

Программирование ПАСКАЛЬ



ЦИКЛЫ



ЦИКЛЫ

Определение. Цикл - это последовательность операторов, которая может выполняться более одного раза.

Определение. Циклический алгоритм - это алгоритм, содержащий один или несколько циклов.

в языке Паскаль существуют более удобные конструкции для организации циклов:

цикл с предусловием;

цикл с постусловием;

цикл со счетчиком.

Цикл с предусловием

Циклы с предусловием используются тогда, когда выполнение цикла связано с некоторым логическим условием. Оператор цикла с предусловием имеет две части: условие выполнения цикла и тело цикла.

При выполнении оператора while определенная группа операторов выполняется до тех пор, пока определенное в операторе while булево условие истинно. Если условие сразу ложно, то оператор не выполнится ни разу.

Общая форма записи следующая

```
while <булево выражение> do  
    begin  
        группа операторов  
    end;
```

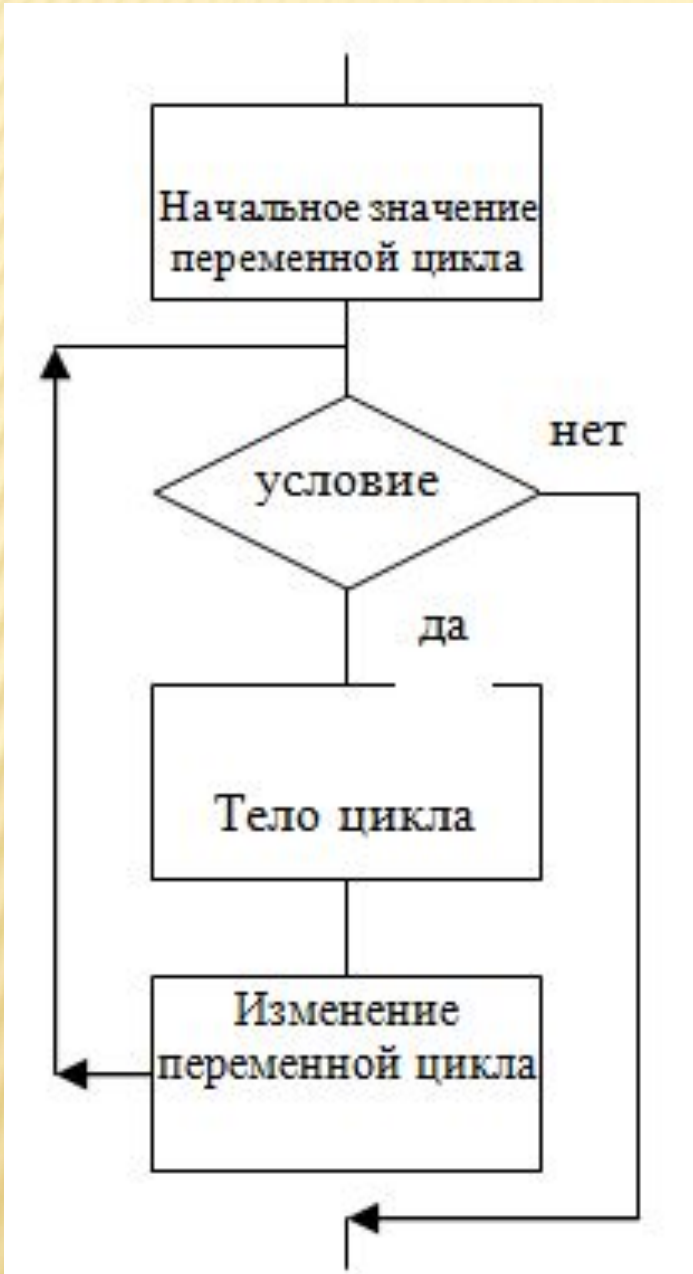
Это звучит примерно так:

```
пока выполняется это условие делай  
    от начала  
        группа операторов  
    до конца;
```

**При использовании цикла с предусловием надо помнить
следующее:**

- 1) значение условия выполнения цикла должно быть определено до начала цикла;
- 2) если значение условия истинно, то выполняется тело цикла, после чего повторяется проверка условия. Если условие ложно, то происходит выход из цикла;
- 3) хотя бы один из операторов, входящих в тело цикла, должен влиять на значение условия выполнения цикла, иначе цикл будет повторяться бесконечное число раз.

