The background features several large, overlapping, semi-transparent swirls in shades of purple, green, and blue. Scattered throughout are numerous small, yellow, triangular shapes that resemble confetti or starbursts.

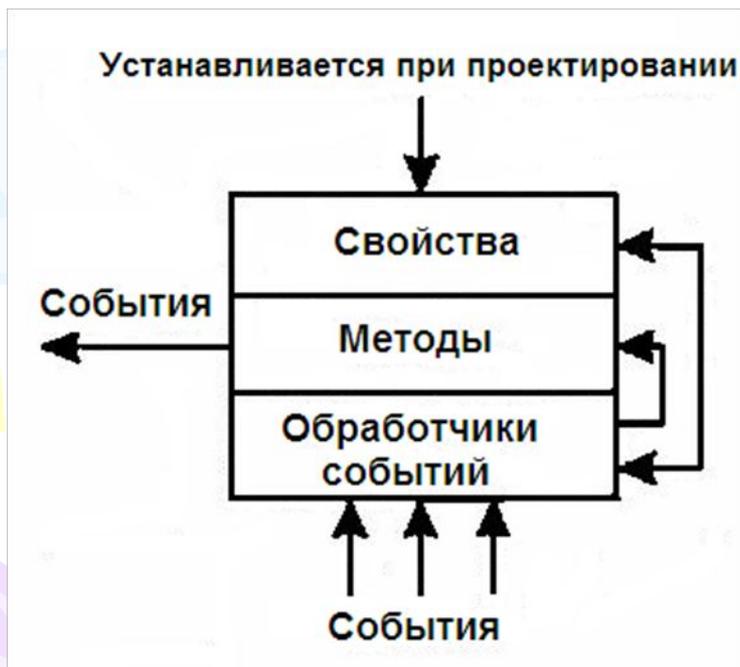
# **Объектная модель Excel**

# Объектная модель

VBA относится к языкам объектно-ориентированного программирования (ООП). ООП можно описать как методику анализа, проектирования и написания приложений с помощью объектов. Что такое объект? Объект позволяет инкапсулировать данные вместе с кодом, предназначенным для их обработки, т. е. объединить их в нечто целое, именуемое объектом. VBA не является объектно-ориентированным языком в строгом понимании этого слова, однако объектный подход играет в нем большую роль. Все визуальные объекты, такие как *рабочий лист (Worksheet)*, *диапазон (Range)*, *диаграмма (Chart)*, *форма (UserForm)*, являются объектами.

# Объектная модель

**Объект это совокупность свойств и методов, а также событий, на которые он может реагировать.**



- *Свойства* (property) – определяет внешний вид объекта и его поведение в вашей программе.
- *Методы* (method) – позволяют разрабатываемому приложению выполнять определенные действия над объектом.
- *События* (events) – описывает внешние воздействия, на которые реагирует объект при выполнении программы

