

Лекция 4. Циклы



А.Ф. ЗУБАИРОВ

Счётный цикл



- Повторяет заданные действия определённое количество раз.
- Синтаксис:

```
For счетчик = нач To кон [Step шаг]  
    [действия]  
    [Exit For]  
    [действия]  
Next [счетчик]
```

СЧЁТНЫЙ ЦИКЛ



- **For** – начало цикла
- Счётчик – целочисленная переменная, значение которой после каждого прохода автоматически увеличивается.
- Нач – начальное значение счётчика (математическое выражение).
- Кон – конечное значение счётчика (математическое выражение) – условие выхода из цикла.
- Шаг – значение, на которое изменяется счётчик после каждого прохода цикла (математическое выражение). Если значение шага не указано, оно принимается равным единице.
- **Exit For** – позволяет выйти из цикла до того, как счётчик достигнет конечного значения.
- **Next** счётчик – конец цикла.

Счётный цикл



```
For I = 1 To 10
```

```
    MsgBox "Здравствуй, мир!"
```

```
Next I
```

Счетный цикл



- Подсчитать сумму натуральных чисел в заданном пользователем отрезке.

```
Dim A As Integer, B As Integer, Sum As  
Integer, I As Integer
```

```
... ` Запрос A и B у пользователя
```

```
Sum = 0
```

```
For I = A To B
```

```
    Sum = Sum + I
```

```
Next I
```

```
MsgBox Sum
```

Условный цикл While



- Повторяет заданные действия до тех пор, пока выполняется заданное условие.

While условие
 действия

Wend

- **While** – начало цикла.
- **Wend** – конец цикла.

Условный цикл While



- Найти максимальный делитель заданного числа.

```
Dim N As Integer, Divider As Integer
```

```
N = Val(InputBox("Введите число"))
```

```
Divider = N \ 2
```

```
While N Mod Divider <> 0
```

```
    Divider = Divider - 1
```

```
Wend
```

```
MsgBox "Мак делитель:" + CStr(Divider)
```

Условный цикл Do



- Условный цикл Do выполняет действия, пока выполняется, либо пока не начнёт выполняться заданное условие.
- Цикл с предусловием (сначала проверяется условие, затем выполняются действия):

```
Do { While | Until } условие
```

```
    выражения
```

```
    [ Exit Do ]
```

```
    выражения
```

```
Loop
```


Условный цикл Do



- Do – начало цикла.
- Если выбрано While, цикл выполняется, пока заданное условие истинно.
- Если выбрано Until, цикл выполняется, пока заданное условие не станет истинным (пока условие ложно).
- Exit Do прекращает выполнение цикла принудительно в независимости от того, выполняется или не выполняется заданное условие.
- Loop – конец цикла.

Условный цикл Do



- Цикл с постусловием (сначала выполняются действия, затем проверяется условие):

Do

выражения

[**Exit Do**]

выражения

Loop {**While** | **Until**} условие

Условный цикл Do



- Найти максимальное простое число, не превосходящее заданное.

```
Dim N As Integer, M As Integer
N = Val(InputBox("Введите число"))
M = N \ 2
Do While N <> 2
    If N Mod M = 0 Then
        N = N - 1
        M = N \ 2
    Else
        If M = 2 Then
            Exit Do
        Else
            M = M - 1
        End If
    End If
Loop
MsgBox N
```

Условный цикл Do



```
Dim MaxWeight As Single, TotalWeight As Single,  
CurrentWeight As Single, Cash As Single, TotalCost As  
Single
```

```
MaxWeight = Val(InputBox("Введите грузоподъемность"))  
Cash = Val(InputBox("Введите сумму на счёте"))
```

```
Do
```

```
    CurrentWeight = Val(InputBox("Масса груза:"))  
    TotalWeight = TotalWeight + CurrentWeight  
    TotalCost = TotalCost + CurrentWeight * 5
```

```
Loop Until MaxWeight < TotalWeight Or TotalCost >  
Cash
```

```
    TotalWeight = TotalWeight - CurrentWeight  
    TotalCost = TotalCost - CurrentWeight * 5
```

```
    MsgBox "Груз массой " & TotalWeight & " принят.  
    Стоимость доставки " & TotalCost
```

Подпрограммы



- Подпрограмма - поименованная или иным образом идентифицированная часть программы, содержащая описание определённого набора действий. Подпрограмма может быть многократно вызвана из разных частей программы.
- Подпрограммы позволяют не повторять в программе идентичные блоки кода, а описывать их однократно и вызывать по мере необходимости.

Подпрограммы



- Выделение набора действий в подпрограмму и вызов её по мере необходимости позволяет логически выделить целостную подзадачу, имеющую типовое решение.
- Любое изменение (исправление ошибки, оптимизация, расширение функциональности), сделанное в подпрограмме, автоматически отражается на всех её вызовах, в то время как при дублировании каждое изменение необходимо вносить в каждое вхождение изменяемого кода.
- Даже в тех случаях, когда в подпрограмму выделяется однократно производимый набор действий, это оправдано, так как позволяет сократить размеры целостных блоков кода, составляющих программу, то есть сделать программу более понятной и обозримой.

