



# **Массивы, процедуры, функции UserForm и элементы управления**

# Массивы

Массив (array) – это коллекция переменных, которые имеют общие имя и базовый тип. Массив является удобным способом хранения нескольких связанных элементов данных. Все элементы данных, сохраняемых в массиве, должны иметь один и тот же тип.



# Массивы

Синтаксис создания массива:

**Dim** *ИмяМассива* (размер) **As** *ТипДанных*

**Пример:** Dim МойМассив(3) As Integer —  
массив целых чисел из 4 элементов

Если вместо слова **Dim** набрать слово **Public**, будет создан массив, областью видимости которого станет вся программа.



# Массивы

Элементы созданного массива не содержат никаких данных. Чтобы сохранить в массиве какое-нибудь значение, нужно указать, какому элементу оно должно быть присвоено.

В большинстве программ при создании массива сразу же инициализируют его, присвоив каждому элементу нулевое значение или пустую строку. Это можно сделать, используя цикл **for next**.

**Пример:**

```
Dim сотрудник(4) As String  
Dim I As Integer
```

```
For I = 0 To 4  
    Сотрудник (i) = ""  
Next I
```



# Массивы

Обычно элементы массива содержат значения, относящиеся к одному типу данных, например только строки или только целые числа. Если же необходимо, чтобы в массиве содержались данные разных типов, при создании массива следует указывать тип данных Object.

**Пример.** Элементы следующего массива «содержат» значения разных типов:

```
Dim МойМассив as Object
```

```
МойМассив(0) = «Спрут»
```

```
МойМассив(1) = 56
```

```
МойМассив(2) = 3.1415
```



# ВАЖНО ПОНИМАТЬ!

Тип данных `Object` содержит адреса, ссылающиеся на объекты. Переменной `Object` можно назначить любой ссылочный тип (строка, массив, класс или интерфейс). Переменная `Object` может также ссылаться на данные любого типа значения (числовые, `Boolean`, `Char`, `Date`, структуры или перечисления).

Независимо от того, на какой тип данных она ссылается переменная `Object`, она содержит не значения данных, а — указатель на значение. Она всегда занимает четыре байта памяти, но не включает в себя область памяти для данных, представляющих значение переменной. Поскольку код использует указатель для определения местоположения данных и переменных типа `Object`, доступ к ним несколько медленнее, чем к данным явно вводимых переменных.

[https://msdn.microsoft.com/ru-RU/library/windows/desktop/twcxd6b8\(v=vs.120\)](https://msdn.microsoft.com/ru-RU/library/windows/desktop/twcxd6b8(v=vs.120))



# Многомерные массивы

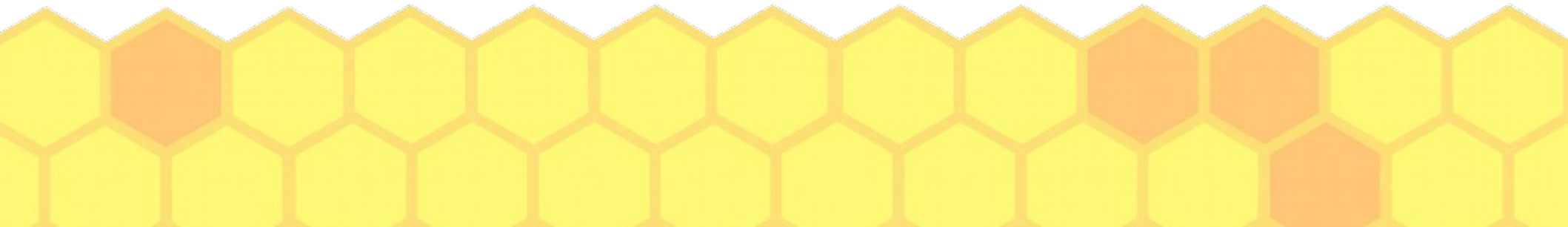
Двухмерный массив создается так же как и одномерный, с той лишь разницей, что, указывая его размер, нужно указать два значения – строки и столбцы:

```
Dim ДвухММассив (Строки, Столбцы) As ТипДанных
```

Размер трехмерного массива будет определяться тремя числами и т. д.:

```
Dim ТрехММассив (X, Y, Z) As ТипДанных
```

При создании массивов, в том числе и многомерных, для хранения значения каждого элемента выделяется оперативная память (даже если это нулевые значения или пустые строки). Таким образом, создавая большой массив, происходит резкое уменьшение объема свободной памяти, что может негативно отразиться на работе программы. Поэтому создавать многомерные массивы следует лишь по мере необходимости.



