

СИНТАКСИС ЯЗЫКА VBA

**МАССИВЫ VBA. СТАТИЧЕСКИЕ И ДИНАМИЧЕСКИЕ
МАССИВЫ.**

к.п.н., доцент Суханова Н.Т.

ПОНЯТИЕ И РАЗМЕРНОСТЬ МАССИВА

Массив - это совокупность переменных, которые имеют общие имя и базовый тип.

Все элементы данных, сохраняемых в массиве, должны иметь один и тот же тип.

Информация, сохраненная в массиве, может быть доступна в любом порядке.

Массив позволяет сохранять и манипулировать многими элементами данных посредством единственной переменной.

Обработку массивов значительно упрощает использование циклов.

ПОНЯТИЕ И РАЗМЕРНОСТЬ МАССИВА

Для хранения значений массива отводится столько ячеек памяти, сколько их содержится в массиве.

Каждая ячейка памяти отводится под отдельный элемент массива. Таким образом, **массив является структурированной переменной** в отличие от скалярной простой переменной. Все элементы массива имеют один и тот же тип.

ПОНЯТИЕ И РАЗМЕРНОСТЬ МАССИВА

При необходимости выполнить некоторое действие над всеми элементами массива можно сослаться на массив как целое по его имени.

Возможны также ссылки на его отдельные элементы. *Каждый отдельный элемент массива* определяется именем массива и значениями индексов.

Например, ссылка **A(7)** означает, что A – это имя одномерного массива с одним индексом, элемент массива имеет значение индекса равное 7.

Одномерные массивы

Одномерный массив - это самый простой вариант массива, использующий обыкновенный список данных.

Например:

Вася, Петя, Коля, Миша, Ваня, Слава, Игорь, Юра

Это строковый массив, состоящий из 8 элементов. Дадим ему название *My_Array*.

My_Array = {Вася, Петя, Коля, Миша, Ваня, Слава, Игорь, Юра}

My_Array(3) = Коля

НУМЕРАЦИЯ ЭЛЕМЕНТОВ МАССИВА

Нумерация элементов в массиве начинается с 0. Такая система нумерации довольно распространена в программировании и называется нумерацией с *нулевой базой*.

- ✓ Поскольку нумерация с нулевой базой не очень удобна, то в VBA имеется директива компилятора, позволяющая исправить это "неудобство": ***Option Base***. Директива компилятора имеет два варианта написания:
Option Base 0 - индексы массивов начинаются с 0 (установка по умолчанию)
Option Base 1 - индексы массивов начинаются с 1

НУМЕРАЦИЯ ЭЛЕМЕНТОВ МАССИВА

Данная директива компилятора помещается в область объявлений модуля перед объявлениями любых переменных, констант или процедур. Нельзя помещать *Option Base* внутри процедуры. Можно иметь только один оператор *Option Base* в модуле, который влияет на все массивы, объявляемые в модуле. Одномерные массивы хорошо подходят для представления простых списков данных. Однако часто бывает необходимо представить таблицы данных в программах с организацией данных в формате строк и столбцов, подобно ячейкам в рабочих листах Excel.

МНОГОМЕРНЫЕ МАССИВЫ

Для этого необходимо использовать *многомерные массивы*. Так адрес каждой ячейки листа состоит из двух чисел, одно из которых (**номер строки**) является первым индексом, а второе (**номер столбца**) - вторым индексом массива.

Такой массив называется **двумерным массивом**. Добавив еще номер листа, получим **трехмерный массив**. VBA позволяет создавать массивы, имеющие до 60 измерений.

СТАТИЧЕСКИЕ И ДИНАМИЧЕСКИЕ МАССИВЫ

Массивы, не меняющие число своих элементов, называются **статическими массивами**.

Примером такого массива может служить вышеприведенный массив *My_Array*, содержащий 8 элементов.

Однако бывают ситуации, когда изначально неизвестно количество элементов в массиве, или же, в процессе работы это количество может изменяться.

Такие массивы называются **динамическими массивами**. Динамический массив может увеличиваться или сжиматься, чтобы вмещать точно необходимое число элементов без напрасного расходования памяти.

