

к.п.н., доцент Суханова Н.Т.

ОБЪЕКТЫ И КОЛЛЕКЦИИ

ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРИЛОЖЕНИЕ

Объектно-ориентированное приложение организует данные и выполняемые операторы программного кода в связанные объекты, что облегчает разработку, организацию и работу со сложными структурами данных и действиями, выполняемыми над этими данными (или с ними). Каждый объект в программном приложении содержит данные и операторы, связанные вместе и образующие единый элемент.

В Excel объектами являются *рабочие книги, листы, диапазоны данных, таблицы, графические объекты, диалоговые окна* и само приложение Excel.

СВОЙСТВА (*PROPERTIES*) ОБЪЕКТОВ VBA

Объекты в Excel имеют свойства, определяющие их вид и поведение: рабочий лист Excel может быть видимым или нет, строки в рабочем листе или таблице имеют свойство высоты, столбцы имеют свойство ширины и так далее.

Свойства регулируют вид и поведение объекта. Чтобы изменить вид и поведение объекта, необходимо изменить его свойства.

ОБЪЕКТЫ РАБОЧЕГО ЛИСТА

Каждый объект рабочего листа сохраняет данные для своих свойств внутри самого рабочего листа вместе с пользовательскими данными. *[Пользовательские данные (user data) — это любые данные, сохраняемые объектом, которые поступают непосредственно от пользователя подобно данным, содержащимся в ячейках рабочего листа.]*

Некоторые объекты имеют свойства с одними и теми же или подобными именами: объекты **Application**, **Workbook** и **Worksheet** в Excel — все имеют свойство **Name**.

МЕТОДЫ ОБЪЕКТОВ

Объекты VBA также имеют поведение и возможности, называемые *методами (methods)*. Объект рабочей книги Excel, например, имеет встроенную способность добавлять к себе новый рабочий лист — метод добавления рабочих листов (называемый **Add**). Документ Word имеет встроенную способность проверять орфографию текста в документе — метод проверки орфографии (называемый **CheckSpelling**).

МЕТОДЫ ОБЪЕКТОВ

Методы изменяют значения свойств объектов, выполняют действия с данными (или над данными), сохраняемыми объектом. Методы во многом похожи на процедуры VBA, но связаны с объектом; к методам объекта можно обратиться, только используя объект.

Для вызова метода объекта (функции обработки данных объекта) указывается не только наименование метода, но и объект (имя), которому принадлежит метод.

ОПЕРАТОРЫ ПРОГРАММ VBA, ИСПОЛЬЗУЮЩИЕ ОБЪЕКТЫ

Операторы программ VBA, использующие объекты, обычно выполняют одно или несколько из следующих действий:

- ✓ определяют текущее состояние или статус объекта путем выборки значения, сохраняемого в определенном свойстве;
- ✓ изменяют состояние или статус объекта установкой значения, сохраненного в определенном свойстве;
- ✓ используют один из методов объекта, обеспечивая выполнение объектом одной из его встроенных задач.

ОПРЕДЕЛЕНИЕ СВОЙСТВА ИЛИ МЕТОДА ОБЪЕКТА

Прежде чем применять свойства и методы некоторого объекта, их следует сначала определить.

В операторах VBA используется следующий общий синтаксис для определения свойства или метода объекта:

<Имя объекта>.<свойство или метод>

НАИБОЛЕЕ ВАЖНЫЕ ОБЪЕКТЫ В EXCEL

Объект	Описание
Application	Само приложение Excel (host-приложение)
Chart	Диаграмма в рабочей книге
Font	Этот объект содержит атрибуты шрифта и стиля для текста, отображаемого в рабочем листе
Name	Заданное имя для диапазона ячеек рабочего листа
Range	Диапазон ячеек (одна или более) или именованный диапазон в рабочем листе
Window	Любое окно в Excel; окна используются для отображения рабочих листов, диаграмм и т.д.
Workbook	Открытая рабочая книга
Worksheet	Рабочая таблица в книге

ИСПОЛЬЗОВАНИЕ СВОЙСТВ ОБЪЕКТОВ

Свойства объектов можно использовать только двумя способами: *получать значение свойства или устанавливать его. Не все свойства объекта изменяемы.* Свойства объектов, которые нельзя изменять, называют *свойствами, доступными только на чтение (read-only)*; свойства, которые можно устанавливать, называют *свойствами, доступными для чтения/записи (read-write)*.

