




Циклические алгоритмы



Цикл - это такая алгоритмическая структура, в которой осуществляется многократное повторение одной (или нескольких) команд.

Виды циклических алгоритмов

```
graph TD; A[Виды циклических алгоритмов] --> B[С условием]; A --> C[С параметром]; B --> D[Цикл с предусловием «ПОКА»]; B --> E[Цикл с постусловием «ДО»];
```

С условием

(заранее **НЕ**известно
сколько раз
выполняется действие)

**Цикл с
предусловием
(«ПОКА»)**

**Цикл с
постусловием
(«ДО»)**

С

параметром

(заранее **известно**
сколько раз
выполняется
действие)

Циклические алгоритмы

Виды

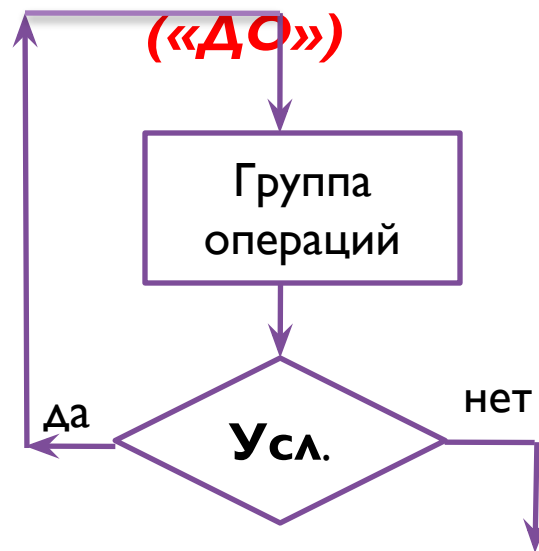
С условием

С параметром

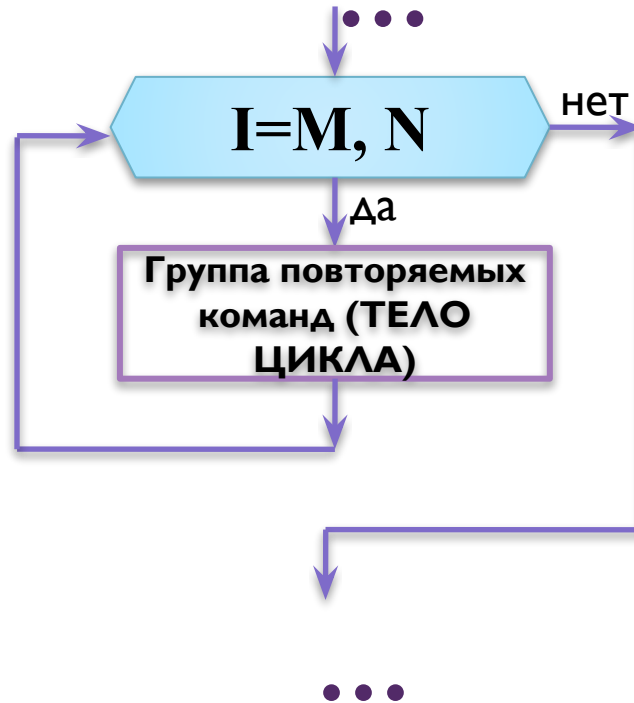
Цикл с
предусловием
(«ПОКА»)



Цикл с
постусловием
(«ДО»)



Цикл с параметром («для каждого»)



I – параметр цикла

M – начальное значение параметра

N – конечное значение

Форма записи цикла с параметром

for $i:=M$ **to** N **do** <оператор>

for $i:=M$ **downto** N **do** <оператор>

i - *параметр цикла* (переменная целого типа, ее значение автоматически изменяется на 1 (увеличивается или уменьшается))

M, N – *границы изменения параметра* (сначала I принимает значение M ; выполнение действий заканчивается, когда $I=N$)

<Оператор> – *оператор, который должен повторяться* (если операторов больше одного, то они заключаются в операторные скобки `begin ... end`)

Примеры

for I:=M **to** N **do** <оператор>

Например:

for I:=1 **to** 10 **do** writeln ('ПРИВЕТ!')

for I:=1 **to** 10 **do** writeln (I)

for I:=M **downto** N **do** <оператор>

Например:

for I:=10 **downto** 0 **do** writeln ('ПРИВЕТ!')

for I:=10 **downto** 0 **do** writeln (I)

Фрагмент программы

```
for i:=1 to 15 do
```

```
begin
```

```
    writeln ('число');
```

```
    readln (a);
```

