



# Операторы и выражения в VBA



# ВЫРАЖЕНИЯ

Выражение – совокупность переменных, констант, знаков операций, имен функций, скобок, которая может быть вычислена в соответствии с синтаксисом языка программирования. Результатом вычисления выражения является величина определенного типа.



# ВЫРАЖЕНИЯ

Используются следующие типы выражений:

- выражение типа даты – вычисляется до значения типа `date`;
- численное выражение – вычисляется до любого числа;
- строковое выражение – имеет результатом значение типа `string`;
- логическое выражение – вычисляется до значения типа `boolean`.



# ВАЖНО !

Очень важно контролировать и знать тип выражения, потому что если выражения содержат несовместимые типы, VBA выдает ошибку времени исполнения – ошибку несовпадения типов (type-mismatch).



# ВАЖНО !

При обработке выражения, содержащего различные типы данных, VBA сначала пытается устранить любое различие типов, преобразуя значения в выражении в совместимые типы данных. Если устранить какие-либо различия преобразованием типов не удастся, отображается ошибка времени исполнения и процедура прекращает выполняться.



# ОПЕРАТОРЫ

Операторы используются для объединения, сравнения или других действий над определенными значениями в выражении. При использовании оператора в выражении элементы данных, над которыми этот оператор выполняет действие, называются операндами: большинству операторов требуются два операнда.



# ОПЕРАТОРЫ

Операторы выполняют операции:

- арифметические;
- логические;
- сравнения (отношения);
- конкатенации (сцепления строк).





# АРИФМЕТИЧЕСКИЕ ОПЕРАТОРЫ

Математическая запись	Запись на Бэйсике	Примечание
$A + B$	$A + B$	Сложение числовых данных, сцепление строковых, логическое «ИЛИ»
$A - B$	$A - B$	Вычитание числовых данных,
$AB$	$A * B$	Умножение числовых данных, логическое «И»
$\frac{A}{B}$	$A / B$	Деление числовых данных
$A^B$	$A ^ B$	Возведение в степень
Целочисленное деление	$A \backslash B$	Частное от деления, дробная часть <b>отбрасывается.(не округляется!)</b> $5 \backslash 2 = 2$
Деление по модулю	$A \bmod B$	Остаток от деления $5 \bmod 2 = 1$



# ОПЕРАТОРЫ СРАВНЕНИЯ

Математическая запись	На VBA	Результат операции
$A = B$	$A = B$	True или False
$A < B$	$A < B$	
$A \leq B$	$A \leq B$	
$A > B$	$A > B$	
$A \geq B$	$A \geq B$	
$A \neq B$	$A < > B$	

# ЛОГИЧЕСКИЕ ОПЕРАТОРЫ

Оператор	Синтаксис	Описание
AND	A AND B	<i>Конъюнкция:</i> Если A и B имеют значение True, то - True. Иначе - False
OR	A OR B	<i>Дизъюнкция:</i> Если любой из операндов имеет значение True, то - True. Иначе - False
NOT	NOT A	<i>Отрицание:</i> Если A имеет значение False, то - True. Иначе - False
XOR	A XOR B	<i>Исключение:</i> Если A имеет значение True или B имеет значение True, то - True. Иначе - False
EQV	A EQV B	<i>Эквивалентность:</i> Если A имеет такое же значение что и B, то - True. Иначе - False
IMP	A IMP B	<i>Импликация:</i> Если A имеет значение True и B имеет значение False, то - False. Иначе - True

# AND

A	B	A&B
0	0	0
0	1	0
1	0	0
1	1	1



# OR

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

# XOR

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

# NOT

A	$\overline{A}$
0	1
1	0

# EQV

A	B	$A \leftrightarrow B$
0	0	1
0	1	0
1	0	0
1	1	1





# IMP

A	B	$A \rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1



# ОПЕРАТОРЫ КОНКАТЕНАЦИИ

Конкатенация - присоединение одной строки к другой .

В VBA имеется два оператора для конкатенации строк: & и +



# ОПЕРАТОР КОНКАТЕНАЦИИ &

Синтаксис : Операнд\_1 & Операнд\_2 [& Операнд\_3..]

Операнд\_N - любое допустимое строковое или численное выражение (которое преобразуется в строковое).

Тип данных результата конкатенации строк - String.