ИСПОЛЬЗОВАНИЕ СВОЙСТВ ОБЪЕКТОВ

Свойства используются в выражениях так же, как любое другое значение переменной или константы. Можно присваивать значение свойства переменной, использовать свойства объектов в выражениях как аргументы к функциям и процедурам или как аргументы для методов какого-либо объекта.

Чтобы присвоить некоторой переменной значение свойства объекта, используйте следующий синтаксис:

Переменная=Объект.Свойство

ИСПОЛЬЗОВАНИЕ СВОЙСТВ ОБЪЕКТОВ

Можно также использовать свойство объекта непосредственно в выражении или в качестве аргумента функции или процедуры.

Пример использования свойств объекта

1. AnyStr = aSheet.Name

В этом примере строка, сохраняемая в свойстве **Name** рабочего листа Excel, на которую ссылается объектная переменная **aSheet**, присваивается переменной **AnyStr**.

ИСПОЛЬЗОВАНИЕ СВОЙСТВ ОБЪЕКТОВ

Пример2. MsgBox InstBook.FullName

InstBook — это переменная, заданная для ссыл-ки на объект открытой рабочей книги. Если **InstBook** ссылается на рабочую книгу с именем **Sales.xls** в папке **My Documents**, то окно сообщения, вызываемое приведенным оператором, отображает строку "C:\My Documents\SALES.XLS".

ИСПОЛЬЗОВАНИЕ СВОЙСТВ ОБЪЕКТОВ

Чтобы задать свойство объекта, надо присвоить свойству новое значение, используя следующий синтаксис:

ИмяОбъекта.Свойство=Выражение

Пример изменения свойств объекта

InstSheet.Name = "Первый квартал"

В данном примере изменяется имя рабочего листа, на который ссылается объектная переменная **InstSheet**, присваивая значение свойству **Name** листа.

Таблица 2. Наиболее полезные свойства объектов в Excel

Свойство	Тип/Что означает	Где найти
ActiveCell	Object: активная ячейка в рабочем листе	Application, Window
ActiveChart	Object: активная диаграмма	Application, Window, Workbook
ActiveSheet	Object: активный лист	Application, Window, Workbook
Address	Возвращает координаты ячейки указанного объекта	Range
Cells	Диапазон объекта Range	Application, Range, Worksheet
Count	Integer: число объектов в коллекции	Все объекты коллекции
Formula	String: формула для ячейки рабочего листа	Диапазон
Name	String: имя объекта	Application, Workbook и в других объектах

ПРОДОЛЖЕНИЕ ТАБЛИЦЫ 2

Saved	Boolean: сохранялась ли рабочая книга после последних изменений	Workbook
Selection	Object: текущий выделенный фрагмент	Application, Window
StatusBar	String: сообщение в статусной строке	Application
ThisWorkBook	Object: рабочая книга, из которой выполняется текущая процедура	Application
Visible	Boolean: отображается или нет объект на экране	Application, Worksheet, Range и в других объектах
Value	(варьируется): действительное значение, отображаемое в ячейке	Range

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ОБЪЕКТА

Методы объекта используются в операторах VBA так, как использовались бы любые встроенные процедуры VBA.

Метод объекта имеет следующий синтаксис:

ИмяОбъекта.Метод

Для методов объектов, имеющих обязательные и необязательные аргументы, используется синтаксис:

ИмяОбъект.Метод Аргумент1, Аргумент2,...