# Динамические массивы

Выбор размера массива может быть затруднен, если неизвестно, сколько данных будет введено в массив, или если объем данных, собираемых для массива, значительно меняется. Для подобных ситуаций VBA поддерживает особый тип массивов, называемый динамическим (dynamic) массивом. Динамические массивы создаются с помощью оператора Dim, Private, Public или Static, причем список размерностей опускается, затем их размер устанавливается с помощью оператора ReDim во время выполнения процедуры. При использовании ключевого слова Preserve данные, имеющиеся в массиве на момент изменения его размера сохраняются

## Примеры:

Dim Month() As String – объявляет динамический массив Month;  
ReDim Preserve Month( 31) – изменяет размер массива до 31 элемента, сохраняя содержимое;

Dim Table() As Integer – объявляет динамический массив;  
ReDim Table(3, 15) – делает массив двумерным;  
ReDim Preserve Table(4, 25) – только изменяет последний размер массива;

Dim Mas as Variant – объявляет переменную типа Variant;  
ReDim Mas(20) As Integer – создает массив 20 целых чисел в Variant.



# Процедуры VBA

Объект процедура — это блок операторов Visual Basic, заключенным в операторе объявления (Function, Sub и др.) и соответствующим End объявления. Все исполняемые операторы в Visual Basic должны быть в пределах некоторой процедуры.

Процедуры VBA бывают следующих типов:

- **процедуры обработки событий**
- **процедуры общего назначения:**  
**подпрограммы и функции**

Процедуры, как и переменные, должны быть объявлены до того, как они могут быть вызваны.

# Процедуры обработки событий

Имя процедуры обработки события, связанного с элементом управления, состоит из имени элемента управления, символа подчеркивания и имени события. Процедуры обработки событий хранятся в разделах модуля формы или отчета, соответствующих связанным с этими процедурами объектам.

## Пример

Sub **Заккрыть\_** click — процедура обработки нажатия кнопки **Заккрыть** в форме



# Процедуры общего назначения

Общие процедуры VBA могут храниться в любом типе модулей VBA, так как они не связаны с конкретным объектом. Они выполняются только тогда, когда явно вызываются другими процедурами. Обычно эти процедуры реализуют какие-то общие действия, которые могут вызываться разными процедурами обработки событий. Объявления общих процедур помещаются в разделе General (Общая область) модуля.



# Подпрограммы VBA

Объект **Sub** процедура (подпрограмма) — это последовательность операторов Visual Basic, заключенным в Sub и End Sub инструкции. Sub процедура выполняет задачу и возвращает управление вызывающему коду, но не возвращает значение вызывающему коду. Упрощенный синтаксис создания процедуры:

```
Sub <имя_процедуры> (<аргумент>, [...n])  
<оператор>, [...n]
```

[Exit Sub] - Немедленно завершает работу Sub процедуры, в котором он отображается. Выполнение продолжается с оператора, следующего за оператором, вызвавшим Sub процедуры.

End Sub

Optional — указывает на необязательный параметр, для него нужно указывать значение по умолчанию defaultvalue

```
<аргумент>::=[Optional][ByValByRef]имя_параметра As datatype[=defaultvalue]
```

# Подпрограммы VBA

## Пример:

```
Sub tellOperator( Optional ByVal task As String = "MyTask" )
```

```
    Dim stamp As Date
```

```
    stamp = TimeOfDay()
```

```
    MsgBox("Starting " & task & " at " & Cstr(stamp))
```

```
End Sub
```



# Функции VBA

Объект **Function** процедура — это последовательность операторов Visual Basic, заключенным в Function и End Function инструкции. Процедура выполняет задачу и возвращает управление вызывающему коду. При возвращении элемента управления также возвращает значение вызывающему коду.



# Функции VBA

Объект **Function** процедура — это последовательность операторов Visual Basic, заключенным в Function и End Function инструкции. Процедура выполняет задачу и возвращает управление вызывающему коду. При возвращении элемента управления также возвращает значение вызывающему коду. Упрощенный синтаксис объявления функции имеет вид:

Function *Имя\_функции* [*<Аргумент>*, [...n]] As *Тип\_данных*

[Statements]

[Return *Выражение*] — Возвращение значения  
вызывающему коду

End Function

*<Аргумент>*::=[Optional][ByValByRef]*имя\_параметра*Asdatatype[=defaultvalue]