ОБЪЯВЛЕНИЕ МАССИВОВ

Объявление массива с использованием оператора ***Dim*** имеет следующий синтаксис:

Dim VarName([Subscripts]) [As Type]

VarName - любое имя массива, использующее допустимый идентификатор имени;

Subscripts - измерение массива. Если размерность массива больше единицы, то *Subscripts* разделяются запятыми.

ОБЪЯВЛЕНИЕ МАССИВОВ

Оператор ***Subscripts*** имеет следующий синтаксис:

[lower To] upper [, [lower To] upper]..

lower - определяет нижний диапазон допустимых индексов для массива (необязательный аргумент);

upper - определяет верхний предел для индексов массива (обязательный аргумент).

ОБЪЯВЛЕНИЕ МАССИВОВ

Примеры правильного объявления массивов:

Dim A(10) As Integer - одномерный статический строковый массив, включающий 10 элементов;

Dim Array_Str (1 To 10) As String - одномерный статический строковый массив, включающий 10 элементов;

Dim Array_Var() - динамический массив;

Dim Array_Mult (0 To 5, 0 To 7) As Integer - двумерный статический массив целых чисел, включающий $6 \times 8 = 48$ элементов.

ОБЪЯВЛЕНИЕ МАССИВОВ

При объявлении массивов следует помнить, что включение оператора ***Subscripts*** в объявлении массива создает **статический массив** с фиксированным числом элементов, пропуск оператора ***Subscripts*** в объявлении массива создает **динамический массив**, а установка директивы компилятора ***Option Base*** влияет на общее число элементов в массиве.

ВВОД ДАННЫХ В МАССИВ

Ввод данных в массив можно осуществлять с помощью:

- ✓ Оператора ввода;
- ✓ Оператора DIM (инициализация в момент создания);
- ✓ Оператора присваивания.

Например:

DIM b() As Double={0.9, 1.7, 2.3, 4.5}

Количество элементов не указано явно, а определяется по количеству чисел, заданных при инициализации.

ВВОД ДАННЫХ В МАССИВ

Например:

DIM b() As Double

B = New Double() {0.9, 1.7, 2.3, 4.5}

Массив создается в одной строке, а инициализируется в другой.

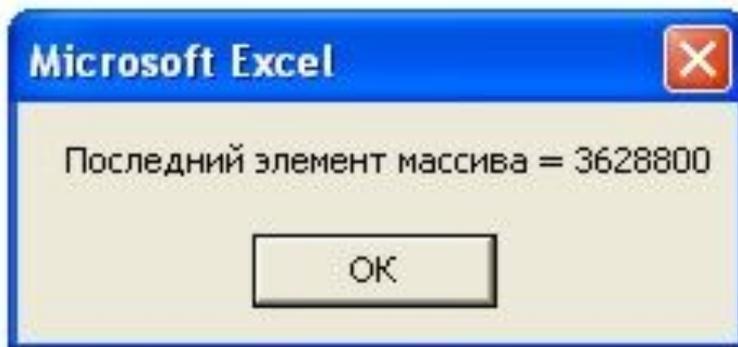
ИСПОЛЬЗОВАНИЕ МАССИВОВ

Для доступа к элементу массива необходимо указывать имя массива, за которым следует значение индекса, заключенное в круглые скобки.

Рассмотрим листинг программы, показывающий элементарное объявление и использование массива (в качестве элементов массива используется факториал номера элемента массива):