Аргумент1, Аргумент2 и так далее представляют аргументы в списке аргументов метода.

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ОБЪЕКТА

Метод может иметь *один* или *несколько* *аргументов* в своем списке или не иметь их совсем; аргументы метода могут быть *обязательными* или *необязательными*.

Другой способ вызова метода заключается в использовании *именованных аргументов*. При вызове метода указываются названия аргументов, за которыми следует оператор := и сами передаваемые значения:

**ИмяОбъекта.Метод Аргумент1:=Значение1,
Аргумент2:=Значение2, ...**

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ОБЪЕКТА

Чтобы использовать значение, возвращаемое методом, необходимо поместить список аргументов метода в круглые скобки и включить вызов метода в оператор присваивания или другое выражение, точно так же, как при использовании функции. Можно игнорировать результат, возвращаемый методом. Для этого вызовите метод без круглых скобок вокруг списка аргументов, как если бы метод не возвращал результата.

Примеры использования возвращаемых значений методов объекта

```
MsgBox myRange.Address
```

```
MsgBox myRange.Address( xIPIC1)
```


ИСПОЛЬЗОВАНИЕ МЕТОДОВ ОБЪЕКТА

Метод **Address** в Excel, который принадлежит объекту **Range**, возвращает адрес диапазона ячеек в рабочем листе как строку. Если переменная **myRange** ссылается на первую ячейку в рабочем листе, то оператор **MsgBox** в приведенной выше строке примера отображает строку **\$A\$1**.

Во втором примере показан вызов метода **Address** с его третьим необязательным аргументом (который определяет стиль возвращаемых координат ячеек).

ТАБЛИЦА 3. НАИБОЛЕЕ ПОЛЕЗНЫЕ МЕТОДЫ В EXCEL

Метод	Назначение	Имеется в объектах
Activate	Активизирует объект	Window, Workbook, Worksheet, Range и в других объектах
Calculate	Выполняет вычисления в открытой рабочей книге, рабочем листе или диапазоне	Application, Range, Worksheet
Clear	Удаляет данные, сохраненные в указанном объекте	Range
Close	Закрывает указанный объект	Window, Workbook, Workbooks
Save	Сохраняет файл рабочей книги	Application, Workbook
SaveAs	Сохраняет указанный объект в другом файле	Workbook, Worksheet
Select	Выбирает указанный объект	Range, Sheets,

ИСПОЛЬЗОВАНИЕ МЕТОДОВ ОБЪЕКТА

Многие методы имеют большое количество аргументов. Чтобы получить список параметров метода, можно воспользоваться системой **Auto Quick Info**. Эта система работает следующим образом: как только вы наберете строку,

например

ActiveSheet.SaveAs () система подсказки выдаст на экран список всех параметров метода **SaveAs**, как это обычно делается для обычных функций.

Согласно выданному системой **Auto Quick**.

ОБЪЕКТНЫЕ ПЕРЕМЕННЫЕ

В дополнение к типам **Byte**, **Integer**, **Long**, **Single**, **Double** и **String** VBA также имеет тип **Object**. Переменные или выражения типа **Object** ссылаются на объект VBA или на объект, принадлежащий приложению, например Excel-объекты **Workbook**, **Worksheet** и **Range**. Как и в случае с другими типами VBA, можно объявлять переменные в модулях, процедурах и функциях с определенным типом **Object**.

Пример:

```
Dim myObject As Object
```


ОБЪЕКТНЫЕ ПЕРЕМЕННЫЕ

Можно задавать переменную **myObject**, чтобы она содержала ссылку на любой объект VBA или объект приложения. Если вы собираетесь использовать переменную типа **Object** для некоторых специфических типов объектов, можно также объявлять объектную переменную для этого специфического типа объекта:

Dim InstBook As Workbook

Объектную переменную **InstBook** можно использовать только для сохранения ссылок на объекты **Workbook**; при попытке присвоить переменной **InstBook** ссылку на объект **Range** или **Worksheet**.

