**РАБОТА С ОБЪЕКТАМИ**

**Использование объектов**

Операторы программ VBA, использующие объекты, обычно выполнявши одно или несколько 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.

Например, можно определить имя активного в данный момент рабочего листа в Excel, выполняя выборку строки, сохраняемой в свойстве **Name** рабочего листа. (Свойство **Name** рабочего листа содержит имя рабочего листа, как показано на ярлыке листа.) Чтобы изменить имя рабочего листа, необходимей присвоить новую строку свойству **Name** этого рабочего листа. Для добавления рабочего листа в рабочую книгу используется метод **Add** рабочей книги.

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:

Object.identifier

Object – любая допустимая ссылка на объект. Объектные ссылки создают заданием переменной для ссылки на объект или использованием методов или свойств объектов, возвращающих объектную ссылку.

Identifier – любое допустимое имя свойства или метода; VBA отображает сообщение о runtime-ошибке при попытке использовать свойства или методы, которые не являются в действительности частью указанного объекта.

Например Word-процедура, выводящая имя активного документа имеет вид:

**Пример 1.** Активный документ

Sub ActDocum()

MsgBox "Активный документ: " & ActiveDocument.Name

End Sub

Точка (.) отделяет объектную ссылку от имени свойства или метода. В некотором смысле эта точка-разделитель также соединяет объектную ссылку с идентификатором свойства или метода. Поскольку идет обращение к свойству или методу посредством объекта, необходимо определять объектную ссылку и идентификатор свойства или метода вместе. Точка-разделитель указывает VBA, где заканчивается объектная ссылка и где начинается идентификатор свойства или метода. В то же самое время точка-разделитель соединяет объектную ссылку и имя свойства или метода для образования единого идентификатора в операторе VBA.

В табл. 1 приведены несколько из наиболее важных объектов (с точки рения программиста VBA) в Excel 2000/2002. В таблице показано имя объекта и краткое описание этого объекта.

Таблица 1 Общие объекты Excel

|  |  |
| --- | --- |
| **Объект** | **Описание** |
| Application | Приложение Excel |
| Chart | Диаграмма в рабочей книге |
| Font | Объект, содержащий 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 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: *получатъ* значение свойства или *устанавливать* его. Не все свойства объекта изменяемы. Свойства объектов, которые нельзя изменять, называют *свойствами, доступными только не чтение* (*read-only*), свойства, которые можно устанавливать, называют *свойствами, доступными на чтение/запись* (*read-write*).

Свойства обычно содержат численные, строковые, значения типа Boolean, хотя некоторые свойства могут возвращать значения типа Object или другие типы данных.

Обращение к свойству объекта имеет следующий синтаксис:

Object.property

Object –допустимая объектная ссылка VBA;

property –любое допустимое имя свойства для объекта, на который выполняется ссылка.

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 *получить значение свойства* и его присвоить некоторой переменной значение свойства объекта, использовать следующий синтаксис:

Variable = Object.Property

Variable – любая допустимая переменная, имеющая совместимый со свойством объекта тип;

Object – любая допустимая ссылка на объект;

Property – любое допустимое имя свойства для объекта, на который выполняется ссылка.

Пример 2.

**Sub** gg()

**Dim** AnyStr **As** String

AnyStr = ActiveSheet.Name

**MsgBox** AnyStr

**End Sub**



Строка, сохраняемая в свойстве Name рабочего листа Excel и на которую ссылается объектная переменная ActiveSheet, присваивается переменной AnyStr.

Можно также использовать свойство объекта непосредственно в каком-либо выражении или в качестве аргумента функции или процедуры. Следующие строки представляют обоснованное использование свойства объекта (в каждой строке ActiveSheet.Name является объектной переменной, заданной для ссылки на рабочий лист Excel):

MsgBox ActiveSheet.Name

AnyStr = "Эта книга имеет имя: " & ActiveSheet.Name

MsgBox LCase(ActiveSheet.Name)

Почти каждый объект в VBA имеет свойство, которое содержит его имя. Следующий оператор использует **MsgBox** для отображения свойства **FullName** – объекте рабочей книги Excel; свойство **FullName** содержит имя диска, путь папке и имя файла рабочей книги:

MsgBox ThisWorkbook.FullName

В приведенном выше примере ThisWorkbook– это переменная, заданная для ссылки на объект открытой рабочей книги. Если ThisWorkbookссылается на рабочую книгу с именем **Sales.xls** в папке My **Documents,** то окно сообщения, вызываемое приведенным выше оператором, отображает строку **"C:\My Documents \SALES.XLS".**

Чтобы *задать (установить) свойство объекта*, надо присвоить свойству новое значение, используя следующий синтаксис:

Object.Property = Expression

Object – любая допустимая объектная ссылка;

Property – любое свойство объекта, на который выполняется ссылка;

Expression – любое выражение VBA, которое вычисляется до типа, совместимого со свойством.