ИСПОЛЬЗОВАНИЕ МАССИВОВ

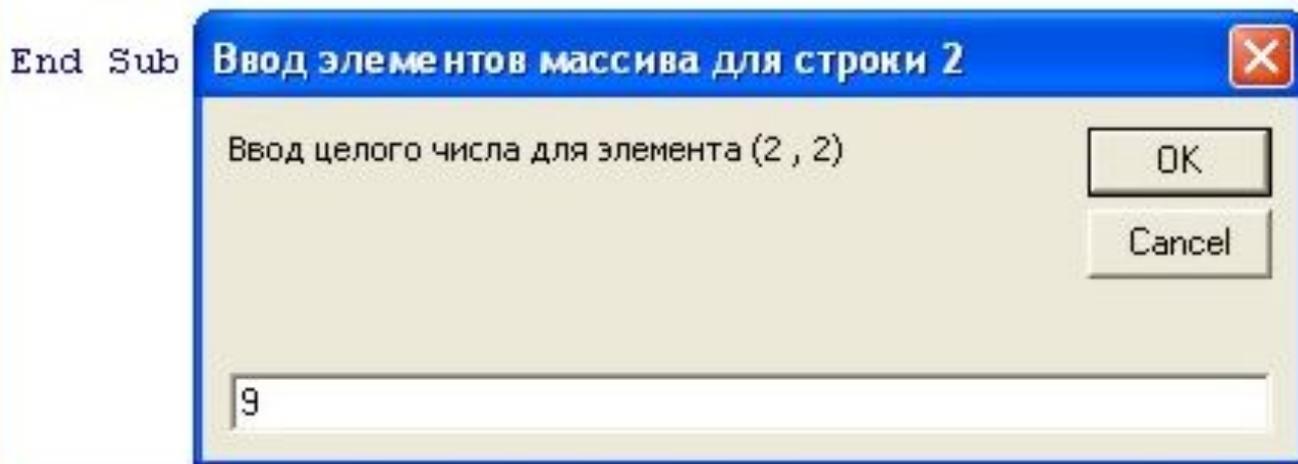
```
Sub example_02()  
    Dim Fctrl(1 To 10) As Single  
    Dim i As Integer  
  
    Fctrl(1) = 1  
  
    For i = 2 To 10  
  
        Fctrl(i) = Fctrl(i - 1) * i  
  
    Next i  
  
    MsgBox ("Последний элемент массива = " & Fctrl(10))  
  
End Sub
```



ИСПОЛЬЗОВАНИЕ МАССИВОВ

При помощи вложенных циклов довольно легко можно организовать инициализацию двумерного массива:

```
Sub example_02 ()  
    Dim Dbl_Array(5, 7) As Single  
    Dim i As Integer, j As Integer  
  
    For i = 1 To 5  
        For j = 1 To 7  
            Dbl_Array(i, j) = InputBox("Ввод целого числа для" & _  
                " элемента (" & i & " , " & j & ")", _  
                "Ввод элементов массива для строки " & i)  
        Next j  
    Next i  
  
End Sub
```



ИЗМЕНЕНИЕ РАЗМЕРНОСТИ ДИНАМИЧЕСКОГО МАССИВА

Если точно неизвестно, сколько элементов потребуется в массиве – используется оператор **ReDim**, который переопределяет размерность массива, а во время объявления не указывать его размерность. Синтаксис *ReDim*:

**ReDim [Preserve] varname(subscripts)
[As Type] [, varname(subscripts) [As Type]]**

varname - имя существующего массива;
subscripts - размерность существующего массива;

Type - любой тип VBA.

ИЗМЕНЕНИЕ РАЗМЕРНОСТИ ДИНАМИЧЕСКОГО МАССИВА

Необходимо использовать отдельный оператор **As Type** для каждого массива, который определяется;

Preserve - необязательный аргумент.

Его использование приводит к тому, что данные уже имеющиеся в массиве, сохраняются после изменения его размерности.

ПРИМЕРЫ ПРАВИЛЬНОГО ИСПОЛЬЗОВАНИЯ ОПЕРАТОРА **REDIM**:

Dim Array_Month() As String - одномерный строковый динамический массив

ReDim Array_Month(29) - устанавливает размерность динамического массива равную 29 элементам

ReDim Array_Month(1 To 30) - изменяет размер массива до 30 элемента

ReDim Preserve Array_Month(1 To 31) - изменяет размер массива до 31 элемента, сохраняя содержимое

Dim Array_DBL() As Single - объявляет динамический массив

ReDim Array_DBL(2, 9) - делает массив двумерным

ReDim Array_DBL(3, 7) - изменяет размер двумерного массива

ReDim Preserve Array_DBL(1 To 3, 1 To 5) - изменяет последний размер массива, сохраняя содержимое