# Объектная модель

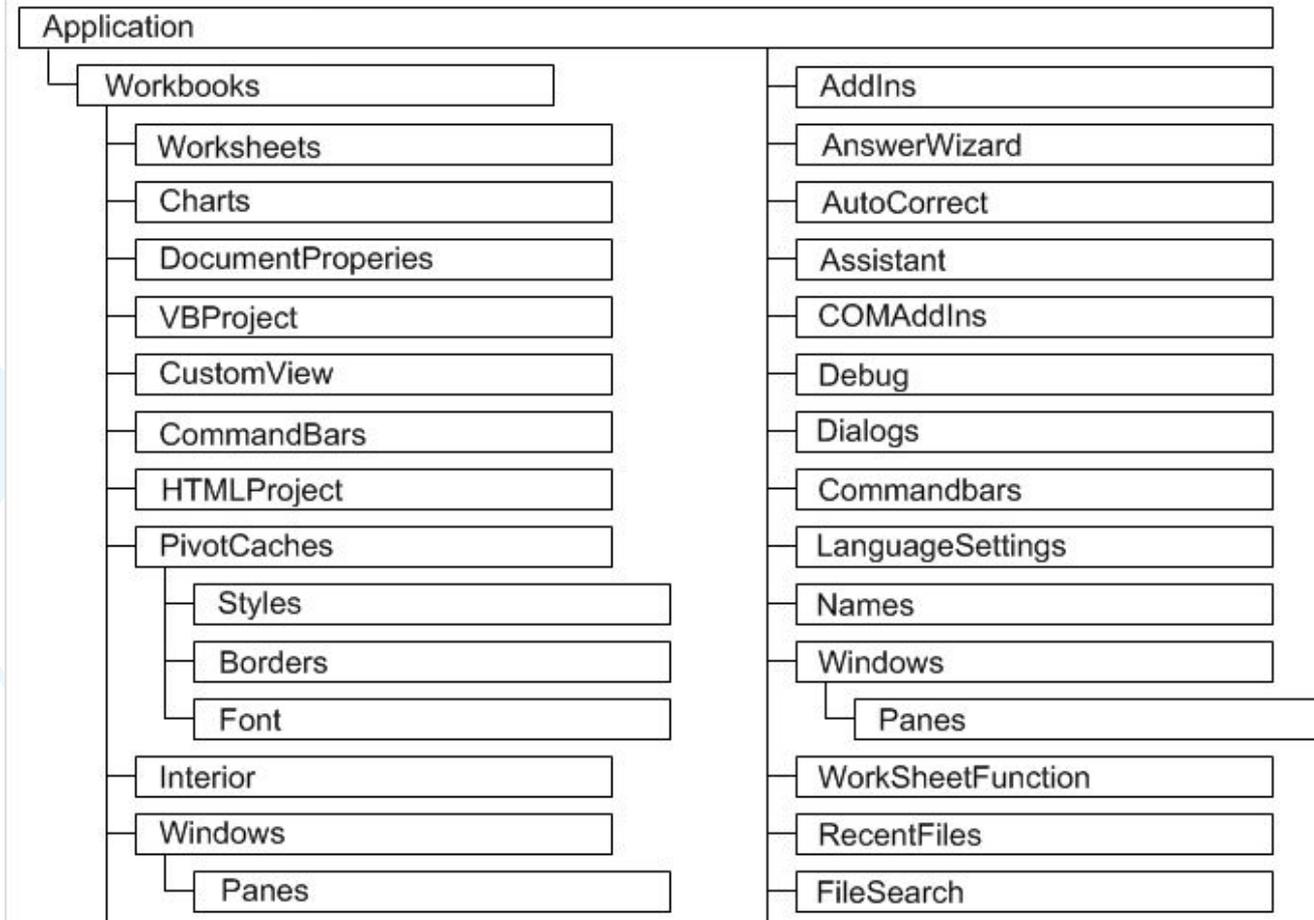
В объектной модели Excel и других приложений Office объекты связаны между собой отношением встраивания.

На нулевом уровне иерархии существует некоторый центральный объект, в который встроены другие объекты, составляющие первый уровень иерархии.

В каждый из объектов первого и последующих уровней могут быть встроены объекты следующего уровня. Так этот процесс продолжается. Таким образом, объекты в этой модели "толстые", поскольку в них встроено большое число других объектов. В особенности это касается объектов, стоящих на верхних уровнях иерархии.

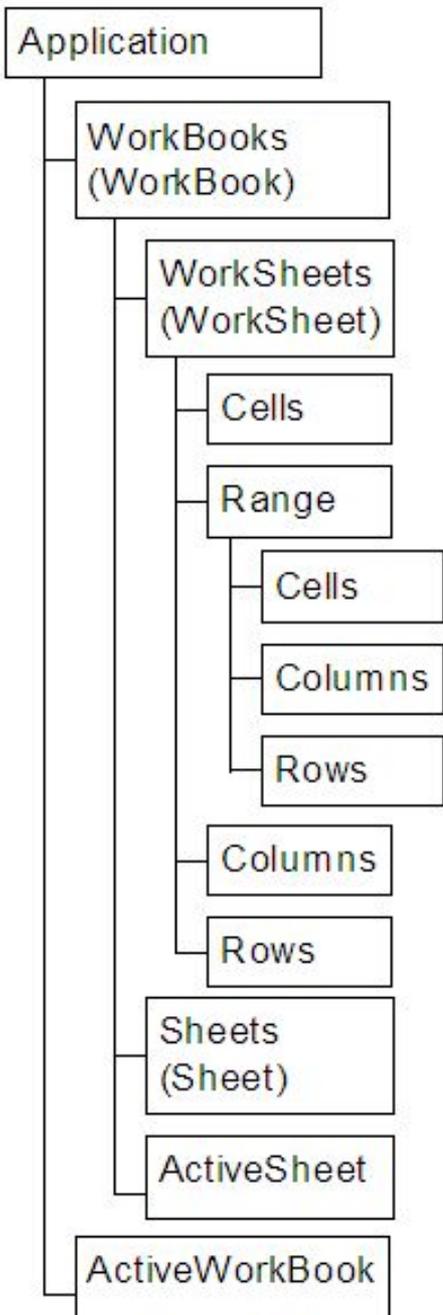
Формально встраивание реализуется с помощью свойств объектов. Свойства могут быть как терминальными, не являющимися объектами, и так называемыми свойствами - участниками, которые возвращают объекты при их вызове.