Пример: Используя оператор **while ... do** вычислить сумму
 $S=1+1/2+1/3+\dots+1/n$



```
Program while_1;  
Var S:real;  
    n,i:integer;  
Begin  
S:=0;  
writeln('Введите n=');  
readln(n);  
i:=1;  
while i<=n do } Условие  
  
    begin } Тело  
    S:=S+1/i;  
    i:=i+1;  
    end; } цикла  
writeln ('S=', S:10:6);  
readln;  
end.
```

Цикл с предусловием

Пример: Вычислить сумму $S=1+1/2+1/3+\dots+1/n$

```
Program w_1;
  var S:real;
      n,i:integer;
begin
  S:=0;
  writeln('введите n->');
  readln(n);
  i:=1;
  while i<=n do
    begin
      S:=S+1/i;
      i:=i+1;
    end;
  writeln ('S=', S:4:6);
  readln;
end.
```

Объяснение решения:

- ✓ Т.к. данная задача обладает свойством Массовость, значение **n** вводится вручную, (например, $n=5$)
- ✓ Вводим дополнительную переменную **i**, задаем ей первоначальное значение равное 1.
- ✓ Задаем условие **$i \leq n$** ,
- ✓ Начинаем считать сумму, обнулив первоначальное значение S
- ✓ **$S=S+1/i$** (цикл)
 - $S=0+1/1=1$ при $i=1$
 - $S=1+1/2=1,5$ при $i=2$
 - $S=1,5+1/3=1,833$ при $i=3$
 - $S=1,833+1/4=2,083$ при $i=4$
 - $S=2,083+1/5=2,2833$ при $i=5$ (Условие выполнено).
- ✓ **$i=i+1$** (Шаг равен 1 по условию задачи)
- ✓ Цикл закончен
- ✓ Вывод на экран результата.

Как изменить программу, чтобы вывести на экран все промежуточные вычисления суммы S?

Задача While1. Используя цикл **WHILE**, составить таблицу значений функции **Y=SIN(X)**, где x меняется от 0 до π с шагом $\pi/12$

```
Program While_1_1;  
var  
  x,y:real;  
begin  
  x:=0;  
  while x <= pi do  
  begin  
    y:=sin(x);  
    writeln('x=', x*180/3.14:3:0, ' y=',y:3:2);  
    x:=x+pi/12;  
  end;  
end.
```

Имя программы
Объявляем блок переменных (у нас их две x и y)

Начало программы
Задаем начальное значение x

Пока выполняется условие $x \leq \pi$
делаем

Начало цикла
Решение уравнения $y = \sin(x)$

Вывод на экран результата с переводом X из радиан в градусы при шаге изменения x , равном $x + \pi/12$
конец цикла
конец программы

Задача While2. Найти значения функции $Y=X^2+1/X^2$,
где X меняется от -2 до 2 с шагом 0,5.

```
Program w1_2;  
var  
  y,x,i:real;  
begin  
  x:=-2;  
  while x<=2 do  
  begin  
    y:=sqr(x)+1/sqr(x);  
    writeln('y=',y:3:3);  
    x:=x+0.5;  
  end;  
end.
```


Цикл с постусловием

REPEAT ... UNTIL - этот оператор переводится «повторять до тех пор, пока не станет истинным условие»,

Этот оператор соответствует следующей базовой структуре алгоритмов:

