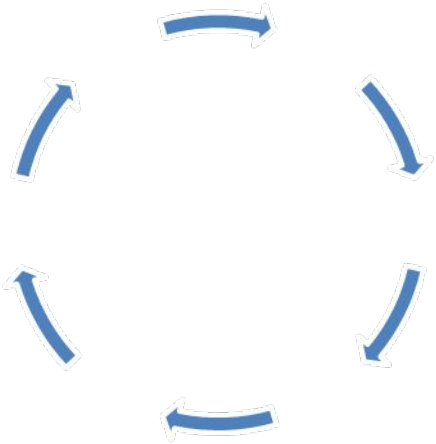
ЦИКЛЫ.

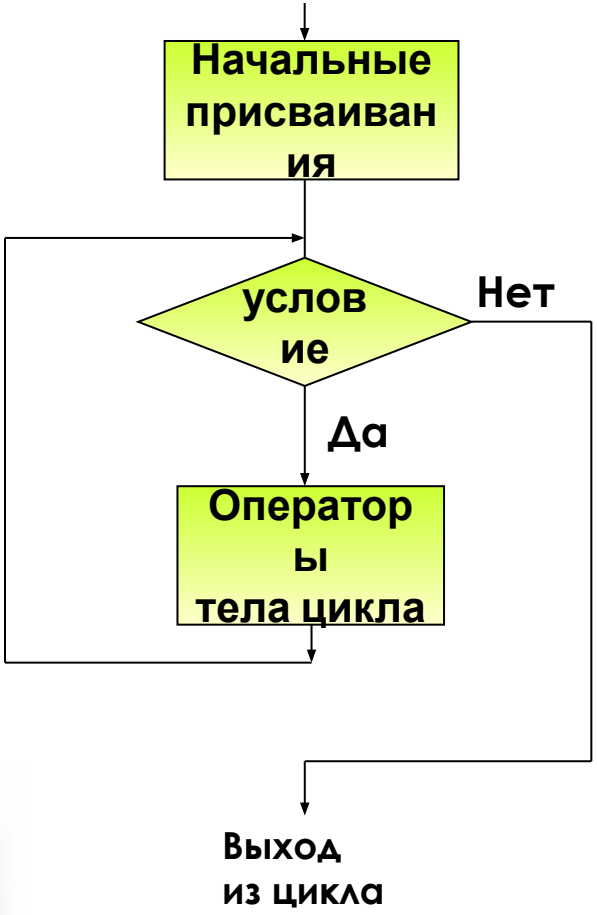
Программирование на Pascal



Цикл с предусловием (While)

Цикл с предусловием

```
while <условие> do begin{тело цикла}end;
```



While <условие> **do**

Begin

<Действия>

End;

Порядок выполнения цикла с предусловием оператора *While*

Синтаксис оператора:

While <условие> do <тело цикла>

<Условие> является логическим выражением. Это означает, что результатом вычисления этого выражения является одно из значений TRUE – истина или FALSE – ложь.
Тело цикла выполняется до тех пор, пока значением условия является TRUE.
Обратите внимание, что возможна ситуация, когда условие всегда будет истинным. В этом случае цикл будет выполняться бесконечно.
Если условие изначально ложно, тело цикла не выполнится ни разу, и управление сразу будет передано на оператор, следующий за телом цикла.

Пример:

```
Eps := 0.1;
i := 2;
While i>Eps do begin
i := i-i/2
end;
```

В данном примере производится расчет некоторой величины, пока получающееся значение не окажется меньше ранее заданной погрешности **Eps.**

WHILE (пока) <условие> DO (выполнять) BEGIN <тело цикла > END;

1. Сначала проверяется значение условия. Пока оно «истинно» выполняются операторы циклической части. Когда значение условия «ложно» происходит выход из цикла.
2. Можно в тело цикла поставить операторы перехода (EXIT и GOTO). Они передают управление за пределы цикла.
3. Этот цикл может работать как цикл с параметром, но при этом необходимо изменять переменную, являющуюся счетчиком цикла, с помощью оператора присваивания.
4. !!! Если при первой проверке условие «ложно», то цикл не выполнится ни разу.