# Объектная модель Excel



# Объектная модель Excel

*Семейство* (Коллекция) представляет собой объект, содержащий несколько других объектов, как правило, одного и того же типа. Например, объект `Workbooks` (рабочие книги) содержит все открытые объекты `Workbook` (рабочая книга). Каждый элемент семейства нумеруется и может быть идентифицирован либо по номеру, либо по имени. Например, `Worksheets (1)` обозначает первый рабочий лист активной книги, а `Worksheets ("Лист1")` — рабочий лист с именем лист1.



# Свойства объекта Application

Объект Excel.Application задает приложение Excel. А посему свойства, методы и события этого объекта должны характеризовать приложение в целом. Понятно, что у этого объекта должно быть свойство Workbooks, возвращающее все открытые в приложении рабочие книги, свойство Windows, возвращающее открытые окна, свойства, такие как CommandBars, возвращающие объекты интерфейса, и другие подобные свойства. У объекта Excel.Application очень большое число свойств, методов и событий

# Свойства - участники объекта

- **WorkBooks** - Коллекция открытых в Excel документов - рабочих книг. Основной объект, благодаря которому можно получить доступ к любому документу Excel и далее работать с объектами этой рабочей книги.
- **Windows** - Коллекция открытых окон во всех рабочих книгах. Дело в том, что одну и ту же рабочую книгу часто полезно открывать в нескольких окнах, что позволяет видеть разные участки рабочей книги. Коллекция Windows позволяет получить доступ к каждому такому окну.
- **Группа активных объектов** - **ActiveWorkbook, ActiveWindow, ActiveSheet, ActiveChart, ActiveCell, ActivePrinter**, - возвращающих активную рабочую книгу, окно, активную рабочую страницу, диаграмму или ячейку, если таковые существуют в момент вызова соответствующего свойства. При отсутствии запрашиваемого активного объекта возникнет ошибка. Особняком стоит свойство, возвращающее активный принтер. Это свойство действительно имеет смысл связать с приложением. Заметьте, что объекты, стоящие на нижних уровнях иерархии, например, **Workbook**, этим свойством не обладают, так что добраться до принтера можно только через объект **Application**.

# Свойства - участники объекта

Краткий обзор тех вложенных в Excel.Application объектов, доступных на этом уровне, по сути, относящихся к нижним уровням иерархии объектной модели Excel:

- Группа коллекций и объектов Range, входящих в состав соответствующего активного объекта - Sheets, Charts, Rows, Columns, Cells, Range - возвращающие соответственно коллекции рабочих страниц, страниц диаграмм активной рабочей книги, объект Range, содержащий все строки, столбцы, ячейки или заданную область активной рабочей страницы. Также как и в случае вызова объектов предыдущей группы, при вызове этих свойств следует быть осторожным, поскольку возникает ошибка, если нет соответствующего активного объекта.
- Свойство Selection возвращает выделенный объект в активном окне. Тип возвращаемого объекта зависит, от текущего выделения. Возвращается Nothing, если в активном окне нет выделенного объекта.
- Свойство ThisWorkbook возвращает текущую рабочую книгу, содержащую выполняемый макрос, один из операторов которого и вызвал это свойство.

# Терминальные свойства

В ниже приведенном обзоре представлено выборочное описание некоторых групп терминальных свойств:

- Группа свойств, задающих свойства приложения по умолчанию, - `DefaultFilePath`, `DefaultSaveFormat`, - путь по умолчанию, формат по умолчанию.
- Группа булевых свойств, позволяющих включить или выключить отображение на экране тех или иных элементов приложения - `DisplayAlerts`, `DisplayCommentIndicator`, `DisplayFormulaBar`, `DisplayStatusBar` и другие `Display`-свойства. Первое из этих свойств позволяет управлять выдачей на экран некоторых сообщений в процессе работы макросов, второе - отображать специальный индикатор при показе комментариев. Более часто приходится использовать управление показом панелей формул и статуса.
- Группа свойств, управляющих размерами главного окна приложения Excel - `Height`, `Width`, `Left`, `Top`, задающие высоту, ширину окна и координаты верхнего левого угла окна.