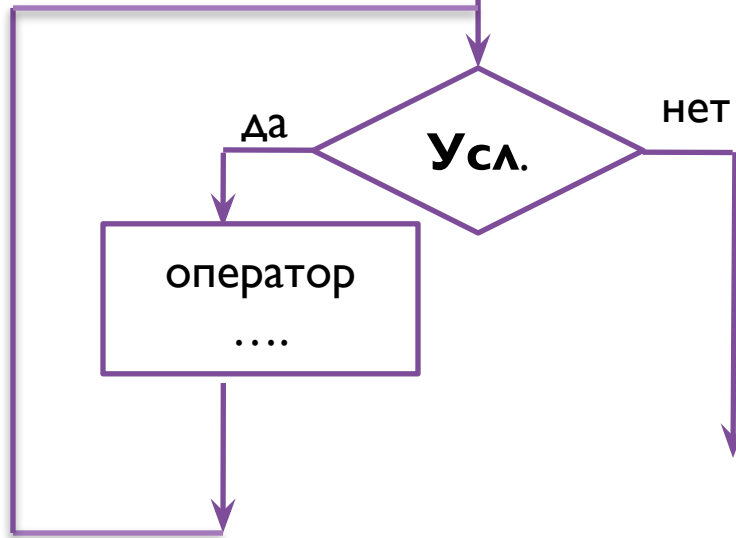
```
    s:=s+a;
```

```
end;
```

Примечание: если цикл содержит **несколько** команд, то применяются операторные скобки (**begin ... end**)

Циклы с условием

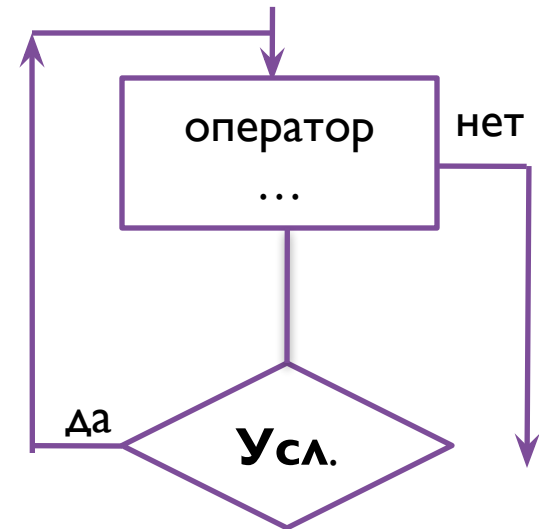
Цикл с предусловием



читается:

ПОКА выполняется условие,
ДЕЛАТЬ ...

Цикл с постусловием



читается:

ПОВТОРЯТЬ оператор **ДО**
некоторого условия

Цикл с предусловием (WHILE)

- известно *условие продолжения* цикла (не забыть и про завершение).
- условие должно проверяться **ДО** *выполнения тела цикла*.
- Цикл выполняется до тех пор, пока *условие ИСТИННО*.
- Оператор может не выполниться ни разу.

Если повторяется **один** оператор, то:
WHILE <условие> **DO** <оператор>

Если повторяется **несколько** операторов, то:

WHILE <условие> **DO**

begin

<оператор1>

<оператор2>

...

end

Цикл с постусловием (REPEAT)

- известно *условие завершения* цикла.
- условие должно проверяться **ПОСЛЕ** *выполнения* тела цикла.
- **Тело цикла** выполняется хотя бы **один** раз.

Repeat

Оператор 1

Оператор 2

....

Until <условие завершения>

- **Тело цикла** – это действие (или последовательность действий) , которое *повторяется*.
- **Условие завершения цикла** – определенное условие, при котором тело цикла **завершает** выполняться.
- **Условие продолжения цикла** - определенное условие, при котором тело цикла **продолжает** выполняться.

Пример

```
i:=1; a:=0;  
WHILE i<=10 DO  
  begin  
    a:=a+5;  
    i:=i+1;  
  end;  
writeln (a);
```

Ответ: 50

For k:=4 **to** 6 **do** sum:=sum+3

k:=4;

while k<=6 **do**

begin

sum:=sum+3;

k:=k+1;

end;

Writeln (sum);

k:=4;

repeat

sum:=sum+3;

k:=k+1

until k>6;

Writeln (sum);

Определите результат выполнения программы

№1 **k:=0;**
 s:=10;
 while s>0 do
 begin
 s:=s-3;
 k:=k+1;
 end;
 writeln('s=',s);
 writeln('k=',k)

№2 **k:=0;**
 s:=10;
 while s<0 do
 begin
 s:=s+3;
 k:=k+1;
 end;
 writeln('s=',s);
 writeln('k=',k)

Определите результат выполнения программы

№
3

```
m:=1; i:=1;  
while i<5 do i:=i+1;  
m:=m+i  
writeln(i,m)
```

Определите результат выполнения программы

№
4

x:=1; y:=1;

If y>0 then x:=x+1 else x:=x-1

while y>0 do

begin

y:=y-1;

x:=y;

end;

writeln(x,y)

Определите результат выполнения программы

№
5

d:=100;

i:=1;

Repeat

d:=d-5;

i:=i+1;

Until i=10

m:=m+1;

writeln(i,m)