REPEAT

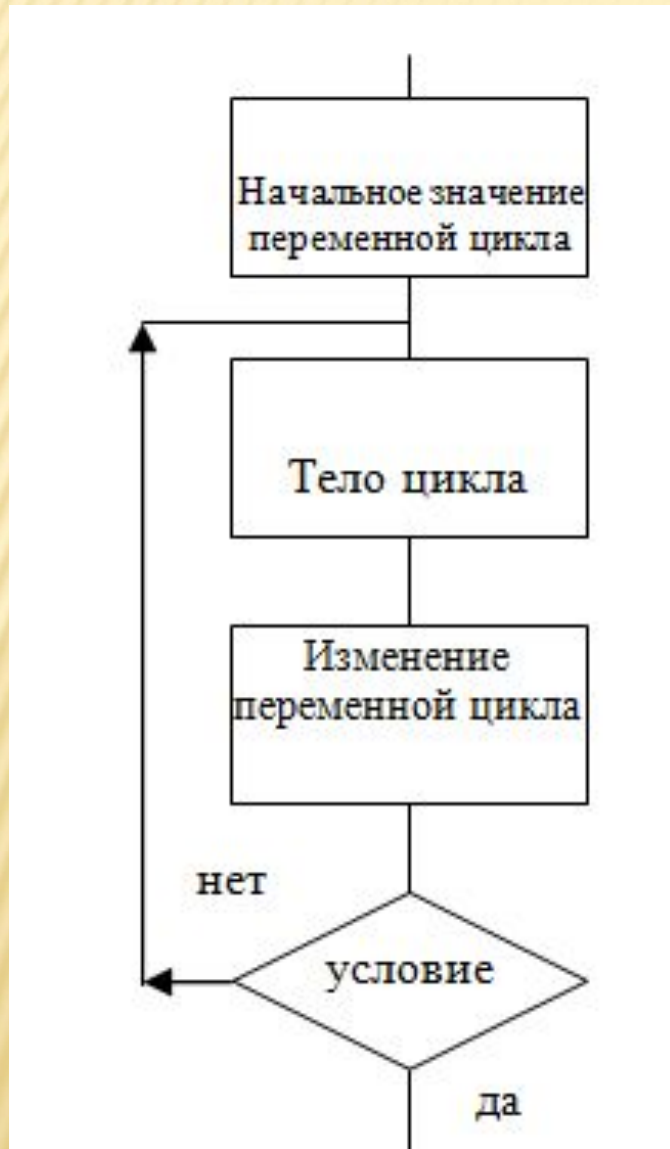
тело цикла;

изменение переменной цикла;

UNTIL *условие окончания цикла;*

Здесь тело цикла всегда выполняется хотя бы один раз. А *условием* является условие выхода из цикла.

Пример: Используя оператор **repeat ... until** вычислить сумму
 $S=1+1/2+1/3+\dots+1/n$



```
program rep_unt;
var S: real;
    n,i:integer;
begin
    writeln('Введите n->');
    readln(n);
    S:=0;
    i:=1;
    repeat
        S:=S+1/i;    - цикл
        i:=i+1;      - изм.переменной
    until i>n;      - условие
    writeln('S=',S:10:6);
    readln;
end.
```

Цикл с постусловием

Пример: Вычислить сумму $S=1+1/2+1/3+\dots+1/n$

```
program rep_unt;
var S: real;
    n,i:integer;
begin
  writeln('Введите n->');
  readln(n);
  S:=0;
  i:=1;
  repeat
    S:=S+1/i;
    i:=i+1;
  until i>n;
  writeln('S=',S:10:6);
  readln;
end.
```

Объяснение решения:

- ✓ Т.к. данная задача обладает свойством Массовость, значение **n** вводится вручную, (например, $n=5$)
 - ✓ Обнулили переменную S. Вводим дополнительную переменную **i**, задаем ей первоначальное значение равное 1.
 - ✓ Начинаем считать сумму
 - ✓ **$S=S+1/i$** (цикл)
 - $S=0+1/1=1$ при $i=1$
 - $S=1+1/2=1,5$ при $i=2$
 - $S=1,5+1/3=1,833$ при $i=3$
 - $S=1,833+1/4=2,083$ при $i=4$
 - $S=2,083+1/5=2,2833$ при $i=5$
 - ✓ **$i=i+1$** изменение переменной цикла (*Шаг равен 1 по условию задачи*)
 - ✓ Задаем условие **$i>n$** , когда условие выполнено
 - ✓ Цикл закончен
 - ✓ Вывод на экран результата.
- Как изменить программу, чтобы вывести на экран все промежуточные вычисления суммы S?**