# Методы Application

- Методы, запускающие вычисления - Calculate, CalculateFull, приводят к перевычислению рабочих страниц всех рабочих книг. Метод CheckSpelling запускает проверку орфографии во всех рабочих книгах.
- Метод Evaluate(Name) преобразует имя объекта в сам объект.  
Группа On-методов, позволяющих запустить на выполнение некоторый макрос.
- Метод OnKey(Key, Procedure) позволяет запустить макрос, заданный вторым параметром метода, при нажатии пользователем комбинации клавиш, заданной первым параметром метода.
- Метод OnTime(EarliestTime, Procedure As String, [LatestTime], [Schedule]) позволяет запустить макрос, заданный вторым параметром метода, в указанное время.

# Работа с рабочими книгами MS Excel

Основные действия, выполняемые над рабочими книгами:

Действие	Объект	Метод
Открытие	Workbooks	Open
Добавление	Workbooks	Add
Заккрытие	Workbooks, Workbook	Close
Сохранение	Workbook	Save, SaveAs, SaveCopyAs
Активизация	Workbook	Activate

# Свойства коллекции **Workbooks**

- **Свойство Count** - возвращает число объектов, содержащихся в коллекции. В данном случае значение этого свойства определяется числом открытых рабочих книг.
- **Свойство Parent** - возвращает объект-родитель для данного объекта.
- **Свойство Item** - возвращает конкретный объект **Workbook** из коллекции **Workbooks**. Синтаксис свойства **Item**: `Expression.Item(Index)`

Элемент	Описание
Expression	Выражение, которое возвращает объект <b>Workbooks</b>
Index	Обязательный параметр типа <b>Variant</b> , который представляет собой имя или порядковый номер рабочей книги в коллекции <b>Workbooks</b>

# Работа с рабочими листами MS Excel

MS Excel позволяет оперировать с несколькими типами листов:

- *Рабочие листы*, содержащие ячейки. Эти листы представляются объектом `Worksheet`.
- *Листы диаграмм*, содержащие диаграммы. Такие листы представляются объектом `Chart`.

Все рабочие листы (объекты `Worksheet`) содержатся в коллекции `Worksheets`. Для получения к ним доступа можно использовать коллекцию `Sheets`, которая содержит все листы рабочей книги, включая и рабочие листы, и листы диаграмм.



# Методы рабочих листов

Действие	Объект	Метод
Добавление рабочего листа	Sheets, Worksheets	Add
Активизация рабочего листа	Sheet, Worksheet	Activate
Перемещение рабочего листа	Sheet, Worksheet	Move



# Методы коллекции Worksheets

**Метод Add** - добавляет рабочий лист в рабочую книгу. Синтаксис метода Add:

Expression.Add (Before, After, Count, Type)

Элемент	Описание
Expression	Любое выражение, которое возвращает коллекцию Worksheets
Before	Необязательный параметр типа Variant. Определяет лист, перед которым будет вставлен добавляемый лист
After	Необязательный параметр типа Variant. Определяет лист, после которого будет вставлен добавляемый лист. В том случае если оба параметра Before и After отсутствуют, то лист будет вставлен перед активным листом
Type	Необязательный параметр типа Variant. Предназначен для определения типа добавляемого листа. Значение этого параметра может равняться значению одной из констант: xlWorksheet, xlExcel4MacroSheet или xlExcel4IntlMacroSheet. По умолчанию используется

# Свойства объекта Worksheet

- **Свойство Cells** - возвращает объект Range, содержащий все ячейки данного рабочего листа.
- **Свойство Columns** - возвращает объект Range, содержащий все столбцы данного рабочего листа.
- **Свойство Rows** - возвращает объект Range, который представляет собой строку, находящуюся в заданном диапазоне.
- **Свойство Visible** - определяет, является ли объект (в данном случае рабочий лист) видимым или нет. Для значения данного свойства, равного True, объект является видимым, в случае False – невидимым. Для рабочих листов это свойство может также принимать значение xlVeryHidden. В этом случае пользователь не может сделать лист видимым. Чтобы лист снова стал видимым, достаточно задать в программе свойству Visible значение True.