Изменить имя рабочего листа, на который ссылается объектная переменная InstSheet**,** присваивая значение свойству Nameлиста можно следующим образом:

Sub rename()

ActiveSheet.Name = "Первый квартал"

End Sub

Изменить текст, отображаемый в строке состояния в нижнем левом углу окна приложения, присваивая строку свойству StatusBarобъекта Application(объект Application– это host-приложение VBA, в данном случае – Excel) можно так:

Sub rename()

Application.StatusBar = "Генерировать отчет за третий квартал"

Application.StatusBar = False

End Sub

Свойство **Application.StatusBar** можно использовать в своих процедурах для отображения сообщений о действиях, которые выполняет процедура, особенно если некоторые из этих действий занимают много времени (например, сортировка длинного списка). Добавляя сообщение в строку состояния, пользователю дается знать, что процедура все еще работает.

Когда процедура выполнена надо возвратиь управление строке состояния при работе в Excel. Для этого необходимо устанавливать свойство **Application.StatusBar** в **False**, иначе Excel продолжает отображать заданное сообщение строки состояния.

В табл. 2 перечислены некоторые из наиболее употребительных или полезных свойств объектов в Excel версии Visual Basic for Application. В таблице представлено имя свойства, тип и значение, а также объекты, которые имеют это свойство.

Таблица 2. Наиболее употребительные свойства объектов

|  |  |  |
| --- | --- | --- |
| **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 |
| Charts | Коллекция диаграмм | Application, Workbook  |
| Count | Integer: число объектов в коллекции | Все объекты коллекции |
| Fofmula | String: формула для ячейки рабочего листа | Диапазон |
| Index | Integer: число объектов в коллекции | Worksheet |
| Name | String: имя объекта | Application, Workbook и в других объектах |
| Path | String: драйвер и каталог, в котором сохранен объект | Addln, Application, Workbook |
| Saved | Boolean: сохранялась ли рабочая книга после последних изменений | Workbook |
| Selection | Object: текущий выделенный фрагмент | Application, Window |
| Sheets | Коллекция листов рабочей книги | Application, Workbook |
| StatusBar | String: сообщение в статусной строке | Application |
| ThisWorkBook  | Object: рабочая книга, из которой выполняется текущая процедура | Application |
| Type | Integer: число, указывающее тип объекта | Window, Worksheet, Chart |
| Visible | Boolean: отображается или нет объект на экране | Application, Worksheet, Range и в других объектах |
| Value | (варьируется): действительное значение, отображаемое в ячейке | Range |
| Workbooks | Коллекция рабочих книг | Application |
| Worksheets | Коллекция рабочих листов  | Application, Workbook |

**Пример 3.** Активный лист, ячейка

**Sub** TestObjectOl()

'имя активного листа

MsgBox ActiveSheet.Name

'адрес активной ячейки:

MsgBox ActiveCell.Address

'содержимое активной ячейки

MsgBox ActiveCell.Formula

'путь сохранения документа

MsgBox ThisWorkbook.Path

'отображение полного имени файла

MsgBox ThisWorkbook.FullName

'закрытие активного документа

ThisWorkbook.Close

**End Sub**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

|  |  |
| --- | --- |
|  |  |

**Использование методов объектов**

 Методы объекта используются в операторах VBA также как любые встроенные процедуры.

Метод объекта имеет следующий синтаксис:

Object.Method

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

Object.Method Argumentl, Argument2, Argument3...

Object – любая допустимая объектная ссылка;

Method – имя любого метода, принадлежащего объекту, на который выполняется ссылка;

Argument1, Argument2 – аргументы в списке аргументов метода. Как в случае с аргументами для вызова процедуры VBA, необходимо перечислять аргументы метода в определенном порядке, отделяя каждый аргумент списке запятой и включая *отмечающие запятые* для пропущенных необязательных аргументов. Метод может иметь один или несколько аргументов в своем списке или не иметь их совсем; аргументы метода могут быть обязательными или необязательными.

**Пример 4.** Рабочие книги Excel имеют метод Activate, который делает рабочую книгу активной и активизирует первый лист в книге. Если задать переменную ThisWorkbookдля ссылки на объект рабочей книги, то приведенный ниже оператор активизирует эту рабочую книгу:

ThisWorkbook.Activate

Хотя метод Activate не имеет аргументов, многие методы объектов имеют один или более аргументов, например метод **SaveAs** объекта рабочей книги Excel; использует один обязательный аргумент и один из нескольких необязательных аргументов:

ThisWorkbook.SaveAs Filename:="C\Temp\NEWFILE.xls"

 В таблице 3. приведены наиболее употребительные методы в Excel VBA

| **Метод** | **Назначение** | **Имеется в объектах** |
| --- | --- | --- |
| Activate | Активизирует объект | Window, Workbook, Worksheet, Range и др. 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  |
| Justify | Выравнивает текст, сохраненный в указанном объекте  | Range  |
| Run | Выполняет указанную процедуру или функцию  | Application, Range  |
| Save | Сохраняет файл рабочей книги  | Application, Workbook  |
| SaveAs | Сохраняет указанный объект в другом файле  | Workbook, Worksheet  |
| Select | Выбирает указанный объект  | Range, Sheets, Worksheets  |
| SendKeys | Пересылает нажатия клавиши в диалоговые окна в host-приложении  | Application  |
| Volatile | Регистрирует функцию как *изменяющуюся* | Application  |

