методов, которые не подходят для включения в любой другой объект, но необходимы для программного управления Excel. Например, существуют свойства объекта Application, предназначенные для управления обновлением экрана и включения предупреждений.

Свойства объекта Application

Глобальные члены

Многие методы и свойства объекта Application являются членами группы <globals>, доступной в самом начале списка классов в окне Object Browser Если свойство или метод входит в группу <globals>, на него можно ссылаться, не указывая ссылку на объект.

Ссылки эквивалентны:

Application.ActiveCell

ActiveCell

Однако, не все свойства объекта Applications глобальны

Например ScreenUpdating, не являются глобальными.

Корректно: Application.ScreenUpdating = False

А применение ScreenUpdating = False приводит к созданию новой переменной ScreenUpdating

Свойства объекта Application типа Active

Объект Application предоставляет множество ссылок, которые можно применять для обращения к активным объектам без указания явного имени. Это даёт возможность создавать универсальный код, который работает с объектами одного и того же типа, но имеющими разные имена.

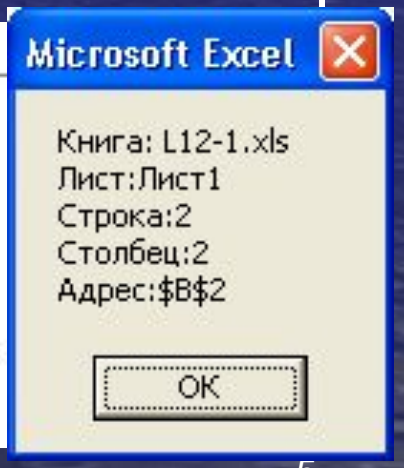
Свойства объекта Application

Следующие свойства объекта **Application** являются глобальными и позволяют ссылаться на активные объекты:

ActiveCell	Активная ячейка
ActiveChart	Активная диаграмма
ActivePrinter	Активный принтер
ActiveSheet	Активный лист
ActiveWindow	Активное окно
ActiveWorkbook	Активная книга
	Выделение

```
Option Explicit

Sub Appl_test()
MsgBox "Книга: " & ActiveWorkbook.Name & _
vbCr & "Лист:" & ActiveSheet.Name & vbCr & _
"Строка:" & ActiveCell.Row & vbCr & "Столбец:" & _
ActiveCell.Column & vbCr & "Адрес:" & ActiveCell.Address
End Sub
```



Свойства объекта Application

Свойство **Selection** не будет возвращать ссылку на объект **Range**, если выделен объект другого типа, например **Shape**, или активный лист не является листом электронной таблицы. Возможно, в макрос потребуется добавить условие, которое будет проверять, выделен ли лист электронной таблицы, перед тем как вставлять данные.

```
If TypeName(ActiveSheet) <> "Worksheet" Or _
    TypeName(Selection) <> "Range" Then _
    MsgBox "Этот макрос может
использоваться" _
& "только вместе с диапазоном", vbCritical
    Exit Sub
End If
```

Пример

Application позволяет вызывать более 400 встроенных функций рабочего листа при помощи конструкции вида:
Application.ФункцияРабочегоЛиста (Аргументы)

Следует обратить внимание, что формат использования матричных функций несколько отличается от формата обычных. (см. пример). Функции листа в VBA задаются только в английском варианте.

Таблица соответствия русских и английских функций см. в файле funcs.xls в папке Microsoft office\...

Примеры использования функций рабочего листа

```
Option Explicit

Sub FunList1()
'демонстрация использования функций рабочего листа
Dim X As Double, Y As Double

X = Application.Pi
Y = Application.Sinh(X)
MsgBox "X=" & X & vbCr & "Y=" & Y
End Sub
```

```
Sub Inverse()
'обращение матрицы с использованием функции рабочего листа
Dim Arr1(1 To 2, 1 To 2) As Double
Dim ArrResult(1 To 2, 1 To 2) As Double
Dim Arr2 As Variant
Dim i As Byte, j As Byte

For i = 1 To 2
    For j = 1 To 2
        Arr1(i, j) = i - j ^ 1.1
    Next j
Next i

Arr2 = Application.WorksheetFunction.MInverse(Arr1)

For i = LBound(Arr2, 1) To UBound(Arr2, 1)
For j = LBound(Arr2, 2) To UBound(Arr2, 2)
    ArrResult(i, j) = Arr2(i, j)
    Debug.Print i, j, Arr2(i, j)
Next j
Next i
End Sub
```