ОБЪЕКТНОЕ ВЫРАЖЕНИЕ

Объектное выражение — это любое выражение VBA, которое определяет отдельный объект. Все объектные выражения должны вычисляться до единственной объектной ссылки {ссылки на объект}; объектные выражения используются с единственной целью — создание ссылок на специфические объекты в программах VBA.

ОБЪЕКТНОЕ ВЫРАЖЕНИЕ

Объектное выражение может состоять из *объектных переменных*, *объектных ссылок* или *объектного метода* или *свойства*, которое возвращает объект. Нельзя использовать переменные типа **Object** или объектные выражения в арифметических, логических или операциях сравнения. Объектная ссылка, созданная с помощью объектного выражения или сохраненная в объектной переменной, является только адресом, указывающим место в памяти компьютера, где сохранен объект, на который выполняется ссылка. Поскольку объектная ссылка — это адрес памяти, арифметические, логические операторы и операторы сравнения не имеют

ОБЪЕКТНАЯ ПЕРЕМЕННАЯ

Перед использованием объектной переменной для ссылки на объект необходимо задать эту переменную, чтобы она содержала ссылку на нужный объект. Присваивание объектной ссылки объектной переменной отличается от присваиваний других переменных; чтобы присвоить объектную ссылку объектной переменной, используйте оператор **Set**. Оператор **Set** имеет следующий синтаксис:
Set Переменная=Объект

ОБЪЕКТНАЯ ПЕРЕМЕННАЯ

Переменная — это любая объектная переменная или переменная типа **Variant**.

Объект — любая допустимая объектная ссылка; это может быть другая объектная переменная или объектное выражение. Если **Переменная** — переменная, объявленная с каким-либо определенным типом (например, **Range** или **Workbook**), этот тип должен быть совместим с объектом, на который ссылается **Объект**.

Пример использования объектных переменных

```
Dim InstSheet As Worksheet
```

```
Set InstSheet = Application.ActiveSheet
```

ОБЪЕКТНАЯ ПЕРЕМЕННАЯ

Чтобы задать отдельный объект в выражении или объектную переменную для ссылки на этот объект, можно использовать методы и свойства, возвращающие объекты, такие как свойства **ActiveWorkbook** и **ActiveSheet** объекта **Application** или метод **Cells** объекта **Worksheet**.

ССЫЛКА НА ОБЪЕКТЫ С ПОМОЩЬЮ **WITH...END WITH**

При написании кода программы можно встретить ссылку на один и тот же объект в нескольких операторах. VBA предоставляет особую структуру — структуру **With...End With**, позволяющую ссылаться на свойства или методы, которые принадлежат одному и тому же объекту, без задания всей объектной ссылки каждый раз. Структура **With...End With** имеет следующий синтаксис:

With Object

' операторы, использующие свойства и методы

Object

End With

ССЫЛКА НА ОБЪЕКТЫ С ПОМОЩЬЮ WITH...END WITH

Пример использования структуры With...End With.

```
Dim FName As String 'имя файла-копии  
With ActiveWorkbook
```

```
' Сформировать новое имя файла из исходного  
FName = Left(.Name, InStr(.Name, ".") - 1) &  
"_bp2.xls"
```

```
FName = Path & "\" & FName.SaveCopyAs  
Filename=FName
```

```
End With
```

```
End Sub
```

ССЫЛКА НА ОБЪЕКТЫ С ПОМОЩЬЮ WITH...END WITH

Сравните этот же код без использования структуры With...End With.

```
Dim FName As String ' имя файла-копии
' Сформировать новое имя файла из исходного
FName = Left(ActiveWorkbook.Name,
InStr(ActiveWorkbook.Name, ".") - 1) _ & "_bp.xls"
FName = ActiveWorkbook.Path & "\" & FName
ActiveWorkbook.SaveCopyAs Filename:=FName
End Sub
```


РАБОТА С КОЛЛЕКЦИЯМИ ОБЪЕКТОВ И КОНТЕЙНЕРАМИ ОБЪЕКТОВ

Коллекция (collection) объектов — это группа связанных объектов, таких как все рабочие листы в рабочей книге или все символы в параграфе. Объект в коллекции называется ***элементом (element)*** этой ***коллекции***.

