

## Оператор цикла *For /Next*

Оператор цикла **For /Next** представляет собой цикл типа арифметической прогрессии.

*Синтаксис:*

<b>For</b> V=a1 <b>To</b> a2 [ <b>Step</b> a3]	- заголовок цикла
[ Операторы]	
<b>[Exit For]</b>	- тело цикла
[ Операторы]	
<b>Next</b> [V]	- конец цикла

**For** - для;

**To** - до;

**Step** - шаг;

**Next** – следующий.

*V* – параметр (счетчик) цикла – переменная.

*a1, a2, a3* – арифметические выражения:

*a1* – начальное значение параметра цикла;

*a2* – конечное значение параметра цикла;

*a3* – шаг изменения параметра цикла (число, которое прибавляется к параметру цикла).

В качестве параметра цикла можно использовать как вещественные, так и целые переменные.

Пример.

```
Private Sub Button1_Click()  
Dim s As Single, i As Integer  
s=0  
For i = 1 To 10 Step 1  
s = s + i ^ 2  
Next i  
TextBox1.Text=CStr(s)  
End Sub
```

## Порядок выполнения оператора цикла:

1. Вычисляются значения арифметических выражений  $a_1, a_2, a_3$ .
2. Параметру цикла присваивается начальное значение.
3. Проверяется условие окончания цикла. Если это условие не выполнено, то переходим к п.4. В противном случае управление передается оператору, следующему за оператором цикла.
4. Выполняются операторы, составляющие тело цикла.
5. Изменяется значение параметра цикла. К значению параметра добавляется шаг. Затем счет повторяется с п.3.

## При организации циклов необходимо учитывать следующее:

1. Число повторений оператора цикла определяется по формуле:

$$N = \begin{cases} \text{int}\left(\frac{a_2 - a_1}{a_3}\right) + 1, & \text{если } \frac{a_2 - a_1}{a_3} \geq 0 \\ 0, & \text{в остальных случаях} \end{cases}$$

2. Войти в область действия цикла можно только через его заголовок.

3. В общем случае, при выходе из цикла, параметр цикла не определен. Для альтернативного выхода из цикла используется оператор **Exit For**.

4. Переменные, входящие в заголовок цикла, могут изменять свои значения внутри области действия цикла. При этом *изменение в теле цикла начального значения параметра цикла, конечного значения параметра цикла и шага не влияет на количество повторений операторов тела цикла и на изменение параметра цикла*

5. Если шаг равен 1, то можно воспользоваться сокращенной формой записи заголовка цикла.

**For V=a1 To a2**

6. В операторе конца цикла параметр цикла может быть опущен, т.е. допустима запись:

**Next**

7. Тело цикла может содержать внутри себя другие циклы. Такая конструкция называется *вложенными циклами*.

# Базовые программы обработки одномерного массива

## 1) Ввод одномерного массива

Пусть имеется одномерный массив из  $n$  элементов

$x_1, x_2, \dots, x_n$ .

Опишем и введем этот массив:

*а) Ввод при помощи текстового окна:*

```
Dim x() As Single
Private Sub Button1_Click()
Dim n As Integer, i As Integer
n = CSng(InputBox("Введите n"))
ReDim x(0 To n-1)
For i = 0 To n-1
x(i) = CSng(TextBox1.Text)
Next i
End Sub
```

*b) Ввод при помощи функции InputBox:*

```
Dim x() As Single
Private Sub Button1_Click()
Dim n As Integer, i As Integer
n = CSng(InputBox("Введите n"))
ReDim x(0 To n-1)
For i = 0 To n-1
x(i) = CSng(InputBox("x(" & CStr(i) & ")", "массив
аргументов"))
Next i
End Sub
```

**2) Вычисление суммы элементов массива:**  $S = \sum_{i=1}^n x_i$

s = 0

For i = 0 To n-1

s = s + x(i)

Next i

**3) Нахождение наибольшего элемента массива и  
порядкового номера этого элемента:**

Max = x(0)

For i = 0 To n-1

If x(i) > Max Then Max = x(i): nm = i

Next i



## 4) Вывод последовательности элементов одномерного массива

a) *Вывод при помощи текстового окна:*

```
Dim x() As Single
Private Sub Button1_Click()
n = CSng(InputBox("Введите n"))
ReDim x(0 To n-1)
For i = 0 To n-1
TextBox1.text= TextBox1.text +CStr(x(i))+vbCrLf
Next i
End Sub
```

*b) Вывод при помощи функции MsgBox:*

```
Dim x() As Single
Private Sub Button1_Click()
n = CSng(InputBox("Введите n"))
ReDim x(0 To n-1)
For i = 0 To n-1
x(i) = MsgBox(CStr(x(i)), , "x(" & CStr(i) & ")")
Next i
End Sub
```

## Вложенные циклы

В VB нет ограничений на операторы, составляющие тело цикла. Среди этих операторов, в свою очередь, могут быть *операторы цикла*.

Когда цикл содержит другой цикл, то говорят, что имеют место *вложенные циклы*:

```
For i = 0 To n-1
  For j = 0 To m-1
    s = s + a(i, j)
  Next j
Next i
```

В нашем примере цикл по **j** вложен в цикл по **i**.

Если в программе имеются два цикла, то могут быть представлены всего два варианта:  
либо эти циклы не пересекаются,  
либо они вложенные.  
Обозначив цикл «[», можно схематически изобразить упомянутые варианты:

		
Не пересекающиеся циклы	Вложенные циклы	Запрещенная схема

## Правила записи вложенных циклов:

1) При записи вложенных циклов используется правило «матрешки», т.е. сначала закрывается цикл, который открылся последним, затем закрывается цикл, который открылся предпоследним и т.д.

Заголовок первого цикла

    Заголовок второго цикла

        Заголовок третьего цикла

        .....

        Конец третьего цикла

    Конец второго цикла

Конец первого цикла

2) Для каждого значения параметра *внешнего* цикла, параметр *внутреннего* цикла изменяется от начального значения до конечного с заданным шагом.

Пример. Что будет напечатано и сколько раз выполнится программа?

```
For i = 1 To 2  
For j = 1 To 3  
K=i*j  
TextBox1.Text= TextBox1.Text  
+CStr(i)+” “+CStr(j)+ “ “+CStr(  
k) +vbCrLf  
Next j  
Next i
```

Ответ:

1	1	1
1	2	2
1	3	3
2	1	2
2	2	4
2	3	6

3) В качестве параметра цикла для вложенных циклов *нельзя* использовать одну и ту же переменную, т.к. изменение переменной внешнего цикла нарушится изменениями переменной внутреннего цикла.

4) Если циклы **For /Next** вложены друг в друга, то в качестве последнего оператора можно использовать один оператор **Next** (если циклы заканчиваются одновременно). В этом случае в операторе **Next** через запятую нужно указать параметры всех закрывающихся циклов в том порядке, как эти циклы закрываются.

```
t = 1
For i = 0 To n-1
  For j = 0 To m-1
    t = t * a(i, j)
  Next i, j
```