Свойства объекта Application

ThisWorkbook	Возвращает рабочую книгу, содержащую выполняющийся в данный момент макрос. Это свойство может возвращать рабочую книгу, отличную от возвращаемой свойством ActiveWorkbook, т. к. выполняемый макрос может находиться в неактивной книге
Calculation	Устанавливает режим вычислений. Допустимые значения: xlCalculationAutomatic (автоматический режим) xlCalculationManual (вычисления выполняются вручную) xlCalculationSemiAutomatic (автоматический режим, не распространяется на таблицы)
Caption	Возвращает текст в строке имени главного окна Excel. Установка свойства равным Empty возвращает заголовок, используемый по умолчанию. В следующем примере первая инструкция устанавливает в качестве заголовка окна приложения текст Отчет, а вторая возвращает имя окна, используемое по умолчанию, т. е. Microsoft Excel: Application.Caption = "Отчет" Application.Caption = Empty

Свойства объекта Application

DisplayAlerts	Допустимые значения: True (отображаются встроенные предупреждения о работе программы) и False (предупреждения не отображаются)
DisplayFormulaBar	Допустимые значения: True (строка формул выводится в окне Excel) и False (строка формул не выводится). В данном ниже примере установлен режим, при котором строка формул не будет выводиться в окне Excel: Application. DisplayFormulaBar = False
DisplayScrollBars	Допустимые значения: True (полосы прокрутки видны в окне Excel) и False (полосы прокрутки не отображаются). Например: Application.DisplayScrollBars = False строка формул не будет выводиться в окне Excel

Примеры

Подавление предупреждений при удалении рабочего листа:

Application.DisplayAlerts = False

ActiveSheet.Delete

Application.DisplayAlerts = True

Стоит избегать выделения объектов средствами кода VBA.

Это редко когда требуется, и при отказе от выделения или активизации объектов код будет работать быстрее.

Если экран необходимо зафиксировать на время работы макроса то:

Application.ScreenUpdating = False

Свойства объекта Application (продолжение)

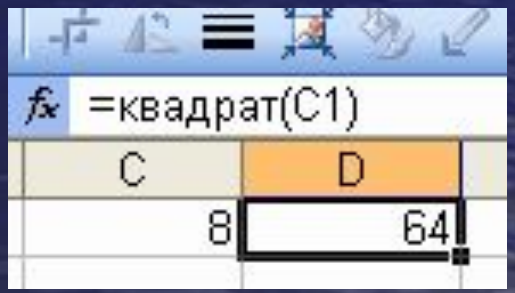
DisplayStatusBar	Допустимые значения: True (строка состояния видна в окне Excel) и False (строка состояния не видна). Например: Application.DisplayStatusBar = True - строка состояния не будет выводиться в окне Excel
EnableCancelKey	Определяет действие при нажатии комбинации клавиш <Ctrl>+<Break>, используемой для прерывания выполнения процедуры. Допустимые значения: xlDisabled (прерывания программы запрещено) xlInterrupt (прерывание процедуры разрешено) xlErrorHandler (прерывание воспринимается как ошибка)
Height	Высота окна приложения в пунктах
Width	Ширина окна приложения в пунктах
Left	Расстояние в пунктах от левой границы окна приложения до левого края экрана
Right	Расстояние в пунктах от правой границы окна приложения до правого края экрана
Top	Расстояние в пунктах от верхней границы окна приложения до верхнего края экрана

Свойства объекта Application

ScreenUpdating	Допустимые значения: True (изображение обновляется во время выполнения программы) и False (изображение не обновляется). Задание False в качестве значения свойства ускоряет выполнение процедуры. В конце процедуры свойству ScreenUpdating необходимо присвоить значение True
StatusBar	Выводит заданный текст в строке состояния. Выполнение приведенного ниже примера позволит вывести текст Ввод данных. в строке состояния: Application.DisplayStatusBar = True Application.StatusBar = "Ввод данных..."
Version	Возвращает номер текущей версии Excel. Применяется для проверки того, что приложение используется в корректной версии. Например: If Application.Version = "8.0" Then Exit Sub
WindowState	Устанавливает размер окна. Допустимые значения: xlMaximized (максимальный) xlMinimized (минимальный) xlNormal (нормальный) Например: Application.WindowState = xlMaximized – устанавливается максимальный размер окна