Сама коллекция является объектом; коллекции имеют собственные свойства и методы. Каждая коллекция, *например*, имеет свойство **Count**, которое возвращает число элементов в коллекции.

РАБОТА С КОЛЛЕКЦИЯМИ ОБЪЕКТОВ И КОНТЕЙНЕРАМИ ОБЪЕКТОВ

Если в активной рабочей книге имеется 16 рабочих листов, то следующее выражение вычисляется до числа 16:

Application.ActiveWorkbook.Worksheets.Count

В этом выражении **Worksheets** — коллекция всех рабочих листов в рабочей книге, **ActiveWorkbook** — свойство Excel-объекта **Application**, возвращающее активную рабочую книгу, а **Count** — свойство коллекции **Worksheets**, возвращающее общее число рабочих листов в коллекции.

РАБОТА С КОЛЛЕКЦИЯМИ ОБЪЕКТОВ И КОНТЕЙНЕРАМИ ОБЪЕКТОВ

Контейнер (container) — это любой объект, содержащий один или несколько других объектов. В данном примере *Application* содержит объект, на который ссылается *ActiveWorkbook*, содержащий в свою очередь, коллекцию объектов **Worksheets**. Все контейнерные объектные ссылки соединяются вместе с помощью точки-разделителя (.) для образования одного объектного выражения.

РАБОТА С КОЛЛЕКЦИЯМИ ОБЪЕКТОВ И КОНТЕЙНЕРАМИ ОБЪЕКТОВ

Объект **Application** включает в себя коллекцию объектов **Книга (Workbook)**. Семейство книг называется **Workbooks** (в конец добавляется буква «s» в английском языке это обозначает множественное число). Обратиться с конкретной книге в приложении можно указав название семейства – **Workbooks** и в скобках имя или номер книги.

РАБОТА С КОЛЛЕКЦИЯМИ ОБЪЕКТОВ И КОНТЕЙНЕРАМИ ОБЪЕКТОВ

Например, **Workbooks** (“Книга.xls”) или **Workbooks(1)**. Объект **Workbook** является частью семейства **Workbooks**, хотя семейство **Workbooks** тоже является объектом. Кроме того, существует еще объект **ActiveWorkbook** – активная книга, который тоже является свойством объекта **Application**. Объекты и **ActiveWorkbook** являются одиночными объектами. Объект **Workbooks** - семейство.

РАБОТА С КОЛЛЕКЦИЯМИ ОБЪЕКТОВ И КОНТЕЙНЕРАМИ ОБЪЕКТОВ

Объект **Workbook** содержит набор свойств объектов:

- ✓ объект семейство (коллекция) **Worksheets** – это рабочие листы конкретной рабочей книги.
- ✓ объект семейство **Sheets** – это рабочие листы в активной рабочей книге.
- ✓ объект семейство **Windows** – это все окна в конкретной рабочей книге.
- ✓ объект **ActiveSheet** - это активный рабочий лист в активной рабочей книге.

РАБОТА С КОЛЛЕКЦИЯМИ ОБЪЕКТОВ И КОНТЕЙНЕРАМИ ОБЪЕКТОВ

Объект **Worksheet** является рабочим листом - элементом семейства **Worksheets**, но чтобы обратиться к конкретному листу из семейства надо указать имя или номер рабочего листа в коллекции. *Например* **Worksheets(“Лист1”)** или **Worksheets(2)**.