# Возвращение значений функциями VBA

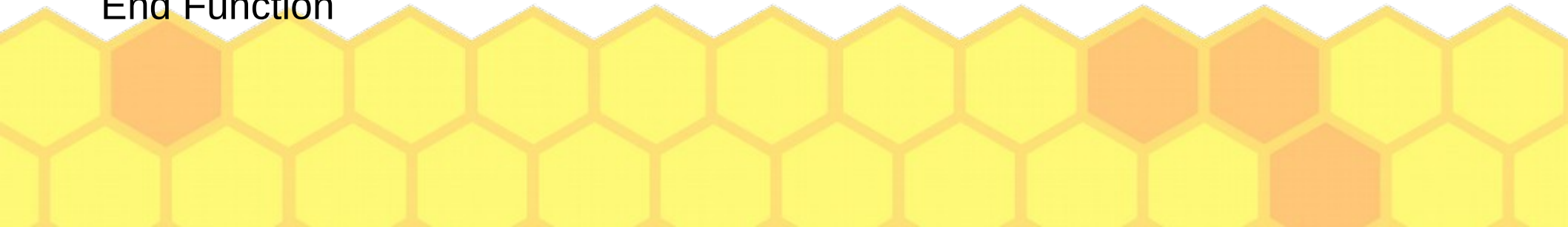
Функция может вернуть значение одним из двух способов:

- Используя синтаксическую конструкцию *Return Выражение*

```
Function Имя_функции [(ParameterList)] As Тип_данных  
    Expression = ...  
    Return Expression  
End Function
```

- Значение присваивается собственному имени функции в одной или нескольких инструкциях процедуры

```
Function Имя_функции [(ParameterList)] As Тип_данных  
    Expression = ...  
    Имя_функции = Expression  
End Function
```





# Функции VBA

## Пример

```
Function hypotenuse(ByVal side1 As Single, ByVal side2 As Single) As Single  
    Return Math.Sqrt((side1 ^ 2) + (side2 ^ 2))  
End Function
```

## Пример

```
Function hypotenuse(ByVal side1 As Single, ByVal side2 As Single) As Single  
    hypotenuse = Math.Sqrt((side1 ^ 2) + (side2 ^ 2))  
End Function
```



# UserForm

Диалоговые окна в VBA называются формами (объект UserForms). Каждому объекту **UserForm** присущи определенные свойства, методы и события, которые он наследует от класса объектов **UserForms**. Диалоговые окна (формы) и элементы управления составляют основу современного визуального интерфейса. Все элементы управления и технология работы с ними в основном стандартизованы и похожи для разных платформ и программных сред. Эти объекты помещены в специальную библиотеку MSForms.



# UserForm

При создании визуального интерфейса следует помнить:

- Все загруженные диалоговые окна представляют коллекцию UserForms со стандартными методами и свойствами. Элемент коллекции – объект класса UserForm – задает отдельное окно.
- Для каждого типа элементов управления в библиотеке msforms имеется класс объектов, имя которого совпадает с именем элемента управления (его типа).
- Диалоговые окна создаются, как правило, визуально. Вначале создается само окно, а затем оно наполняется элементами управления при помощи соответствующей панели элементов.
- Всякий раз, создавая новый объект UserForm, одновременно создается новый подкласс объекта UserForm. Любые процедуры или функции, написанные в разделе General (общий) модуля класса, относящегося к форме, становятся дополнительными методами для отдельного подкласса объекта
- Событийные процедуры следует записывать в модуль класса, который является частью User Form. При этом такие процедуры должны иметь имена в виде ObjectName\_EventName, где ObjectName – имя формы или элемента управления, а EventName – имя события, с которым идет работа. Такой формат имени позволяет VBA сопоставлять заданному событию требуемую процедуру.

# Элементы управления VBA















Элемент управления	Назначение
CheckBox  (флажок)	Стандартный флажок (квадратное окно, содержащее, если элемент выбран, галочку). Используйте флажки для выбора вариантов, которые не являются взаимо-исключающими
OptionButton  (переключатель)	Стандартная кнопка-переключатель (круглое окно, при выборе в центре него находится черная точка). Используйте OptionButton, когда пользователю необходимо сделать выбор между положениями «включено/выключено» или «истина/ложь». Кнопки-переключатели, как правило, объединяются вместе при помощи рамки для создания группы переключателей
ToggleButton  (выключатель)	Выключатели служат для той же цели, что и флажки, но выводят установки в виде кнопки, находящейся в «нажатом» или «отжатом» состоянии
Frame  (рамка)	Визуально и логически объединяет некоторые элементы управления (особенно флажки, переключатели и выключатели)
CommandButton  (кнопка)	Используйте кнопки для выполнения таких действий, как Cancel (Отмена), Save (Сохранить), OK и т.д. Когда пользователь щелкает по кнопке, выполняется VBA-процедура, закрепленная за данным элементом управления
TabStrip  (набор вкладок)	Этот элемент управления состоит из области, в которую вы помещаете другие элементы управления (такие как текстовые поля, флажки и т.д.) и полосы кнопок табуляции. Используйте элемент управления TabStrip для создания диалоговых вкладок, отображающих одни и те же данные в различных категориях
MultiPage  (набор страниц)	Этот элемент управления состоит из нескольких страниц. Вы можете выбрать любую из них, щелкнув по соответствующей вкладке. Используйте элемент управления MultiPage для создания диалоговых окон с вкладками
ScrollBar  (полоса прокрутки) и SpinButton  (счетчик)	Элемент управления ScrollBar позволяет выбирать линейное значение аналогично тому, как это можно сделать при помощи счетчика. Элемент управления SpinButton является специальной разновидностью текстового поля



