

БАЗОВАЯ СТРУКТУРА ЦИКЛ

Понятие цикла

- При решении задач часто приходится многократно проводить одни и те же действия, например вычислять значения функций по одним и тем же математическим формулам для различных значений входящих в них величин. Такие многократно повторяющиеся участки вычислительного процесса называются *циклами*.

Цикл – это оператор или группа операторов, которые программа многократно выполняет до тех пор, пока не получит команду начать выполнение чего-либо другого. Такая последовательность команд называется **телом цикла**.

Циклические алгоритмические структуры
бывают 2-х типов:

- **Циклы со счетчиком (арифметические циклы)**, в которых тело цикла выполняется определенное количество раз;
- **Циклы с условием (итерационные циклы)**, в которых тело цикла выполняется, пока условие истинно.

Для организации цикла необходимо выполнить следующие действия:

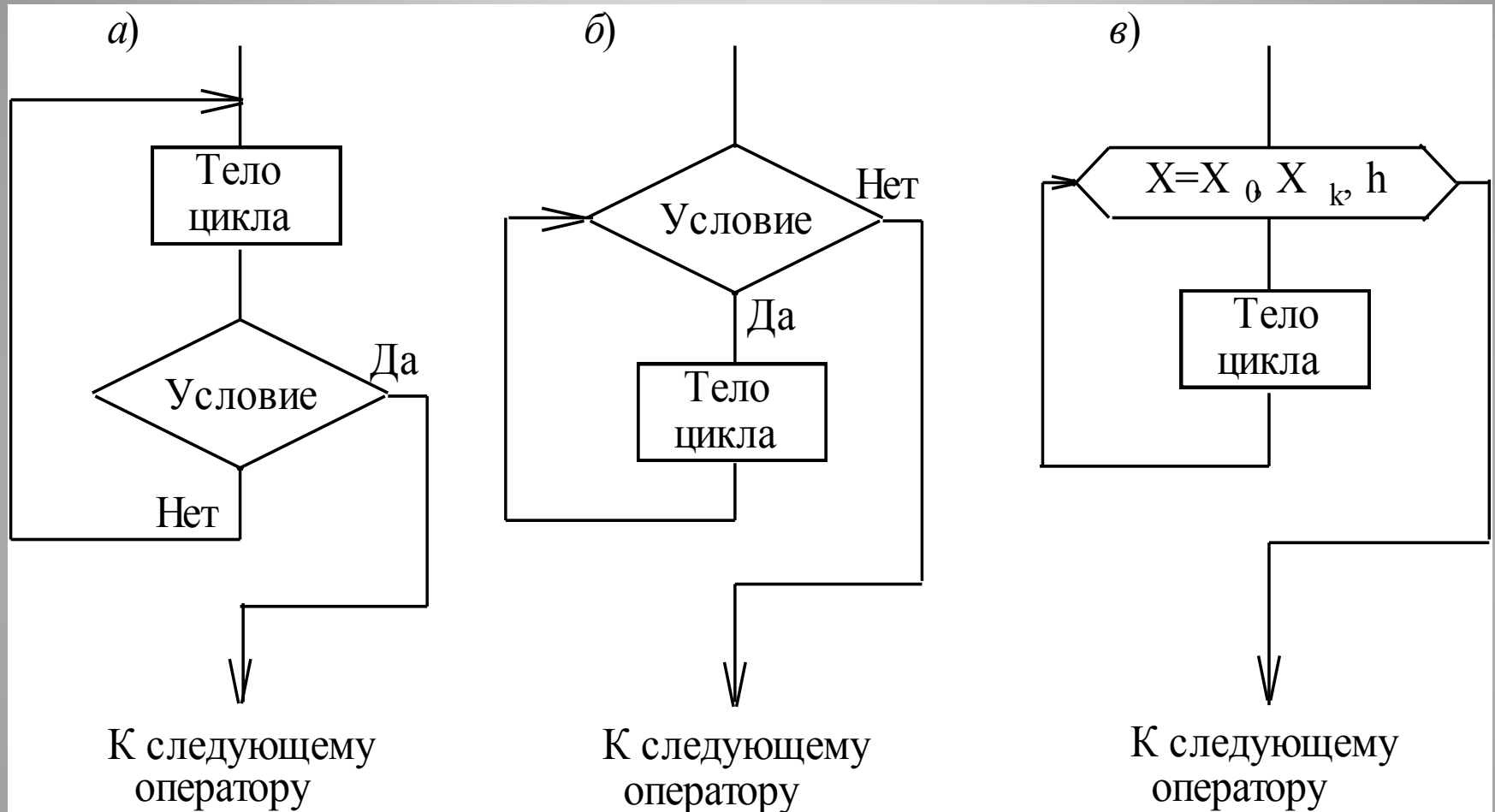
- 1) задать перед циклом начальное значение переменной, изменяющейся в цикле и шаг ее изменения;
- 2) произвести необходимые действия (вычисления);
- 3) изменить переменную на величину шага;
- 4) проверить условие окончания (или повторения) цикла;
- 5) переходить к п.2, если цикл не закончен.