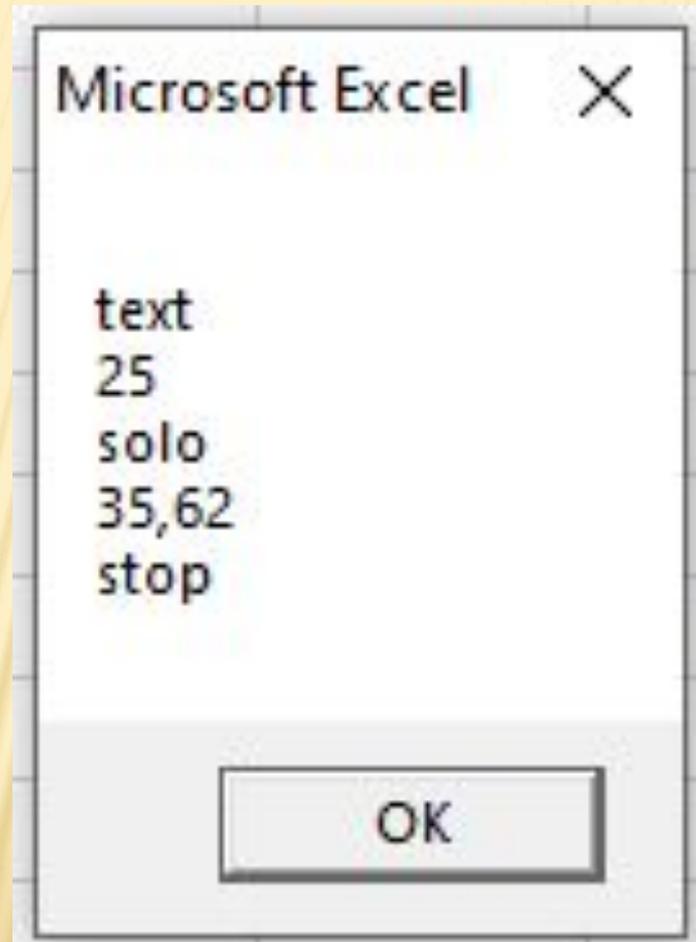
ФУНКЦИИ ARRAY

Функция **Array** возвращает массив элементов типа **Variant** из первоначального списка элементов, перечисленных через запятую. Нумерация элементов в массиве начинается с нуля. Обратиться к элементу массива можно, указав в скобках его номер (индекс).

ПРИМЕР ФУНКЦИИ ARRAY

```
Sub Test1 ()  
Dim a () As Variant  
a = Array ("text", 25, "solo", 35.62,  
"stop")  
MsgBox a(0) & vbCrLf & a(1) &  
vbCrLf & a(2) & vbCrLf & a(3) &  
vbCrLf & a(4)  
End Sub
```

ПРИМЕР ФУНКЦИИ ARRAY



Функции **LBound**, **UBound**

Функции ***LBound***, ***UBound*** возвращают нижнее и верхнее граничные значения индексов статического или динамического массива.

Синтаксис:

LBound (*array_Name* [, *dimension*])

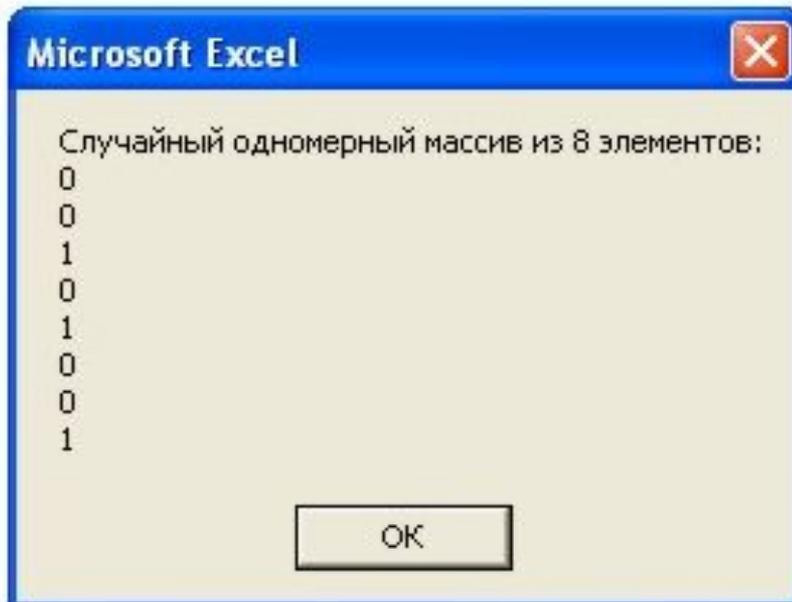
UBound (*array_Name* [, *dimension*])

array_Name - имя массива

dimension - целое число (необязательный аргумент). Определяет измерение массива, для которого надо получить верхний или нижний предел. При отсутствии *dimension* возвращается предел для первого измерения массива.