Рис.20 Панель элементов

Элемент управления	Назначение
Label  (надпись, метка)	Позволяет создавать заголовки элементов управления, которые не имеют собственных встроенных заголовков
TextBox  (текстовое поле)	Окно редактируемого текста свободной формы для ввода данных. Может быть одно- или многострочным
ComboBox  (поле со списком)	Этот элемент управления объединяет окно редактирования и окно списка. Используйте, когда хотите предложить пользователю выбрать значение, но при этом дать ему возможность ввести данные, отсутствующие в списке
ListBox  (список)	Отображает список значений, из которых пользователь может сделать выбор. Окна списка можно использовать, чтобы дать возможность пользователю выбрать только одно значение или же несколько

Элемент управления	Назначение
Image  (рисунок)	Элемент управления Image позволяет вывести на форме графическое изображение. Используйте Image для вывода графических изображений в любом из следующих форматов: *.bmp, *.cur, *.gif, *.ico, *.jpg или *.wmf



# Свойства элементов управления VBA

Свойство	Где применяется	Описание
Accelerator	CheckBox, Tab, CommandButton, Label, Page, OptionButton, ToggleButton	Содержит символ, используемый в качестве быстрой клавиши вызова, элемента управления, при нажатии Alt+<клавиша быстрого вызова> происходит выбор элемента управления
BackColor	Все элементы	Число, представляющее определенный цвет фона элемента управления
Caption	CheckBox, CommandButton, Frame, Label, OptionButton, ToggleButton, Page, Tab, UserForm	Для надписи — текст, отображаемый элементом управления. Для других элементов управления — надпись, которая появляется на кнопке или вкладке или рядом с рамкой, флажком или переключателем
Cancel	CommandButton	Задаёт кнопку отмены диалогового окна. При нажатии на эту кнопку или клавишу Esc диалоговое окно исчезает. Только одна кнопка формы может иметь данное свойство
ControlTip-Text	Все элементы управления	Устанавливает текст, который отображается в виде всплывающей подсказки (ControlTip, называемой также ToolTip), когда указатель мыши помещается на элемент управления

# Свойства элементов управления

Свойство	Где применяется	Описание
Default	CommandButton	Определяет заданную по умолчанию кнопку. Когда пользователь нажимает в процессе диалога клавишу Enter, эта кнопка ведет себя так, как если бы по ней щелкнули мышью
Enabled	Все элементы управления	Хранит значение типа Boolean, определяющее, доступен или нет элемент управления. Если Enabled имеет значение False, то элемент управления продолжает отображаться в диалоговом окне, но не может быть выбран
ForeColor	Все элементы управления	То же самое, что и BackColor, но устанавливает цвет для переднего плана элемента управления, как правило, символов текста
List	ComboBox	Массив типа variant (одно- или многомерный), представляет список, содержащийся в элементе управления
Max	ScrollBar, SpinButton	Переменная типа Long, определяющая максимальное значение счетчика, или значение, при котором полоса прокрутки находится в самом верху (для вертикальной полосы) или справа (для горизонтальной)
Min	ScrollBar, SpinButton	Переменная типа Long, определяющая минимальное значение счетчика, или значение, при котором полоса прокрутки находится в самом низу (для вертикальной полосы) или слева (для горизонтальной)
Name	Все элементы управления	Содержит имя элемента управления. Вы можете установить данное свойство только с помощью Properties Window
RowSource	ComboBox	Задаёт источник, из которого ListBox берет список объекта. В Excel VBA RowSource обычно использует диапазон рабочего листа
Selected	ListBox	Возвращает массив значений типа Boolean для списка, который допускает множественный выбор. Каждый элемент массива содержит по одному элементу, соответствующему каждому пункту списка. Если значение элемента в массиве selected равно True, то соответствующий пункт списка выбран

# Свойства элементов управления VBA

Свойство	Где применяется	Описание
TabIndex	Все элементы управления	Число, указывающее положение элемента управления в порядке табуляции (может иметь значение от 0 до значения, равного количеству элементов управления на форме)
TabStop	Все элементы управления	Значение типа Boolean, указывающее, может ли элемент управления быть выбран клавишей Tab. Если значение TabStop равно False, вы тем не менее можете щелкнуть на элементе и таким образом его выбрать
Value	Все элементы управления	Значение текущих установок элемента управления: текст в текстовом поле, какие выбраны флажки и переключатели, индекс выбранного раздела списка или число, указывающее текущее положение полосы прокрутки или счетчика
Visible	Все элементы управления	Значение типа Boolean, указывающее, является ли элемент управления видимым