Как видно из приведенного словесного алгоритма цикла, последние четыре действия повторяются многократно. Переменная, изменяющаяся в цикле, называется ***параметром цикла***. Действия, повторяемые многократно, называются ***телом цикла***.

Возможны три способа организации циклического алгоритма:

- 1) цикл “повторять до выполнения условия” (цикл с постусловием и цикл с предусловием);
- 2) цикл “повторять, пока выполняется условие” (цикл с постусловием и цикл с предусловием);
- 3) счетный цикл (цикл повторяется заданное число раз).

Схемы организации циклов



а – цикл с постусловием "повторять... до", б – цикл с предусловием "пока повторять...", в – счетный цикл

Циклы можно реализовать 4-мя способами:

- с использованием оператора условного перехода IF ... THEN (все циклы);
- с использованием оператора цикла FOR ... NEXT (счетный цикл);
- с использованием оператора цикла DO ... LOOP (все циклы);
- с использованием оператора цикла WHILE ... WEND (цикл "пока" с предусловием).

**Циклические программы с
использованием оператора
условного перехода IF ... THEN**

Примером цикл-ой программы с использованием If...Then является **табулирование функции** – вычисление значения функции в зависимости от аргумента, величина которого изменяется на постоянную величину, называемую шагом.

Пример. Протабулировать функцию $y=e^{2x}$ на интервале $16 \leq x \leq 40$ с шагом 4.

```
Sub task_1()  
'присваиваем начальное значение переменной x и номера столбца i, задаем шаг h  
x = 16: h = 4: i = 1  
'начало цикла  
метка: y = Exp(2 * x) 'вычисляем функцию  
'выводим результат в лист Excel  
Cells(1, i) = x  
Cells(2, i) = y  
'изменяем значение аргумента на величину шага  
x = x + h  
'изменяем значение номера столбца на величину шага  
i = i + 1  
'проверяем, не превысило ли значение x верхний предел  
'вычислений; если нет, то продолжаем вычисления,  
'если да - выходим из цикла  
If x <= 40 Then GoTo метка  
End Sub
```

операторы с метка: до IF...THEN повторяются неоднократно и представляют собой тело цикла.

Строка $x = x + h$ называется оператором изменения значения переменной цикла.

определяет условие повторения или окончания цикла

Результат выполнения программы

A2		f_x 78962960182680,7						
	A	B	C	D	E	F	G	H
1	16	20	24	28	32	36	40	
2	7,9E+13	2,35E+17	7,02E+20	2,09E+24	6,24E+27	1,86E+31	5,54E+34	
3								
4								
5								



Оператор арифметического цикла For...Next

Цикл FOR...NEXT – это цикл с заранее заданным количеством повторений.

Формат записи оператора:

.....

FOR счетчик = начало **TO** конец [**STEP** шаг]

[операторы тела цикла]

NEXT счетчик

.....

- Счетчик – переменная, используемая как счетчик цикла (параметр цикла);
- начало – начальное значение счетчика;
- конец – конечное значение счетчика;
- шаг – шаг изменения значения счетчика.

Начало, конец, шаг могут быть числовым значением, другой переменной, арифметическим выражением.

Примечание: Параметры, указанные в квадратных скобках, являются необязательными (в программе могут опускаться). Если операнд STEP опущен, то значение шага = 1.

При первом проходе цикла параметр цикла (счетчик) равен начальному значению. При достижении оператора NEXT к значению счетчика прибавляется значение шага. Если счетчик не превышает конечного значения, цикл повторяется. Далее цикл выполняется до тех пор, пока текущее значение счетчика не выйдет за рамки конеч. знач. цикла.

Примечание.

- 1) Если начало = конечному зн. счетчика, то цикл выполняется один раз, вне зависимости от знач. шага.
- 2) Если шаг = нулю, цикл продолжается неопределенное количество раз (т. е. программа "зависает").
- 3) Цикл FOR ... NEXT выполняется только в том случае, если нач. знач-е параметра цикла плюс шаг $<$ или $=$ конеч. знач-ю параметра цикла. Если конеч. знач-е счетчика $<$ нач. знач-я, то шаг должен быть отрицательным.