Подпрограммы

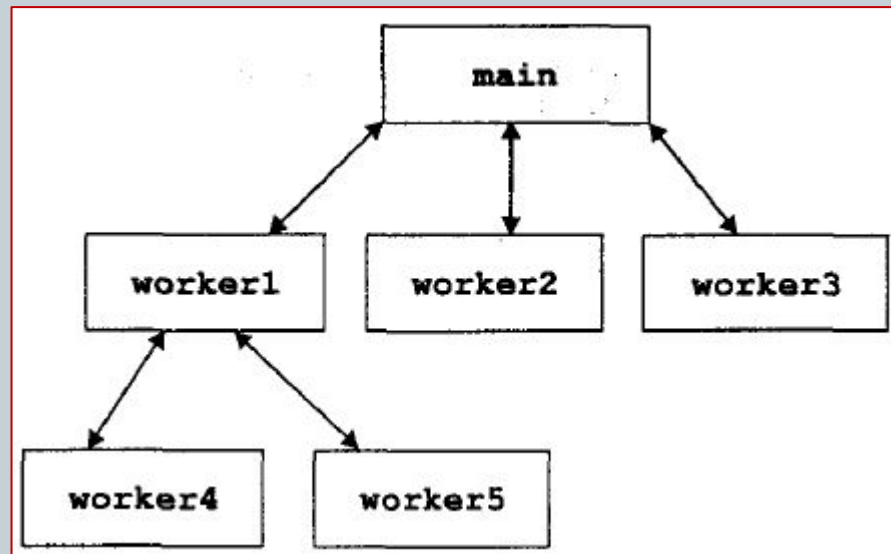


- **Процедура** — это независимая именованная часть программы, которую после однократного описания можно многократно вызвать по имени из последующих частей программы для выполнения определенных действий.
- **Функция** — подпрограмма, обрабатывающая переданные ей аргументы и возвращающая в конце работы значение. Функция может использоваться в математических выражениях, т. к. возвращает результат.

Подпрограммы



Вызов подпрограммы – обращение к подпрограмме. Главная программы вызывает подпрограммы. Каждая из подпрограмм в свою очередь также может вызывать другие подпрограммы.



Параметры подпрограмм



- Все переменные, объявленные внутри подпрограммы являются *локальными переменными* – они известны только той подпрограмме, в которой определены.
- Параметры позволяют подпрограммам обмениваться информацией.
- Параметры, указываемые при описании подпрограммы называются *формальными*.
- Параметры, передаваемые в подпрограмму при вызове, называются *фактическими*.
- При вызове подпрограммы фактические параметры, указанные в команде вызова, становятся значениями соответствующих формальных параметров, чем и обеспечивается передача данных в подпрограмму.
- Количество, порядок и тип формальных и фактических параметров должны совпадать.

Процедура



- Объявление процедуры:

```
Sub Имя (список_формальных_параметров)  
    действия
```

```
End Sub
```

- Список формальных параметров:

```
имя1 As тип1, имя2 As тип2, ...
```

Процедура



- Процедура ShowMax принимает 3 параметра и выводит на экран максимальный.

```
Sub ShowMax (A As Integer, B As Integer, C As
Integer)
    If A > B And A > C Then
        MsgBox A
    ElseIf B > C Then
        MsgBox B
    Else
        MsgBox C
    End If
End Sub
```

Процедура



- **Вызов процедуры:**

Имя список_фактических_параметров

- **Список фактических параметров:**

выражение1, выражение2, ...

```
Dim X As Integer, Y As Integer, Z As  
Integer
```

...

```
ShowMax X + 6, Y - Z, Z * 3
```

Процедура



ShowMax X + 6, Y - Z, Z * 3

X + 6

Y - Z

Z * 3

Sub ShowMax(A **As** Integer, B **As** Integer, C **As** Integer)

Функция



- Объявление функции:

```
Function Имя (спис_форм_парам) As тип  
    действия
```

```
End Function
```

ТИП – тип значения, которое возвращает функция.

ФУНКЦИЯ



- Для того, чтобы вернуть значение, необходимо присвоить его специальной переменной, имя которой совпадает с именем функции. Эту переменную объявлять не нужно.

```
Function Max(A As Integer, B As Integer, C As
Integer) As Integer
    If A > B And A > C Then
        Max = A
    ElseIf B > C Then
        Max = B
    Else
        Max = C
    End If
End Function
```

Функция



- **Вызов функции:**

имя (факт_парам)

- **Так как функция возвращает значение, функции используются в выражениях:**

```
Result = X + Y * Max(Var1, Var2, Var3)
```


Передача параметров



- **Передача по значению:**

В памяти создается копия передаваемого значения и подпрограмма работает с созданной копией. При изменении формального параметра внутри подпрограммы фактический параметр в программе не изменяется.

- **Передача по ссылке:**

Подпрограмма работает с той же ячейкой памяти, в которой хранится передаваемое значение.

При изменении формального параметра внутри подпрограммы изменяется значение фактического параметра в программе. *По умолчанию параметры передаются по ссылке.*

Передача параметров



- Для передачи по значению в списке формальных параметров перед параметром указывается

ByVal:

A As Integer, **ByVal** C As Integer

- Для передачи по ссылке в списке формальных параметров перед параметром указывается

ByRef, либо ничего не указывается.

ByVal A **As** Integer, **ByRef** C **As** Integer,
D **As** Single

Передача по значению



```
Function Foo(ByVal A As Integer) As Integer
    A = A + 1
    Foo = A
End Function
```

```
Sub Main()
    Dim Var1 As Integer, Var2 As Integer
    Var1 = 1           ` Var1 - 1
    Var2 = Foo(Var1)  ` Var1 - 1, Var2 - 2
End Sub
```

Передача по ссылке



```
Function Foo(ByRef A As Integer) As Integer
    A = A + 1
    Foo = A
End Function

Sub Main()
    Dim Var1 As Integer, Var2 As Integer
    Var1 = 1           ` Var1 - 1
    Var2 = Foo(Var1)  ` Var1 - 2, Var2 - 2
End Sub
```