Методы объекта Application

Calculate	Вызывает принудительное вычисление во всех открытых рабочих книгах. Например: Application.Calculate
Run	Запускает на выполнение подпрограмму или макрос. Синтаксис: Run (Macro, Arg1, Arg2, ...) • Macro – строка с именем макроса • Arg1, Arg2, ... – аргументы передаваемые макросу. • Например: Application.Run Macro:= "Расчет" – запускает макрос Расчет
Volatile	Вызывает перевычисление функции пользователя при изменении значений параметров. Например, функция Квадрат будет автоматически пересчитывать результат на рабочем листе при изменении значения аргумента: Function Квадрат(x as Double) Application.Volatile = True Квадрат = x*x End Function



Методы объекта Application (продолжение)

Wait	Временно приостанавливает работу приложения без остановки работы других программ. Синтаксис: Wait(Time) •Time – время, в которое предполагается возобновить работу приложения В следующем примере показывается, как установить время, чтобы возобновление работы приложения началось в 17 часов: Application.Wait "17:00:00"
OnKey	Устанавливает выполнение специфицированной процедуры при нажатии заданной комбинации клавиш. Синтаксис: OnKey(Key, Procedure) •Procedure – имя выполняемой подпрограммы при нажатии клавиш •Key – строка, определяющая комбинацию клавиш, которая должна быть нажата. В этой строке можно также указывать специальные клавиши, используя следующие коды:

Run (пример)

```
Option Explicit
```

```
Sub Test()
    MsgBox Dif(1, "Line")
    MsgBox Dif(1, "Square")
    MsgBox Dif(1, "Cube")
End Sub
```

```
'Дифференцирование функции f в точке x
Function Dif(ByVal x As Double, f) As Double
Dim dx As Double, dy As Double
  dx = 0.001 'создаем приращение аргумента
'вычисляем приращение функции
  dy = Application.Run(f, x + dx / 2) - _
  Application.Run(f, x - dx / 2)
  Dif = dy / dx 'вычисляем производную
End Function
```

```
Function Line(ByVal x) As Double
    Line = x
End Function
```

```
Function Square(ByVal x) As Double
    Square = x ^ 2
End Function
```

```
Function Cube(ByVal x) As Double
    Cube = x ^ 3
End Function
```

Коды клавиш для OnKey

OnKey (продолж.)

- <Backspace> – {BACKSPACE} или {BS}
- <Break> – {BREAK}
- <Caps Lock> – {CAPSLOCK}
- <Delete> или <Del> – {DELETE} или {DEL}
- <вниз>– {DOWN}
- <End> – {END}
- <Enter> (цифровая клавиатура) – {ENTER}
- <ESC> – {ESCAPE} или {ESC}
- <Home> – {HOME}
- <Ins>или <Insert>– {INSERT}
- <<-> – {LEFT}
- <Num Lock> – {NUMLOCK}
- <Page Down> – {PGDN}
- <Page Up> – {PGUP}
- <Return> – {RETURN}
- <->> – {RIGHT}
- <Scroll Lock> – {SCROLLLOCK}
- <Tab>- {TAB}
- <вверх>- {UP}
- ОТ <F1> до<F15> – ОТ {F1}до{F15}

Коды клавиш для OnKey

Допустимо использование сочетания одновременно нажатых клавиш. С этой целью для перечисленных трех клавиш установлены следующие коды:

<Shift> - +

<Ctrl> - ^

<Alt> - %

В примере процедуре Амортизация назначена комбинация клавиш <Ctrl>+<+>, а процедуре ПроцентнаяСтавка – <Shift>+<Ctrl>+<->.>:

```
Application.OnKey "^{+}", "Амортизация"

Application.OnKey "+^ {RIGHT}", _"
ПроцентнаяСтавка"
```

Пример

```
Private Sub Workbook_Open() ‘располагается в коде
    листа
  ‘Отслеживаем нажатие клавиши
  Application.OnKey "{DEL}", "MyDel"
End Sub

Private Sub Workbook_BeforeClose(Cancel As Boolean)
‘располагается в коде листа
  ‘Восстанавливаем стандартную реакцию на клавишу
  Application.OnKey "{DEL}"
End Sub

Sub MyDel() ‘располагается в основном модуле
‘Описание действий при нажатии на клавишу
        Msgbox "Нажата клавиша DEL"
 End Sub
```

Методы объекта Application (продолжение)