Функции LBound, UBound

```
Sub example_02()  
    Dim i As Integer, j As Integer, dim_val As String  
    Dim Sngl_Array(3 To 10) As Integer  
  
    dim_val = ""  
  
    For i = LBound(Sngl_Array) To UBound(Sngl_Array)  
        Sngl_Array(i) = Rnd  
        dim_val = dim_val + CStr(Sngl_Array(i)) + Chr(13)  
    Next i  
  
    MsgBox "Случайный одномерный массив из 8 элементов: " & Chr(13) & dim_val  
  
End Sub
```



ОЧИСТКА И УДАЛЕНИЕ МАССИВОВ ПРИ ПОМОЩИ ERASE

Оператор *Erase* позволяет выполнять очистку для статических массивов и удаление - для динамических.

Когда элементы массива заполнены, данные в массиве остаются до тех пор, пока пользователь не присвоит новые значения элементам массива или пока VBA не освободится от массива.

ОЧИСТКА И УДАЛЕНИЕ МАССИВОВ ПРИ ПОМОЩИ ERASE

Зачастую бывает, что в дальнейших вычислениях динамический массив ни при каких обстоятельствах использоваться не будет, поэтому нецелесообразно «держат» его в памяти компьютера, т.к. это может сказаться на скорости работы программы. Или же может понадобиться очистить все значения в статическом массиве, устанавливая числовые значения на 0, а строковые - на пустые строки. Это можно осуществить при помощи вложенных циклов.

ОПЕРАТОР ERASE

Оператор **Erase** обнуляет (если массив статический) или очищает (если массив динамический) массив.

Оператор **Erase** удаляет из памяти динамические массивы, освобождая область памяти, ранее используемую этим массивом. При удалении динамического массива с помощью оператора **Erase** необходимо повторно создать массив с помощью оператора **ReDim** перед тем, как можно будет использовать этот определенный динамический массив снова.

ТЕРМИН «ОБНУЛЕНИЕ» МОЖНО ПРИМЕНИТЬ ТОЛЬКО К МАССИВУ ЧИСЛОВОГО ТИПА.

```
Dim Massiv1 (4, 3) As String, Massiv2 ()  
As Variant
```

' переопределяем динамический массив

```
ReDim Massiv2 (2, 5, 3)
```

' очищаем массивы

```
Erase Massiv1
```

```
Erase Massiv2
```

Обратите внимание, что оба массива при таком способе очистки будут возвращены в исходное состояние, которое они имели сразу после объявления:

статический Massiv1 сохранит размерность (4, 3);

динамический Massiv2 не сохранит размерность ().

ВТОРОЙ СПОСОБ ОЧИСТКИ (ОБНУЛЕНИЯ) МАССИВОВ

Динамический массив можно очистить (обнулить) без использования цикла с помощью оператора **ReDim**. Просто переопределите его с той же размерностью.

Пример:

```
Dim Massiv() As Double
```

```
'переопределяем массив
```

```
ReDim Massiv(5, 6, 8)
```

```
'очищаем массив
```

```
ReDim Massiv(5, 6, 8)
```

ПОВЕДЕНИЕ ОПЕРАТОРА *ERASE* ДЛЯ СТАТИЧЕСКИХ МАССИВОВ ЗАВИСИТ ОТ КОНКРЕТНОГО ТИПА ЭЛЕМЕНТОВ МАССИВА (СМ. ТАБЛИЦУ):

Тип статического массива	Действие оператора <i>Erase</i>
Любой числовой тип	Устанавливает элементы массива на 0
Любой строковый тип	Устанавливает элементы массива на строку нулевой длины, а для строк фиксированной длины - как все символы пробела
Тип <i>Variant</i>	Устанавливает элементы массива на <i>Empty</i>
Тип <i>Object</i>	Устанавливает элементы массива на <i>Nothing</i>
Любой пользовательский тип	Устанавливает каждую переменную в пользовательском типе индивидуально: численные - на 0; строковые - на строки нулевой длины; <i>Variant</i> - <i>Empty</i> ; <i>Object</i> - <i>Nothing</i>