Пример. Протабулировать функцию $y=e^{2x}$ на интервале $16 \leq x \leq 40$ с шагом 4.

```
Sub task_()  
Dim x As Integer  
Dim y As Integer, i As Byte  
i = 1  
For x = 16 To 40 Step 4  
    y = Exp(2 * x)  
    Cells(1, i) = x  
    Cells(2, i) = y  
    i = i + 1  
Next  
End Sub
```

Тело цикла
выполняется до
тех пор пока
значение x не
превысит
конечного
значения цикла

Итерационные алгоритмы

Оператор цикла Do...Loop

DO...LOOP универсальный оператор. С его помощью можно организовать любой цикл (с постусловием, с предусловием, цикл "ПОКА", цикл "ДО" и счетный цикл). Однако DO...LOOP удобнее всего использовать при организации **итерационных циклов**, т. е. циклов с заранее неизвестным числом повторений

Главной особенностью циклов с условием является **условие**, которое может быть любым выражением, принимающим значение **True (Истина)** или **False (Ложь)**.
В VBA есть два основных цикла с условием.

Цикл с предусловием

Do While <Условие>

Тело цикла

Loop

Do Until <Условие>

Тело цикла

Loop

Если **While**, то тело цикла выполняется,
когда <Условие> \equiv **True**,

если **Until**, то тело цикла выполняется,
когда <Условие> \equiv **False**

Цикл с постусловием

Do

Тело цикла

Loop While <Условие>

Do

Тело цикла

Loop Until <Условие>

В этом варианте операторы цикла будут выполнены один раз обязательно, а будет ли их повторение, зависит от выполнения/невыполнения условия.

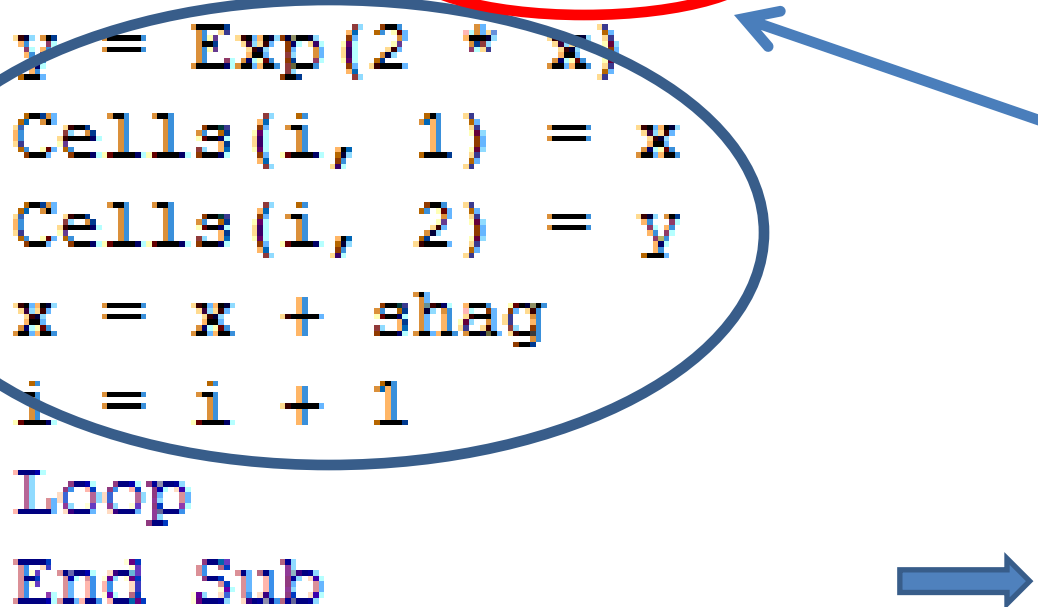
Условие – любое логическое выражение, допустимое на языке VB.

Тело цикла – любое количество операторов, которые составляют тело цикла.

Примеры решения задач с использованием оператора цикла DO...LOOP с **предусловием** (с проверкой условия логического выражения в начале)

Вычислить функцию $y=e^{2x}$ на интервале $16 \leq x \leq 40$ с шагом 4.


```
Sub Task_3 ()  
x = 16: shag = 4: i = 1  
Do While x <= 40  
y = Exp(2 * x)  
Cells(i, 1) = x  
Cells(i, 2) = y  
x = x + shag  
i = i + 1  
Loop  
End Sub
```



Тело цикла
работает до
тех пор пока
условие
истинно (True)

```
Sub Task_3 ()  
x = 16: shag = 4: i = 1  
Do Until x > 40  
y = Exp(2 * x)  
Cells(i, 1) = x  
Cells(i, 2) = y  
x = x + shag  
i = i + 1  
Loop  
End Sub
```