# Свойства объекта Worksheet

**Свойство Range** - возвращает объект Range, который представляет собой ячейку или диапазон ячеек. Для объекта Worksheet данное свойство имеет два варианта синтаксиса.

- Expression.Range(Cell1);
- Expression.Range(Cell1, Cell2).

Expression	Любое выражение, возвращающее объект Range
Cell1	Данный параметр имеет разный смысл для разных вариантов синтаксиса. Для первого варианта – это имя диапазона ячеек, для второго варианта – ячейка в левом верхнем углу диапазона ячеек
Cell2	Определяет ячейку в нижнем правом углу диапазона ячеек

# Обработчики событий

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

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*.

Обработать можно события следующих объектов Excel:

- Application
- Workbook
- Worksheet
- Chart

# Обработчики событий

Обработчики событий дают возможность привязать свой код к действиям пользователя, например к открытию или закрытию книги, активации таблицы, сохранению документа ... Обработчики событий создаются с модулях лисов или книги (в зависимости от того, с каким объектом будет связано это событие).

Например:

```
Private Sub Worksheet_BeforeRightClick(ByVal  
Target As Excel.Range, Cancel As Boolean)
```

```
.....
```

```
End Sub
```

# Обработчики событий

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

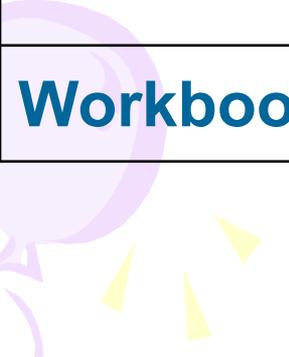
*Public WithEvents App As Word.Application*

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



# События Application

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



# События объекта Workbook

- **Событие Activate** - происходит при активизации рабочей книги, рабочего листа, листа диаграммы или внедренной диаграммы. Для активизации объекта можно использовать метод Activate или интерфейс пользователя.
- **Событие Deactivate** - происходит, когда рабочая книга, рабочий лист, лист диаграммы или внедренная диаграмма становятся неактивными.
- **Событие NewSheet** - возникает в случае создания листа в рабочей книге. Процедура обработки данного события имеет один параметр Sh, который представляет собой созданный лист (объект Worksheet или Chart).

Пример:

```
Private Sub Workbook_NewSheet(ByVal Sh as Object)  
    Sh.Move Before:= Sheets(1)  
End Sub
```

# События объекта Workbook

- **Событие SheetActivate** - возникает в процессе активизации какого-либо листа в рабочей книге. Процедура обработки данного события, так же как и в предыдущем случае, имеет один параметр Sh.
- **Событие SheetDeactivate** - возникает в процессе деактивизации какого-либо листа в рабочей книге. Процедура обработки данного события имеет один параметр Sh.
- **Событие BeforeClose** - возникает перед закрытием рабочей книги. В том случае если в рабочей книге имеются не сохраненные изменения, то это событие происходит перед тем, как пользователю будет выдан запрос на сохранение изменений. Процедура обработки данного события имеет один аргумент Cancel. Этот аргумент позволяет отменить закрытие книги, если в процедуре обработки события ему присвоить значение True.

# События объекта Workbook

• **Событие BeforeSave** - происходит перед тем, как рабочая книга будет сохранена. Процедура обработки данного события имеет два параметра: SaveAsUI и Cancel. Для вывода окна диалога Сохранение документа на экран параметру SaveAsUI необходимо присвоить значение True. Как и в предыдущем случае, аргумент Cancel позволяет отменить сохранение в рабочей книге, если ему задать значение True.

• **Событие Open** - происходит при открытии рабочей книги. Представленный ниже пример процедуры обработки данного события разворачивает окно MS Excel на весь экран:

Пример

```
Private Sub Workbook_Open()  
    Application.WindowState = xlMaximized  
End Sub
```

# События объекта Worksheet

Событие	Когда возникает	Параметры события
SheetActivate (Sh As Object)	Страница становится активной.	Вновь активированная страница передается обработчику события в качестве параметра.
SheetBeforeRightClick (Sh As Object, Target As Range, Cancel As Boolean)	Аналогично двойному щелчку, но при нажатии правой клавиши мыши.	Параметры сохраняют смысл, описанный для обработчика событий двойного щелчка.