Задача Rep..Un1. Используя цикл **REPEAT** составить таблицу значений функции $Y=e^{x/3}$, где x меняется от 1 до 6 с шагом 0,5.

```
program repeat_1;  
var x,y:real;  
begin  
    writeln('  x          y');  
    writeln('-----');  
    x:=1;  
    repeat  
        y:=EXP(x/3);  
        writeln(x:6:2,y:15:2);  
        x:=x+0.5;  
    until x>6;  
    writeln('-----');  
    readln;  
end.
```


Задача Rep..Un3. Написать программу, когда пользователь вводит значение N, а программа выводит на экран таблицу умножения на N от 1 до 10.

```
Program repeat_3;
```

```
var
```

```
  i,x:integer;
```

```
begin
```

```
  write('Введите число :');
```

```
  readln(x);
```

```
  i:=1;
```

```
    repeat
```

```
      p:=i*x;
```

```
  writeln(i:2,' * ',x,' = ',p:5);
```

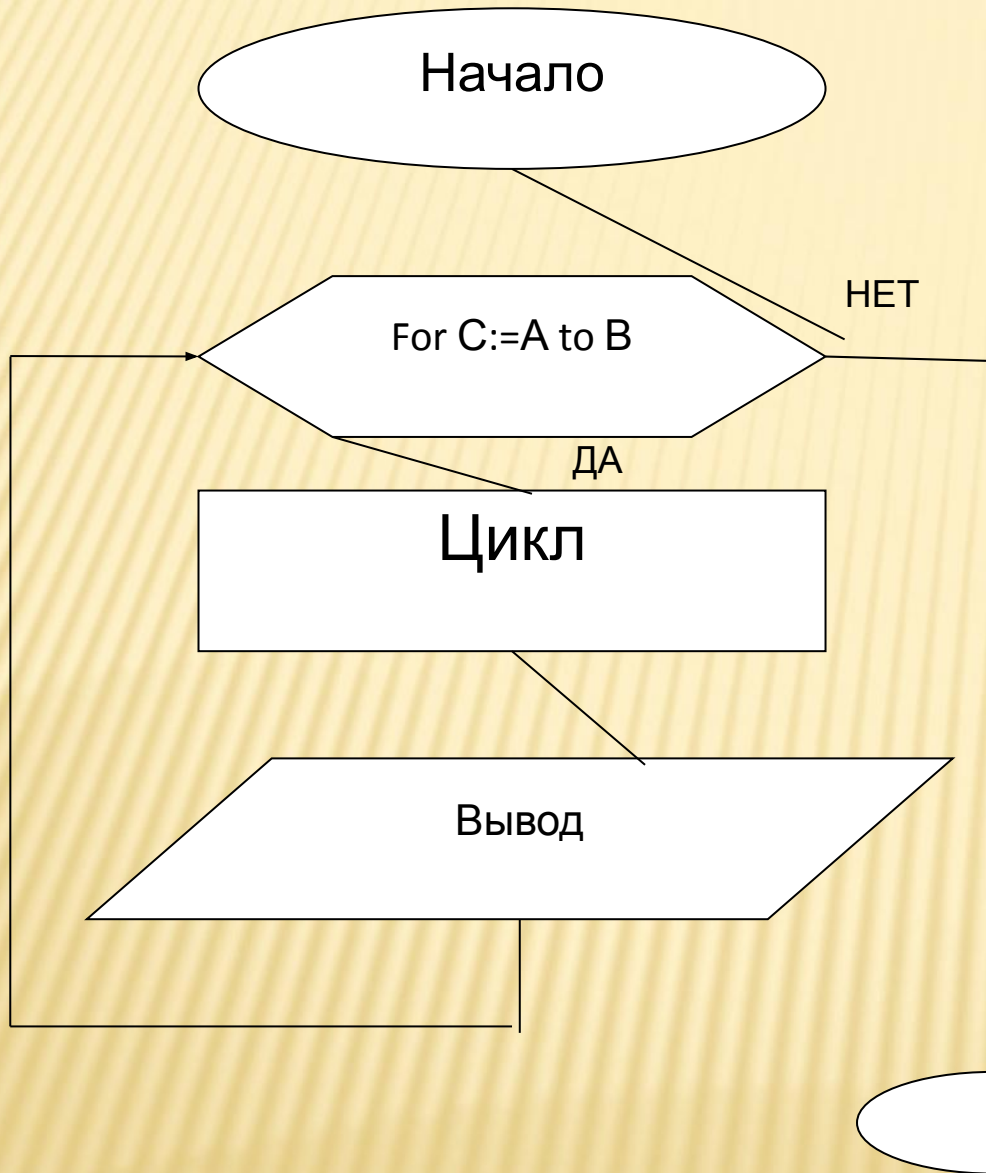
```
  i:=i+1;
```

```
    until i>10;
```

```
end.
```