ОБЪЕКТ APPLICATION

Объект **Application** - это главный (корневой) объект в иерархии объектов Excel, представляющий само приложение Excel. Он имеет огромное число свойств и методов, позволяющих установить общие параметры приложения Excel. Кроме того, объект **Application** через свойство **WorksheetFunction** предоставляет возможность использовать в коде все встроенные функции рабочего листа. Это семейство возвращает **WorksheetFunction** объект, являющийся контейнером всех функций рабочего листа.

ОБЪЕКТ APPLICATION

Например, в следующем примере находится максимальное значение из диапазона A1:A4:

Макс=Application.WorksheetFunction.

Max(Range("A1:A4"))

Функции рабочего листа можно включать в код непосредственно через объект **Application**, опуская свойство **WorksheetFunction**. *Например, в*

следующем примере переменной Pi присваивается значение Пи, а переменной Сумма присваивается

значение суммы из диапазона A1:A4: Pi =

Application.Pi()

Сумма = Application.Sum(Range("A1:A4"))

СВОЙСТВА ОБЪЕКТА APPLICATION

Свойство	Описание
ActiveWorkbook, ActiveSheet, ActiveCell, Activechart	Возвращают активную рабочую книгу, лист ячейку, диаграмму. ActiveCell содержится в ActiveSheet, а ActiveSheet и ActiveChart в ActiveWorkbook. В следующем примере в активной ячейке устанавливается
	полужирный шрифт и в нее вводится строка текста "Отчет за май": <i>With ActiveCell</i> <i>.Font Bold = True</i> <i>.Value = "Отчет за май"</i> <i>End With</i>
ThisWorkbook	Возвращает рабочую книгу, содержащую выполняющийся в данный момент макрос. 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 (автоматически); - xlCalculationManual (вручную); - xlCalculationSemiAutomatic (автоматически, кроме таблиц);

СВОЙСТВА ОБЪЕКТА APPLICATION

Caption	Возвращает или устанавливает текст из заголовка главного окна Excel. Установка значения свойства равным Empty возвращает заголовок, используемый по умолчанию. В следующем примере первая инструкция устанавливает в качестве заголовка окна приложения текст "Отчет за 2000 год", а вторая возвращает окну имя, используемое по умолчанию, т. е. Excel Application.Caption = "Отчет за 2000 год" Application.Caption = Empty
DisplayScrollBars	Логическое свойство, регулирующее отображение полос прокрутки.
DisplayStatusBar	Логическое свойство, регулирующее отображение строки состояния..
Height и Width	Высота и ширина окна приложения в пунктах.
Left и Top	Расстояние в пунктах от левой границы окна приложения до левого и верхнего края экрана.
Right	Расстояние в пунктах от правой границы окна приложения до правого края экрана.
StatusBar	Возвращает или устанавливает текст, выводимый в строке состояния. В данном примере в строке состояния выводится "Ввод данных..." Application.DisplayStatusBar = True Application.StatusBar = "Ввод данных..."

МЕТОДЫ ОБЪЕКТА APPLICATION

Метод	Описание
Calculate	<p>Вызывает принудительное вычисление во всех открытых рабочих книгах, или специфицированном рабочем листе или диапазоне. В следующем примере первая инструкция приводит к перерасчету во всех открытых книгах, вторая инструкция - к перерасчету на указанном рабочем листе активной рабочей книге, а третья - в указанном диапазоне:</p> <ul style="list-style-type: none">- Application.Calculate- Worksheets("Отчет").Calculate- Worksheets("Отчет ").Range("A1:C10").Calculate
Help	<p>Отображает справку.</p> <p>Синтаксис: Help(HelpFile, HelpContextID)</p> <ul style="list-style-type: none">- HelpFile - имя HLP-файла. Если значение этого параметра не указано, то отображается файл справки Microsoft Excel;- HelpContextID - номер раздела справки. Если значение этого параметра не указано, то отображается оглавление справки. <p>Application.Help "Notepad.hlp"</p>

МЕТОДЫ ОБЪЕКТА APPLICATION

Volatile	Вызывает перевычисление функции пользователя при изменениях в ячейках рабочего листа. Например, Function Квадрат(x) Application.Volatile Квадрат = x^2 End Function
OnKey	Устанавливает выполнение специфицированной процедуры при нажатии заданной комбинации клавиш.