# События объекта Worksheet

Событие	Когда возникает	Параметры события
<code>SheetCalculate (Sh As Object)</code>	При перевычислениях рабочей страницы или при любых изменениях данных, отображаемых на диаграмме страницы диаграмм.	В зависимости от того, на странице какого типа произошло событие, параметр <code>Sh</code> представляет либо объект <code>Workbook</code> либо объект <code>Chart</code> .
<code>SheetDeactivate (Sh As Object)</code>	Страница перестает быть активной, поскольку	Деактивированная страница передается обработчику события в качестве параметра.

# События объекта Worksheet

Событие	Когда возникает	Параметры события
SheetBeforeDoubleClick (Sh As Object, Target As Range, Cancel As Boolean)	При двойном щелчке левой клавиши мыши на рабочей странице, но до того, как выполнится макрос, задающий реакцию на щелчок. Событие не возникает на Chart-страницах.	Первый параметр передает обработчику события объект Sh, задающий рабочую страницу, на которой был произведен двойной щелчок. Вторым параметром Target возвращает ячейку (объект Range), ближайшую к указателю мыши в тот момент, когда был произведен щелчок. Если в обработчике события изменить значение параметра Cancel на True, то отменится выполнение макроса, задающего

# События объекта Worksheet

Событие	Когда возникает	Параметры события
SheetChange (Sh As Object, Target As Range)	При изменениях в ячейках рабочей книги, инициированные пользователем или внешней ссылкой. Не возникает для Chart-страниц.	Параметр Sh задает объект Worksheet - страницу, в ячейках которой произошли изменения. Параметр Target задает область изменения - объект Range.
SheetSelectionChange (Sh As Object, Target As Range)	При изменении области выделения рабочей страницы. Не возникает для Chart-страниц.	Параметр Sh задает объект Worksheet - страницу, содержащую новую область выделения. Параметр Target задает область выделения - объект Range.

# Работа с ячейками MS Excel

Работа с ячейками рабочего листа MS Excel выполняется с помощью свойств и методов объекта Range. Этот объект позволяет изменять такие атрибуты ячеек, как вид границ, шрифт, значения и формулы, и выполнять многие другие операции, которые приведены в таблице

# Работа с ячейками MS Excel

Действие	Метод
Задание активной ячейки	Activate
Удаление содержащихся в ячейках значений и формул	ClearContents
Удаление форматирования ячеек	ClearFormats
Копирование содержимого ячеек в буфер обмена	Copy
Удаление ячеек	Delete
Заполнение ячеек содержимым других ячеек	FillDown, FillUp, FillLeft, FillRight
Добавление ячеек	Insert
Объединение ячеек	Merge
Разбиение ячеек	UnMerge
Выделение одной или нескольких ячеек	Select

# Методы объекта Range

- **Метод Activate** - задает активную ячейку в выделенном диапазоне ячеек (для выделения ячейки следует использовать метод Select объекта Range). Например, если объект Range представляет диапазон ячеек A1:C5, то результатом выполнения приведенных ниже инструкций будет активизация ячейки C1:

```
Range("A1:C5").Select `Выделяем ячейки A1:C5
```

```
Range("C1").Activate `Делаем активной ячейку C1
```

- **Метод ClearContents** - очищает формулы и значения, содержащиеся в ячейках, представляемых объектом Range. Данный метод очищает только содержимое ячеек, сохраняя их форматирование.
- **Метод ClearFormats** - удаляет все форматирование ячеек, не воздействуя при этом на хранящиеся в них данные.
- **Метод Copy** - позволяет скопировать содержимое ячеек в буфер обмена.

# Методы объекта Range

• **Метод FillDown** - предназначен для распространения содержимого и форматирования ячейки (или ячеек) из верхней строки диапазона в остальные строки диапазона. Действия, выполняемые данным методом, аналогичны действиям, выполняемым с помощью интерфейса пользователя при распространении содержимого ячейки с использованием мыши.

Пример, который позволяет заполнить все ячейки диапазона C1:C5 содержимым ячейки C1:  
`Worksheets("Лист1").Range("C1:C5").FillDown`

• **Методы FillUp, FillLeft, FillRight** - выполняют действия, аналогичные действиям, выполняемым методом FillDown. Отличие состоит в том, что метод FillUp копирует ячейки, содержащиеся в нижней строке диапазона, метод FillLeft копирует ячейки, содержащиеся в крайнем правом столбце диапазона, а метод FillRight – в крайнем левом столбце.