ЦИКЛ С ПАРАМЕТРОМ (FOR – ДЛЯ)

Цикл с шагом 1 или -1



Этот оператор соответствует следующей базовой структуре алгоритмов:

FOR <переменная цикла> := <нач. значение> **TO** <конечн. значение>

DO

Здесь *for*, *to*, *do* - зарезерв. слова (для, до, выполнить);

begin

тело цикла;

end;

Этот цикл работает (начальное значение – конечное значение + 1) раз

В качестве переменной цикла в этих операторах можно использовать только переменные типа *INTEGER*.

Существует разновидность этого цикла с шагом -1 :

FOR *переменная цикла* := начальное значение

DOWNTO *конечное значение* **DO**

begin

тело цикла;

end;

Этот цикл работает (*конечное значение* – *начальное значение* + 1) раз.

Пример: Используя цикл FOR найти сумму $S=1+1/2+1/3+\dots+1/n$.

```
program for_1;  
var S:real;  
    i,n:integer;  
begin  
    writeln('Введите n->');  
    readln(n);  
    S:=0;  
    for i:=1 to n do  
        begin  
            S:=S+1/i;  
        end;  
    writeln('S=',S:10:6);  
    readln;  
end.
```


Задача For1.

Даны целые числа K и N ($N > 0$). Вывести N раз число K .

Организуем простой цикл от 1 до требуемого числа.

```
program For_1;  
var  
  k,n,i:integer;  
begin  
  write('введите число k=');  
  Readln(k);  
  write('введите число n=');  
  readln(n);  
  for i:=1 to n do  
    write(k,' ');  
end.
```

ЗАКЛЮЧЕНИЕ

ЦИКЛ С ПРЕДУСЛОВИЕМ И ЦИКЛ С ПОСТУСЛОВИЕМ

- НЕИЗВЕСТНО КОЛИЧЕСТВО ПОВТОРЕНИЙ ЦИКЛА;
- ШАГ МОЖЕТ БЫТЬ НЕ ЦЕЛЫМ ЧИСЛОМ;
- ЦИКЛ С ПРЕДУСЛОВИЕМ МОЖЕТ НЕ ВЫПОЛНИТЬСЯ НИ РАЗУ;
- ЦИКЛ С ПОСТУСЛОВИЕМ ХОТЯ БЫ 1 РАЗ ВЫПОЛНИТСЯ;

ЦИКЛ СО СЧЕТЧИКОМ

- ИЗВЕСТНО КОЛИЧЕСТВО ПОВТОРЕНИЙ;
- ШАГ ВСЕГДА РАВЕН 1 (Т.Е. ПЕРЕМЕННАЯ ЦЕЛОЧИСЛЕННАЯ).