OnRepeat, OnUndo	Определяет процедуру, выполняемую при выборе команды Правка, Повторить (Edit, Repeat) и Правка, Отменить (Edit, Undo) соответственно. Синтаксис: OnRepeat (Text, Procedure) OnUndo (Text, Procedure) Text – строка, задающая текст команды Правка, Повторить (Edit, Repeat) Procedure – имя подпрограммы, выполняемой при выборе команды Правка, Повторить (Edit, Repeat)
Quit	Закрывает приложение Application.Quit

Метод Evaluate

Может использоваться для расчета значения формул листов Excel и генерации ссылок на объекты Range. Стандартный синтаксис вызова метода Evaluate выглядит следующим образом:
Evaluate("Выражение")

Кроме этого, существует сокращенная форма вызова, в которой отсутствуют двойные кавычки, а выражение заключается в квадратные скобки, например:

[Выражение]

На месте Выражения может находиться любое действительное выражение на листе с или без знака равенства слева. Также это может быть ссылка на диапазон ячеек. Расчеты на листе могут включать в себя функции, недоступные в VBA через объект WorksheetFunction. Также это могут быть формулы массивов на листе

Следующие два примера являются эквивалентными и возвращают значение True, если ячейка A1 пустая, и False в противном случае:

```
MsgBox Evaluate("=ISBLANK(A1)")
MsgBox [ISBLANK(A1)]
```

Evaluate (продолжение)

Два способа использования метода **Evaluate** для генерации ссылки на объект **Range** с присвоением значения этому объекту:

```
Evaluate("A1").Value = 10
[A1].Value = 10
```

Эти выражения эквивалентны. Выражение можно сократить еще больше, опустив свойство **Value**, так как это принятое по умолчанию свойство объекта **Range**:

```
[A1] = 10
```

Метод Inputbox

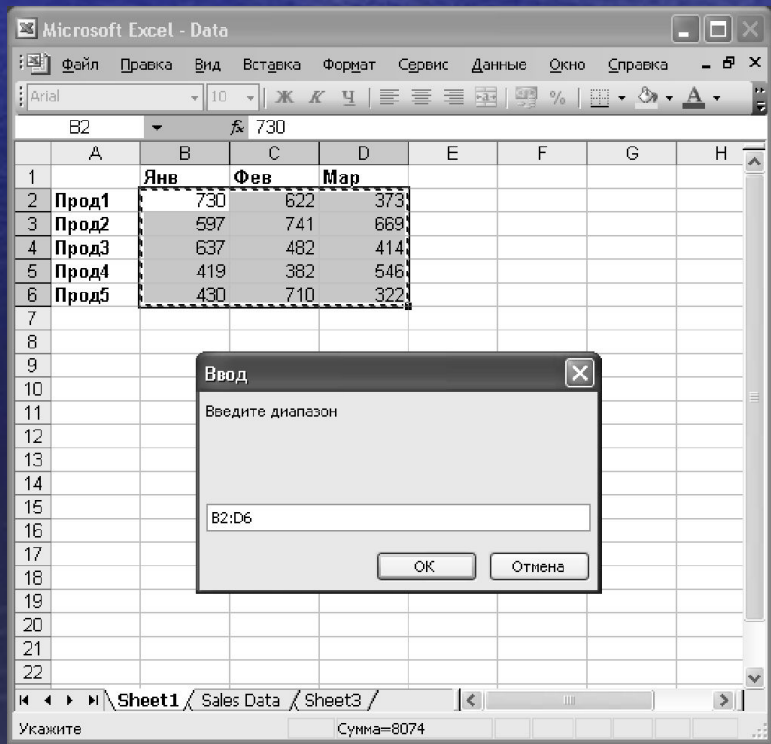
Answer = Application.InputBox(prompt:="Введите диапазон", Type:=8)

Параметр Type может принимать следующие значения (или сумму этих значений):

0	Формула
1	Число
2	Текст (строка)
4	Логическое значение (True или False)
8	Ссылка на ячейку, как объект Range
16	Значение ошибки, например, #N/A
64	Массив значений

Пример. Ввод диапазона

```
Public Sub SelectRange()
Dim aRange As Range
On Error Resume Next
Set aRange = Application.InputBox(prompt:="
    Введите диапазон", Type:=8)
If aRange Is Nothing Then
    MsgBox "Операция отменена"
Else
    aRange.Select
End If
End Sub
```



События объекта Application

NewWorkbook	При создании новой рабочей книги
WorkbookActivate	При активизации рабочей книги
WorkbookBeforeClose	Перед закрытием рабочей книги
WorkbookBeforePrint	Перед печатью рабочей книги
WorkbookBeforeSave	Перед сохранением рабочей книги
WorkbookNewSheet	При добавлении нового листа в рабочую книгу
WorkbookOpen	При открытии рабочей книги
WorkbookDeactivate	Когда активная книга теряет фокус

Объект Workbook и семейство Workbooks

В иерархии Excel объект **workbook** (рабочая книга) идет сразу после объекта **Application** и представляет файл рабочей книги. Рабочая книга хранится либо в файлах формата XLS (стандартная рабочая книга) или XLA (полностью откомпилированное приложение). Свойства и методы рабочей книги позволяют работать с файлами.