**Пример 5.** Метод Clear для ячейки и диапазона

**Sub** TestMethodsOl()

'тестирование метода Clear

'содержимое активной ячейки до применения метода Clear

MsgBox ActiveCell.Formula, , "До метода Clear"

'выполнения Clear

ActiveCell.Clear

'содержимое активной ячейки

'после применения метода Clear

MsgBox ActiveCell.Formula, , "После метода Clear"

'Очистить диапазон ячеек:

Range("C6:D7").Clear

**End Sub**



|  |  |
| --- | --- |
|  |  |



**Объявление объектных переменных**

В дополнение к типам Byte, Integer, Long, Single, Double и StringVBA также имеет тип Object. Переменные или выражения типа Object ссылаются на объект VBA или на объект, принадлежащий host-приложению, такой как Excel-объекты Workbook, Worksheet и Rangeили Word-объекты Document, Range, Paragraph.

Как и в случае с другими типами VBA, можно объявлять переменные в мо­дулях, процедурах и функциях с определенным типом Object, что показано в следующем операторе:

**Dim** myObject **As** Object

Если переменная типа Object используется для некоторых специфических типов объектов, можно также объявлять объектную переменную для этого специфического типа объекта:

**Dim** InstBook **As** Workbook

Объектную переменную InstBook, создаваемую этим оператором **Dim**, можно использовать только для сохранения ссылок на объекты Workbook**. П**ри попытке присвоить переменной InstBookссылку на объект Rangeили WorksheetVBA отображает сообщение об ошибке несовпадения типов.

**Объект в выражениях**

*Объектное выражение* – это 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 может состоять из объектных переменных, объектных ссылок или объектного метода или свойства, которое возвращает объект:

|  |  |
| --- | --- |
| **Application** | **Имя объекта: ссылается на объект приложения** |
| Application.ActiveSheet  | Объектное свойство, которое возвращает ссылку на объект: активный лист  |
| Application.Workbooks  | Объектный метод, который возвращает коллекцию объектов: все открытые рабочие книги  |
| ABook | Объектная переменная: инициализированная в операторе Set, ссылается на 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, в действительности, 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 смысла.

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** Var = Object

Var – это любая объектная переменная или переменная типа Variant;

Object – любая допустимая объектная ссылка; это может быть другая объектная переменная или объектное выражение.

Если Var – переменная, объявленная с каким-либо определенным типом (например, Range или Workbook), этот тип должен быть совместим с объектом, на который ссылается *Object.*

**Dim** InstSheet **As** Worksheet

**Set** InstSheet = Application.ActiveSheet

Чтобы задать отдельный объект в выражении или объектную переменную для ссылки на этот объект, используются методы и свойства, возвращающие объекты, такие как свойства ActiveWorkbookи ActiveSheetобъекта Application или метод Cells объекта Worksheet(в Excel).

Хотя стандартные операторы сравнения (<, <=, >, >=, о, =) не являются значимыми при использовании с объектами, VBA предоставляет один опера­тор сравнения, предназначенный исключительно для использования с объектными выражениями и переменными – оператор Is:

Objectl **Is** Object2

Object1 и Object2– любые допустимые объектные ссылки. Оператор Is используется для определения того, обозначают ли две объектные ссылки один и тот же объект. Результат операции сравнения Is равен True,если объектные ссылки являются одними и теми же, иначе – False.

**Пример 6.** Процедура создания резервной копии активной рабочей книги

Sub SaveActiveBook()

'Создает копию активной рабочей книги под новым именем,

'используя метод SaveCopyAs. Новое имя имеет добавку "\_bp"

Dim FName As String 'имя файла-копии

Dim OldComment As String 'комментарии

'сохранить комментарии исходного файла

OldComment = ActiveWorkbook.Comments

'добавить новые комментарии к backup-файлу

ActiveWorkbook.Comments = "Резервная копия файла " & ActiveWorkbook.Name & ", выполненная процедурой SaveActiveBook"

'Сформировать новое имя файла из исходного

FName = Left(ActiveWorkbook.Name, InStr(ActiveWorkbook.Name, ".") - 1) & "\_bp.xls"

' добавление к имени файла пути

FName = ActiveWorkbook.Path & "\" & FName

ActiveWorkbook.SaveCopyAs Filename:=FName

ActiveWorkbook.Comments = OldComment ' восстановить комментарии

End Sub

В этой процедуре используются два объекта Excel, их свойства и методы. Объектная ссылка ActiveWorkbookявляется свойством Excel-объекта Application, которое возвращает объектную ссылку на активную в данный момент рабочую книгу.

В строке OldComment =… копируется строка свойства Commentsобъекта ActiveWorkbook в переменную OldComment. Затем в строках ActiveWorkbook.Comments = … задается новое значение для свойства Commentsобъекта ActiveWorkbook(это свойство изменяется и для исходной рабочей книги). Свойство Commentsэтого объекта содержит текст комментария, обычно вводимого диалоговое Properties(Окно свойств), которое Excel отображает при сохранении рабочей книги в первый раз или при использовании команды File | Properties(Файл | Свойства).