# Методы объекта Range

**Метод Delete** - удаляет ячейки, определяемые объектом Range. Синтаксис этого метода:

`Expression.Delete(Shift)`

Элемент	Описание
Expression	Выражение, возвращающее объект Range
Shift	Необязательный параметр типа Variant. Этот параметр задает механизм сдвига ячеек в процессе замены удаленных ячеек. Его значение может равняться значению одной из двух констант: <code>xlShiftToLeft</code> или <code>xlShiftUp</code> . Если аргумент Shift отсутствует, то MS Excel определяет способ сдвига ячеек в зависимости от формы диапазона

# Методы объекта Range

**Метод Insert** - позволяет вставить ячейку или диапазон ячеек в рабочий лист. При вставке другие ячейки сдвигаются одним из способов, который определяется аргументом Shift. Синтаксис метода Insert:

Expression.Insert(Shift) )

Элемент	Описание
Expression	Выражение, возвращающее объект Range
Shift	Необязательный параметр типа Variant. Этот параметр задает механизм сдвига ячеек в процессе замены удаленных ячеек. Его значение может равняться значению одной из двух констант: xlShiftToLeft или xlShiftUp. Если аргумент Shift отсутствует, то MS Excel определяет способ сдвига ячеек в зависимости от формы диапазона

# Методы объекта Range

► **Метод Merge** - позволяет объединить несколько ячеек в одну. При этом в качестве объединяемых используются ячейки, определяемые объектом Range.

## Примечание:

Ячейка, получившаяся в результате объединения, содержит значение ячейки, расположенной в левом верхнем углу диапазона.

► **Метод Select** - позволяет выделить ячейки, определяемые объектом Range. Ниже в качестве примера приведена инструкция, реализующая выделение диапазона ячеек C1:F5:

```
Range("C1:F5").Select
```

► **Метод UnMerge** - разбивает объединенную ячейку на отдельные ячейки

# Объектные переменные

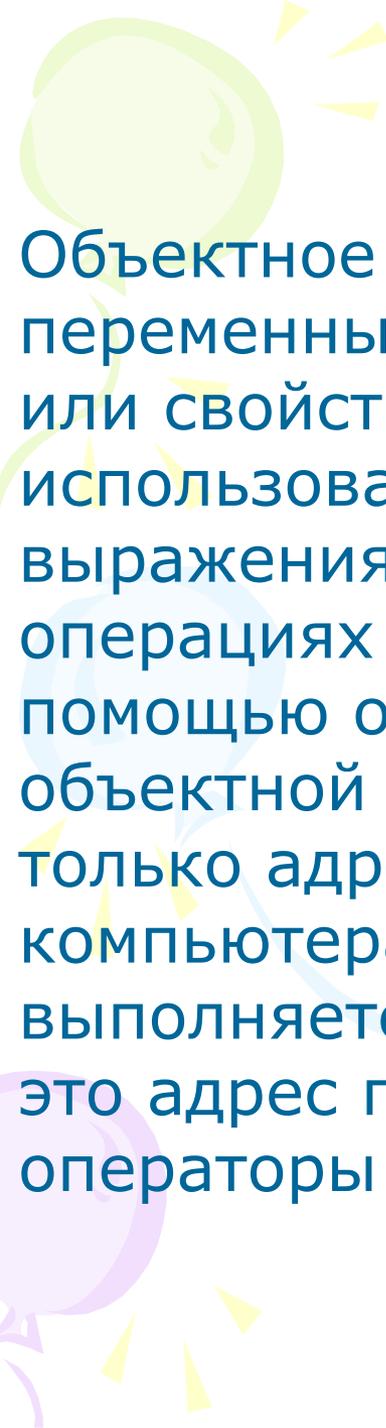
VBA имеет тип Object. Переменные или выражения типа Object ссылаются на объект VBA или на объект, принадлежащий приложению, например Excel-объекты Workbook, Worksheet и Range.

```
Dim myObject As Object
```

```
Dim InstBook As Workbook
```

*Объектное выражение (object expression)* — 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 ваших программах VBA.