Свойства объектов Workbook и семейства Workbooks

ActiveSheet	Возвращает активный лист книги. Например: MsgBox "Имя активного листа " & ActiveSheet.Name – выводит в диалоговом окне имя активного рабочего листа
ActiveDialog	Возвращает активное диалоговое окно
ActiveChart	Возвращает активную диаграмму
Sheets	Возвращает семейство всех листов книги Пример: Set newSheet = Sheets.Add(Type:=xlWorksheet) For i = 1 To Sheets.Count newSheet.Cells(i, 1).Value=Sheets(i).Name Next i
Worksheets	Возвращает семейство всех рабочих листов книги

Свойства объекта Workbook и семейства Workbooks

Charts	Возвращает семейство всех диаграмм книги (которые не внедрены в рабочие листы)
Count	Возвращает число объектов семейства workbooks
HasPassword	Допустимые значения: True (если у документа имеется пароль защиты), False (в противном случае)
WriteReserved	Допустимые значения: True (если документ закрыт для записи), False (в противном случае)
Saved	Допустимые значения: True (если не производились изменения в документе со времени его последнего сохранения), False (в противном случае)

Методы объекта Workbook и семейства Workbooks

Activate	Активизирует рабочую книгу так, что ее первый рабочий лист становится активным. Например: Workbook.Activate
Add	Создает новый объект для семейства Workbooks. Синтаксис: Add (Template) Template – задает шаблон, на основе которого создается новая рабочая книга. Допустимые значения: xlWBATChart, xlWBATExce14IntlMacroSheet, xlWBATExce14MacroSheet или xlWBATWorksheet. Если аргумент Template опущен, то создается новая рабочая книга с количеством листов, заданных свойством SheetsInNewWorkbook

Методы объекта Workbook и семейства Workbooks

Protect	Защищает рабочую книгу от внесения в нее изменений. Синтаксис: Protect (Password, Structure, Windows) Password – строка, используемая в качестве пароля для защиты книги •Structure – допустимые значения True (защищена структура книги, т. е. взаимное расположение листов) и False (не защищена) •windows – допустимые значения True (защищено окно книги) и False (не защищено) Устанавливается защита для активной рабочей книги: ActiveWorkbook.Protect Password:= "PSW"
Unprotect	Снятие защиты с рабочей книги. Синтаксис: Unprotect (Password) Password – строка, используемая в качестве пароля для защиты листа снимается защита с активной рабочей книги: ActiveWorkbook. Unprotect Password:= "PSW"

Методы объекта Workbook и семейства Workbooks

Close	Закрытие рабочей книги
Open	Открытие существующей рабочей книги
OpenText	Открытие текстового файла, содержащего таблицу данных
Save	Сохранение рабочей книги
SaveAs	Сохранение рабочей книги в другом файле. Синтаксис: SaveAs (Filename) Filename – строка, указывающая имя файла, в котором будет сохранена рабочая книга В следующем примере активная рабочая книга сохраняется в файле с именем NewVers: ActiveBook.SaveAs Filename:= “NewVers"
SaveAsCopy	Сохранить рабочую книгу в другом файле, оставляя рабочую книгу в памяти с прежним именем. Синтаксис: SaveAs(Filename, FileFormat) Filename – строка, указывающая имя файла, в котором будет сохранена рабочая книга Активная рабочая книга сохраняется в файле с именем NewName: ActiveBook.SaveAsCopy Filename:= "NewName"

События объекта Workbook и семейства Workbooks

BeforeClose	При закрытии рабочей книги
BeforePrint	Перед печатью рабочей книги
BeforeSave	Перед сохранением рабочей книги
Deactivate	Когда рабочая книга теряет фокус
NewSheet	При добавлении нового листа
Open	При открытии рабочей книги
SheetActivate	При активизации любого рабочего листа
SheetDeactivate	Когда рабочий лист теряет фокус