Тело цикла
работает до
тех пор пока
условие
ложно
(False)

Примеры решения задач с использованием оператора цикла DO...LOOP с **постусловием** (проверкой условия логического выражения в конце)

```
Sub Task_4()  
x = 16: shag = 4: i = 1  
Do  
y = Exp(2 * x)  
Cells(i, 1) = x  
Cells(i, 2) = y  
x = x + shag  
i = i + 1  
Loop While x <= 40  
End Sub
```

```
Sub Task_5()  
x = 16: shag = 4: i = 1  
Do  
y = Exp(2 * x)  
Cells(i, 1) = x  
Cells(i, 2) = y  
x = x + shag  
i = i + 1  
Loop Until x > 40  
End Sub
```

Результат программ

	A	B	C
1	16	7,9E+13	
2	20	2,35E+17	
3	24	7,02E+20	
4	28	2,09E+24	
5	32	6,24E+27	
6	36	1,86E+31	
7	40	5,54E+34	
8			
9			

Оператор цикла WHILE ...
WEND

WHILE ... WEND – управляющий оператор. Выполняет блок операторов до тех пор, пока указанное условие истинно.

Условие может быть только в начале цикла. Этот оператор дублирует аналогичный цикл DO WHILE ... LOOP

Общий вид оператора

While <Условие>

Тело цикла

Wend

Условие – любое логическое выражение.

Тело цикла– любое количество операторов, которые составляют тело цикла.

Вычисление суммы

Вычисление конечной суммы сводится к нахождению суммы заданного количества слагаемых:

$$S = f(1) + f(2) + \dots + f(n) = \sum_{i=1}^n f(i)$$

где i – номер слагаемого; $f(i)$ – слагаемое с номером i .

Вычисление организуется в виде циклического алгоритма, когда при каждом прохождении цикла номер слагаемого i увеличивается на 1, а сумма изменяется на величину i -го слагаемого:

$$S_i = S_{i-1} + f(i)$$

Цикл повторяется до тех пор, пока не будут просуммированы все n слагаемых. Для того, чтобы начальное содержимое ячейки суммы не исказило результат, сумма предварительно должна быть обнулена

$$S_0 = 0$$

Вывод результата, поскольку он является единственным, осуществляется после окончания работы цикла.

Пример. Вычислить сумму ряда N натуральных чисел.

```
Sub Demo_Summa()  
Dim i As Integer  
Dim sStart, sEnd As Integer  
Dim Sum As Long  
sStart = 1  
sEnd = InputBox("Введите длину ряда:", "Запрос программы")  
Sum = 0  
For i = sStart To sEnd  
Sum = Sum + i  
Next i  
MsgBox "Сумма чисел от " & sStart & " до " & sEnd & " равна " & Sum, ,  
"Решение задачи"  
End Sub
```

Запрос программы ✕

Введите длину ряда:

OK

Cancel

125

Решение задачи ✕

Сумма чисел от 1 до 125 равна 7875

OK

Вычисление произведения

Вычисление *конечного* *произведения*
представляет собой процесс нахождения
произведения заданного количества
сомножителей по формуле

$$P = f(1) \cdot f(2) \cdot \dots \cdot f(n) = \prod_{i=1}^n f(i)$$

Как и суммирование, вычисление произведения организуется с помощью циклического процесса по рекуррентному соотношению

$$P(i) = P_{i-1} \cdot f(i)$$

В отличие от суммирования начальное значение произведения $P_0 = 1$

Пример. Вычислить факториал числа N . $y = n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n = \prod_{i=1}^n i$

```
Sub Demo_Proizvedenie()
```

```
Dim n, i As Integer
```

```
Dim p As Double
```

```
n = InputBox("Введите число n= ", "Запрос программы")
```

```
p = 1
```

```
For i = 1 To n
```

```
p = p * i
```

```
Next i
```

```
MsgBox "Факториал чисел от 1 до " & n & " равен " & p, ,  
"Решение задачи"
```

```
End Sub
```

Запрос программы ✕

Введите целое число n=

50

OK

Cancel

Решение задачи ✕

Факториал чисел от 1 до 50 равен 3,04140932017134E+64

OK

Пример. Вычислить с помощью цикла наибольшее положительное число n , удовлетворяющее условию $4 \cdot n^3 - n^2 + 3 \cdot n < 2,5 \cdot 10^8$

Решение

```
Sub task()
```

```
n = 1
```

```
While 4 * n ^ 3 - n ^ 2 + 3 * n < 2.5E+08
```

```
n = n + 1
```

```
Wend
```

```
MsgBox "Наибольшее целое N=" & n - 1
```

```
End Sub
```