Если операнд в выражении конкатенации строк имеет значение Empty или Null, VBA интерпретирует этот операнд как строку нулевой длины (строка не содержащая символов).



# ОПЕРАТОР КОНКАТЕНАЦИИ

+

Этот оператор имеет такой же синтаксис и требования, как и оператор (&). Однако следует понимать, что в VBA основное предназначение оператора (+) - это арифметическое сложение. Поэтому, чтобы избежать двусмысленности чтения программного кода, для конкатенации строк настоятельно рекомендуется использовать именно оператор (&).



# ОПЕРАТОР LIKE

Оператор Like дает возможность выполнять особый тип операции сравнения строк, и его можно использовать только со строками.

Оператор Like тестирует строку для определения того, совпадает ли она с заданным шаблоном .



# ОПЕРАТОР LIKE

Оператор Like дает возможность выполнять особый тип операции сравнения строк, и его можно использовать только со строками.

Оператор Like тестирует строку для определения того, совпадает ли она с заданным шаблоном .



# ОПЕРАТОР LIKE

Синтаксис:

результат = строка Like образец

Если строка подходит под маску, то  
результат True.

Если нет - False.

Если одно из выражений Null  
- результат тоже Null.





# ОПЕРАТОР LIKE

Самое главная функция оператора Like - это проверка на принадлежность строки какой-нибудь маске. В маске можно использовать следующие спец. символы:

? Любой отдельный символ

\* Ноли или более символов

# Любая цифра (0–9).

[charlist] Любой отдельный символ, попадающий в список charlist

[!charlist] Любой отдельный символ, не попадающий в список charlist



# ОПЕРАТОР LIKE

Dim MyCheck

MyCheck = "aBBBa" Like "a\*a" ' Возвратит True.

MyCheck = "F" Like "[A-Z]" ' Возвратит True.

MyCheck = "F" Like "[!A-Z]" ' Возвратит False.

MyCheck = "a2a" Like "a#a" ' Возвратит True.

MyCheck = "aM5b" Like "a[L-P]#[!c-e]" ' Возвратит True.

MyCheck = "BAT123khg" Like "B?T\*" ' Возвратит True.

MyCheck = "CAT123khg" Like "B?T\*" ' Возвратит False.

myString = "312T-87GD-8922"

If myString Like "####[A-Z]-##[A-Z][A-Z]-#####" Then ..???



# ОПЕРАТОРЫ УСЛОВНОГО ПЕРЕХОДА

Оператор условного перехода— это структура, которая выбирает ту или иную ветвь кода процедуры на основе некоторого предопределенного условия или группы условий.

Простейшими VBA-операторами изменения порядка выполнения кода являются операторы If ... Then и If ... Then ... Else.



# ОПЕРАТОРЫ УСЛОВНОГО ПЕРЕХОДА

If условие Then

оператор 1

оператор 2

....

оператор n

End If

Dim num1 As Integer = 10

Dim num2 As Integer = 9

If (num1 > num2) Then

....

End If



# ОПЕРАТОРЫ УСЛОВНОГО ПЕРЕХОДА

If условие Then

оператор

....

Else

оператор

....

End If

Dim num1 As Integer = 10

Dim num2 As Integer = 9

If (num1 > num2) Then

.....

Else

.....

End If



# ОПЕРАТОРЫ УСЛОВНОГО ПЕРЕХОДА

If условие Then

оператор

....

Elseif

оператор

....

Else

.....

EndIf

Dim num1 As Integer = 10

Dim num2 As Integer = 9

If (num1 > num2) Then

....

Elseif num1 < num2 Then

...

Else

...

End If



# ОПЕРАТОРЫ УСЛОВНОГО ПЕРЕХОДА

```
Dim R As New Random()  
Dim num1 As Integer = R.Next(100)  
If num1 = 1 Then  
...  
Elseif num1 = 2 Then  
...  
Elseif num1 = 3 Then  
...  
Else  
...  
End If
```





# ОПЕРАТОРЫ УСЛОВНОГО ПЕРЕХОДА

Конструкция `Select Case` позволяет обрабатывать сразу несколько условий. После слов `Select Case` указывается сравниваемое выражение. Значение этого выражения последовательно сравнивается со значениями, помещенными после оператора `Case`. И в случае если значения совпали, то выполняется блок команд, помещенных после данного оператора `Case`. Конструкция завершается словами `End Select`. Если необходимо определить действия, которые будут выполняться, если совпадений не выявлено, то используется оператор `Case Else`, после которого помещаем блок действий по умолчанию. Блок `Case Else` необязателен и может не употребляться.