Свойства объекта Worksheet и семейство Worksheets

Name	Возвращает имя рабочего листа. Первому листу активной рабочей книги присваивается имя Итоги: Worksheets(1).Name= "Итоги"
Visible	Лист виден (True) или нет (False), xlVeryHidden – лист можно отобразить только программным путём
UsedRange	Возвращает диапазон (Range) который содержит данные Worksheets("Sheet1").Activate ActiveSheet.UsedRange.Select
StandardHeight	Возвращает стандартную высоту всех строк листа в пунктах
ActiveCell	Возвращает активную ячейку активного листа
Intersect	Возвращает диапазон, являющийся пересечением 2-х диапазонов Set isect = Application.Intersect(Range("rg1"), Range("rg2")) isect.Select
Union	Возвращает диапазон, являющийся объединением нескольких диапазонов

Методы объекта Worksheet и семейство Worksheets

Activate	Активирует рабочий лист Worksheets(1).Activate
Add	Создает новый рабочий лист. Синтаксис: Add(Before, After, Count, Type) Before – указывает лист, перед которым будет размещен новый рабочий лист After – указывает лист, после которого будет размещен новый рабочий. Если аргументы Before и After опущены, то новый лист размещается перед активным листом Count – число добавляемых листов, по умолчанию имеет значение 1 Type – указывает тип добавляемого листа. Допустимые значения: xlWorksheet (по умолчанию), xlExce14MacroSheet и xlExcel4 IntlMacroSheet. Например: ActiveWorkbook.Worksheets.Add – вставляется новый лист перед активным листом активной рабочей книги

Методы объекта Worksheet и семейство Worksheets

Protect	Защищает рабочий лист от внесения в него изменений Синтаксис: Protect (Password, DrawingObjects, Contents, Scenarios, UserInterfaceOnly) Password – строка, используемая в качестве пароля для защиты листа DrawingObjects –допустимые значения: True (графические объекты защищены) и False (графические объекты не защищены). По умолчанию используется значение False Contents – допустимые значения: True (ячейки защищены) и False (ячейки не защищены). По умолчанию используется значение True scenarios –допустимые значения: True (сценарии защищены) и False (сценарии не защищены). По умолчанию используется значение True UserInterfaceOnly – допустимые значения: True (лист защищен от изменений со стороны пользователя, но не подпрограммы VBA) и False (лист защищен от изменений со стороны как пользователя, так и подпрограммы VBA). По умолчанию используется значение False В примере установлена полная защита активного рабочего листа от любых изменений со стороны пользователя: Active Sheet.Protect Password:="Секрет", DrawingObjects :=True, Contents:=True, Scenarios: =True

Методы объекта Worksheet и семейство Worksheets

Unprotect	Снятие защиты с рабочего листа. Синтаксис: Unprotect (Password) Password – строка, используемая в качестве пароля для защиты листа В примере снимается защита с активного рабочего листа: ActiveSheet.Protect Password: ="Секрет"
Delete	Удаляет рабочий лист. Worksheets(1).Delete - удаляется первый рабочий лист из активной рабочей книги
Copy	Копирование рабочего листа в другое место рабочей книги. Синтаксис: Copy (Before, After) Before – рабочий лист книги, перед которым вставляется данный After – рабочий лист, после которого вставляется данный Одновременно допустимо использование только одного из аргументов. В примере Лист1 активной рабочей книги копируется после Лист3 той же рабочей книги: Worksheets("Лист1").Copy after:=Worksheets ("Лист3")

Методы объекта Worksheet и семейства Worksheets

Move	Перемещение рабочего листа в другое место рабочей книги. Синтаксис: Move (Before, After) Before – лист рабочей книги, перед которым вставляется данный After – лист, после которого вставляется данный Одновременно допустимо использование только одного из аргументов.
Evaluate	Преобразует выражение в объект или значение. Используется при вводе формул и ячеек из диалоговых окон.

События объекта Worksheet и семейства Worksheets

BeforeClose	При закрытии рабочей книги
BeforePrint	Перед печатью рабочей книги
BeforeSave	Перед сохранением рабочей книги
Deactivate	Когда рабочая книга теряет фокус
NewSheet	При добавлении нового листа
Open	При открытии рабочей книги
SheetActivate	При активизации любого рабочего листа
SheetDeactivate	Когда рабочий лист теряет фокус

СПАСИБО ЗА ВНИМАНИЕ!