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, в действительности, является только адресом, указывающим место в памяти компьютера, где сохранен объект, на который выполняется ссылка. Поскольку объектная ссылка

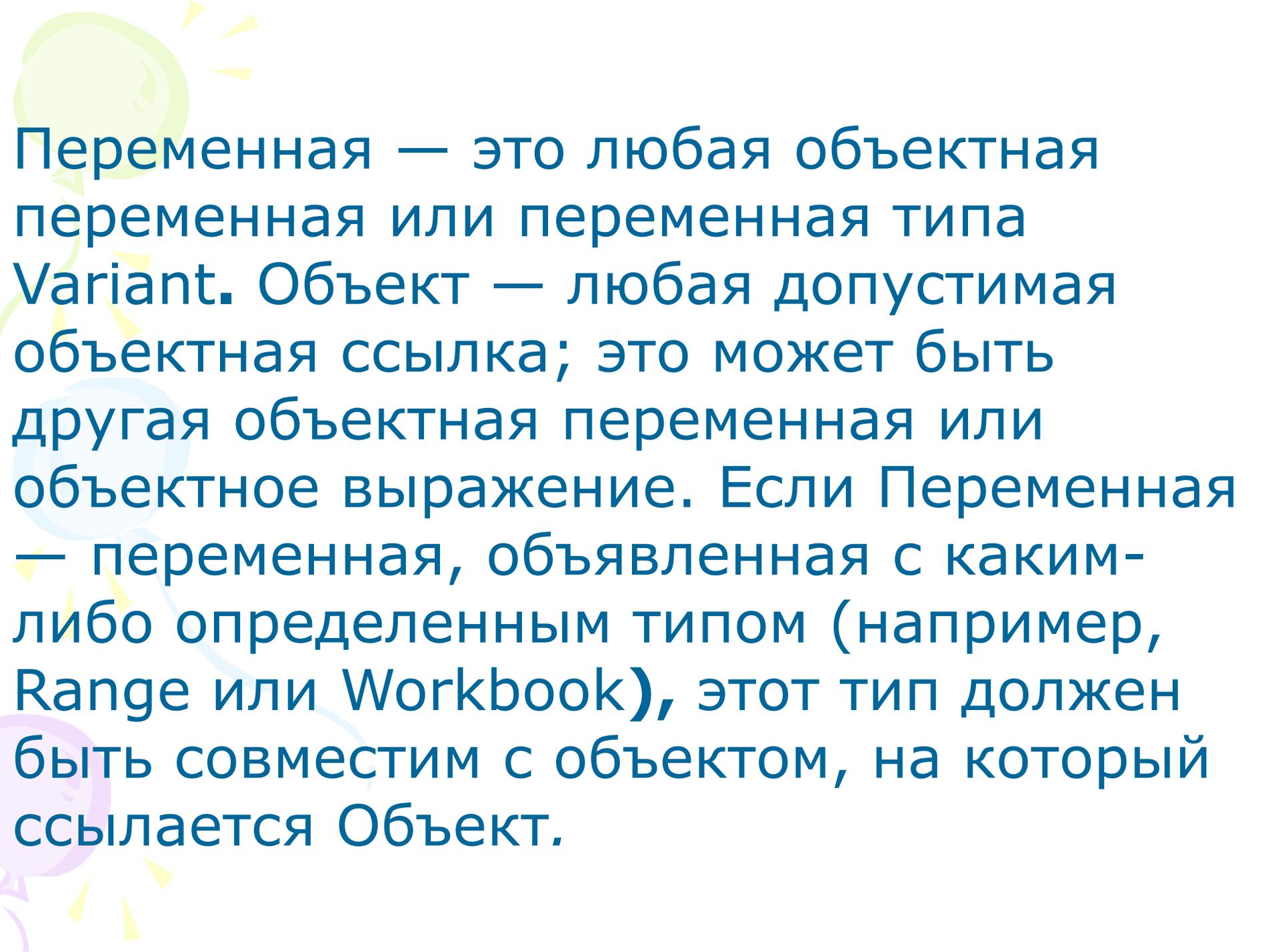


Объектное выражение может состоять из объектных переменных, объектных ссылок или объектного метода или свойства, которое возвращает объект. Нельзя использовать переменные типа Object или объектные выражения в арифметических, логических или операциях сравнения. Объектная ссылка, созданная с помощью объектного выражения или сохраненная в объектной переменной, 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 Переменная=Объект



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.

# Оператор присоединения

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`

`Object` — это любая допустимая объектная ссылка.