# ОПЕРАТОРЫ УСЛОВНОГО ПЕРЕХОДА

```
Dim R As New Random()
```

```
Dim num1 As Integer = R.Next(100)
```

```
Select Case num1
```

```
    Case 1
```

```
        Console.WriteLine("Число num1 равно 1")
```

```
    Case 2
```

```
        Console.WriteLine("Число num1 равно 2")
```

```
    Case 3 To 25
```

```
        Console.WriteLine("Число num1 находится на отрезке от 3 до 25")
```

```
    Case Else
```

```
        Console.WriteLine("Число num1 больше 25")
```

```
End Select
```



# ОПЕРАТОР БЕЗУСЛОВНОГО ПЕРЕХОДА

Всегда изменяет порядок выполнения операторов в процедуре или функции VBA. При этом VBA не проверяет никаких условий, а просто переходит к выполнению кода с другого места. Оператор GoTo имеет следующий синтаксис:

GoTo Метка

Метка – любое обозначение или номер строки в той же процедуре или функции, которая содержит оператор GoTo. При выполнении оператора GoTo VBA немедленно переходит к выполнению оператора в строке, определенной с помощью метки.



# ОПЕРАТОР БЕЗУСЛОВНОГО ПЕРЕХОДА

```
Dim time As String = "day"
```

```
    If time = "day" Then
```

```
        GoTo M1
```

```
    Else
```

```
        GoTo M2
```

```
    End If
```

```
M1:    Console.WriteLine("Hello World")
```

```
M2:    Console.WriteLine("Good Bye World")
```



# ЦИКЛ For..Next

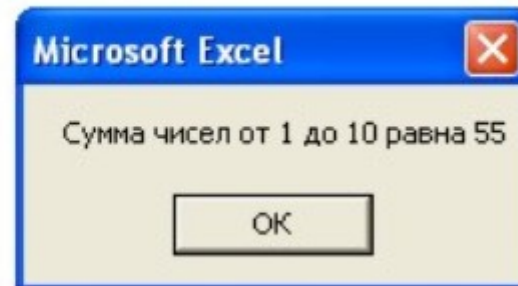
СИНТАКСИС:

For Counter = Start To End [Step  
StepSize]

Statements

Next [counter]

```
Sub example_01()  
  
    Dim i As Integer, SUM As Integer  
  
    SUM = 0  
  
    For i = 1 To 10  
        SUM = SUM + i  
    Next i  
  
    MsgBox "Сумма чисел от 1 до 10 равна " & SUM  
  
End Sub
```



# Цикл For Each..Next

СИНТАКСИС:

For Each Element In Group  
Statements Next [Element]

```
Function SheetExists(sName As String) As Boolean

    Dim aSheet As Object
    SheetExists = False

    'ЦИКЛ ПО ВСЕМ ЛИСТАМ СО СРАВНЕНИЕМ ИМЕН С sName
    For Each aSheet In ActiveWorkbook.Sheets
        If (StrComp(aSheet.Name, sName, 1) = 0) Then
            SheetExists = True
            Exit For
        End If
    Next aSheet

End Function
```

# Цикл Do .. While

Синтаксис:

Do While Condition  
Statements  
Loop

```
Sub example_02()  
    Dim i As Integer, SUM As Integer  
  
    i = 1: SUM = 0  
  
    Do While i <= 10  
        SUM = SUM + i  
        i = i + 1  
    Loop  
  
    MsgBox ("Сумма цифр от 1 до 10 = " & SUM)  
  
End Sub
```





# Цикл Do .. Until

VBA выполняет цикл пока логическое выражение, представленное с помощью Condition, равно False.

Синтаксис:

Do Until Condition    Statements Loop



# Цикл Do .. Until

```
Sub example_02()  
    Dim i As Integer, SUM As Integer  
  
    i = 1: SUM = 0  
  
    Do While i <= 10  
        SUM = SUM + i  
        i = i + 1  
    Loop  
  
    MsgBox ("Сумма цифр от 1 до 10 = " & SUM)  
  
End Sub
```