ОБЪЕКТ **WORKBOOK** И СЕМЕЙСТВО **WORKBOOKS**

В иерархии **Excel** объект **Workbook** 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.

ОБЪЕКТ WORKBOOK И СЕМЕЙСТВО WORKBOOKS

Свойство	Описание
ActiveSheet	Возвращает активный лист книги. В следующем примере устанавливается имя активного рабочего листа: <code>ActiveSheet.Name = "Отчет"</code>
Sheets	Возвращает семейство всех листов книги.
Worksheets	Возвращает семейство всех рабочих листов книги.
Name	Возвращает или устанавливает имя книги.
Path	Возвращает полное имя папки, в которой находится книга.
FullName	Возвращает полное имя книги, включая путь. Например, <code>Имя = ActiveWorkbook.FullName</code> .
Saved	Логическое свойство, которое устанавливает, не производились ли изменения в книге со времени ее последнего сохранения.
WriteReserved	Логическое свойство, которое устанавливает, закрыта ли книга для записи.

МЕТОДЫ ОБЪЕКТА WORKBOOK И СЕМЕЙСТВА WORKBOOKS

Метод	Описание
Activate	Активизирует рабочую книгу так, что ее первый рабочий лист становится активным. Например, <code>Workbooks("Отдел кадров").Activate</code>
Add	<p>Создает новый объект в семействе <code>Workbooks</code>.</p> <p>Синтаксис: <code>Add (Template)</code> где <code>Template</code> - необязательный. Задаёт шаблон, на основе которого создается новая рабочая книга. Допустимые значения:</p> <ul style="list-style-type: none">- <code>xlWBATChart</code>;- <code>xlWBATExcel4IntlMacroSheet</code>;- <code>xlWBATExcel4MacroSheet</code>;- <code>xlWBATWorksheet</code>. <p>Если аргумент <code>Template</code> опущен, то создается новая рабочая книга с количеством листов, заданных свойством <code>SheetsInNewWorkbook</code>.</p>

МЕТОДЫ ОБЪЕКТА WORKBOOK И СЕМЕЙСТВА WORKBOOKS

Protect	<p>Защищает рабочую книгу от внесения в нее изменений. Синтаксис:</p> <ul style="list-style-type: none">- Protect (Password, Structure, Windows)- Password - необязательный. Строка, используемая в качестве пароля для защиты книги. Если параметр опущен, то книга защищена без пароля;- Structure - необязательный. Логический параметр, который устанавливает, защищена ли структура книги, т.
	<p>е. взаимное расположение листов;</p> <ul style="list-style-type: none">- Windows - необязательный. Логический параметр, который устанавливает, защищено ли окно книги. <p>В следующем примере устанавливается защита для активной рабочей книги:</p> <pre>ActiveWorkbook.Protect Password:= "ВинниПух"</pre>
Unprotect	<p>Снятие защиты с рабочей книги. Синтаксис:</p> <pre>Unprotect(Password)</pre> <p>где Password - необязательный. Строка, используемая в качестве пароля для защиты листа. В следующем примере снимается защита с активной книги:</p> <pre>ActiveWorkbook.Unprotect Password:= "ВинниПух"</pre>
Close	<p>Закрытие книги.</p>

МЕТОДЫ ОБЪЕКТА WORKBOOK И СЕМЕЙСТВА WORKBOOKS

Open	<p>Открытие существующей книги. Синтаксис: Open(FileName, Readonly, Password, Converter, AddToMRU)</p> <ul style="list-style-type: none">- FileName - обязательный. Имя открываемого файла;- Readonly - необязательный. Логический параметр, задающий открытие файла в режиме, доступном только для чтения;- Password - необязательный. Строка с паролем для защищенной книги;- Notify - необязательный. Логический параметр, задающий, надо ли извещать пользователя о том, что файл доступен в режиме только для чтения;- AddToMRU - необязательный. Логический параметр, задающий, надо ли добавить открываемый файл в список недавно использованных файлов. <p>Пример: Workbooks.Open "000 Рога и Копыта"</p>
Save	<p>Сохраняет книгу. Пример: ActiveWorkbook.Save.</p>

ОБЪЕКТ WORKSHEET И СЕМЕЙСТВО WORKSHEETS

В иерархии Excel объект worksheet идет сразу после объекта workbook и представляет рабочий лист.

Свойства объекта **WORKSHEET** и семейства **WORKSHEETS**

Свойства	Описание
Name	Возвращает имя рабочего листа: Worksheets(1).Name="Итоги"
Visible	True (False) – рабочий лист видим (невидим) на экране.
Range	Возвращает ссылку на указанный диапазон ячеек. Например: ActiveSheet.Range("B1")
UsedRange	Возвращает диапазон ячеек рабочего листа.
ActiveCell	Возвращает активную ячейку рабочего листа.

МЕТОДЫ ОБЪЕКТА WORKSHEET И СЕМЕЙСТВА WORKSHEETS

Методы	Описание
Activate	Активизирует рабочий лист: Worksheet(2).Activate
Add	Создает новый рабочий лист. Параметры: Before – лист, перед которым будет размещен новый лист; After – лист после которого будет помещен новый лист; Count – число добавляемых листов; Type – тип добавляемого листа. Например, ActiveWorkbook.Worksheets.Add
Delete	Удаляет рабочий лист: Worksheets(1).Delete
Evaluate	Преобразует текстовую строку в объект Excel или значение. Используется, например, для ввода ссылок на ячейки: MyCell = InputBox("Введите имя ячейки") Evaluate(myCell).Value = "Новое значение"
Copy	Копирование активного рабочего листа в другое место рабочей книги: Worksheets("Лист2"). _ Copy After:= Worksheets("Лист3")
Move	Перемещение активного рабочего листа в другое место рабочей книги: Worksheets("Лист2"). _ Move After:= Worksheets("Лист3")

ОБРАБОТЧИКИ СОБЫТИЙ

Событие — это действие, распознаваемое объектом, для которого можно запрограммировать отклик.

Например, в качестве события можно использовать открытие (закрытие) документа, щелчок мыши, нажатие клавиши. Набор действий или повторяющихся явлений, которые можно сопоставить с кодом VBA, называется *событиями*, а специальный тип процедуры, которая выполняется при возникновении события, называется *обработчиком событий*.

ОБРАБОТЧИКИ СОБЫТИЙ

Обработать можно события следующих объектов

Excel:

- ✓ Application
- ✓ Workbook
- ✓ Worksheet
- ✓ Chart

Обработчики событий дают возможность привязать код к действиям пользователя, например к открытию или закрытию книги, активации таблицы, сохранению документа.

Обработчики событий создаются в модулях лисов или книги (в зависимости от того, с каким объектом будет связано это событие).

ОБРАБОТЧИКИ СОБЫТИЙ

В некоторых ситуациях события для объектов не появляются в окне редактора кода (например, Application). В этом случае необходимо явно объявить этот объект с событиями — при помощи ключевого слова **WithEvents**, например так:

Public WithEvents App As Excel.Application

Делается это в области объявлений модуля (Declarations). После этого в редакторе кода Visual Basic появляется новый объект **App** со всеми необходимыми событиями.

СОБЫТИЯ ОБЪЕКТА APPLICATION

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

СОБЫТИЯ ОБЪЕКТА WORKBOOK

Событие	Описание
BeforeClose	При закрытии 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

СОБЫТИЯ ОБЪЕКТА WORKSHEET

Событие	Описание
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
Sheet Deactivate	Когда рабочий лист теряет фокус