Надо знать:

1. **Можно использовать сложные условия**

```
while (a < b) and (b < c) do begin
   {тело цикла}
end;
```

2. Если в теле цикла только один оператор, с[illegible]ь:

```
while a < b do
    a := a + 1;
```

Надо знать:

3. условие пересчитывается **каждый раз** при входе в цикл
4. если условие на входе в цикл ложно, цикл не выполняется ни разу

```
a := 4; b := 6;
while a > b do
   a := a - b;
```

5. если условие никогда не станет ложным, программа зациклится

```
a := 4; b := 6;
while a < b do
   d := a + b;
```

Сколько раз выполняется цикл?

```
a := 4; b := 6;
while a < b do a := a + 1;
```

2 раза
a = 6

```
a := 4; b := 6;
while a < b do a := a + b;
```

1 раз
a = 10

```
a := 4; b := 6;
while a > b do a := a + 1;
```

0 раз
a = 4

```
a := 4; b := 6;
while a < b do b := a - b;
```

1 раз
b = -2

```
a := 4; b := 6;
while a < b do a := a - 1;
```

зацикливание

Замена `for` на `while` и наоборот

```
for i:=1 to 10 do begin
  {тело цикла}
end;
```

```
i := 1;
while i <= 10 do begin
  {тело цикла}
  i := i + 1;  end;
```

```
for i:=a downto b do
  begin
  {тело цикла}
end;
```

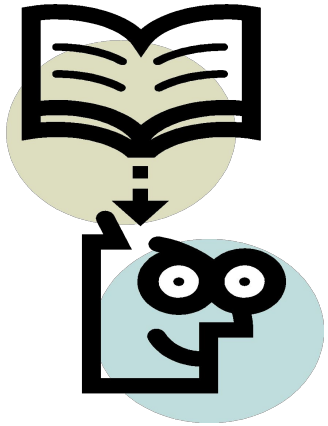
```
i := a;
while i >= b do begin
  {тело цикла}
  i := i - 1;  end;
```

Замена цикла `for` на `while` возможна всегда.

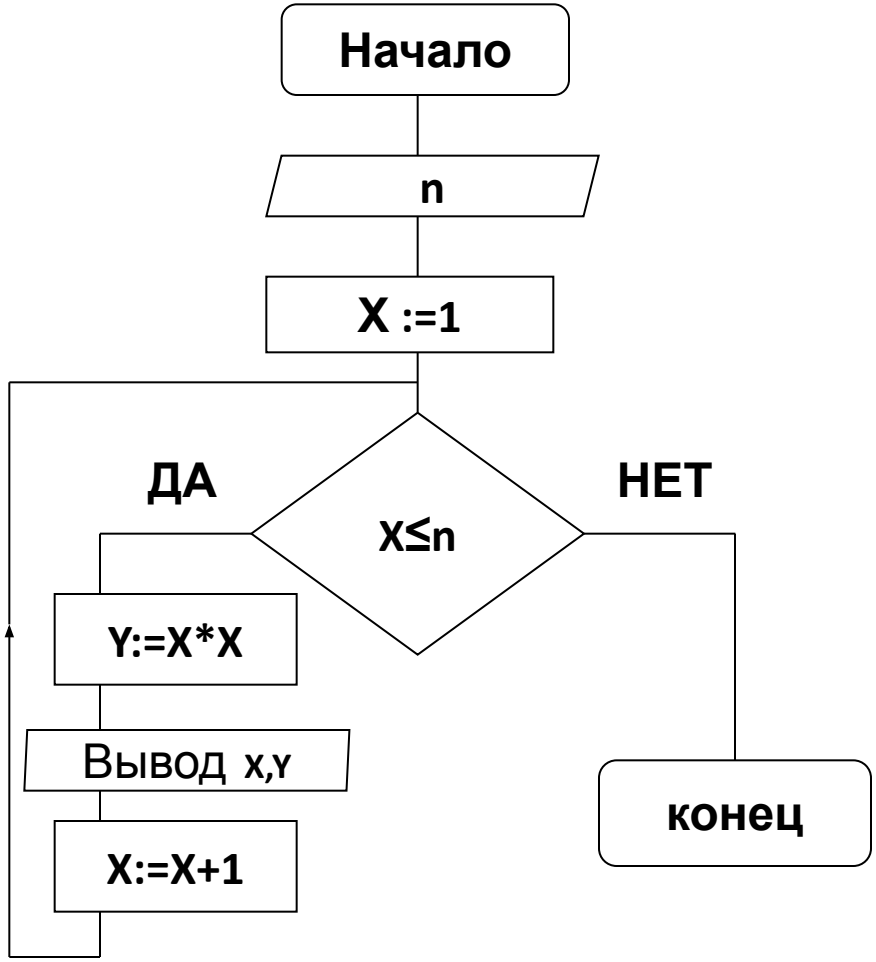
Замена `while` на `for` возможна только тогда, когда можно заранее рассчитать число шагов цикла.

Задача

- Вычислить значения функции $У = X^2$, для всех X от 1 до n с шагом 1.
- Решение



Решение:



- Паскаль

```
Program Task;
Uses Crt;
Var  x,y,n: Integer;
Begin
 ClrScr;
 Write('n=');ReadLn(n);
  x := 1;
  While x<=n do
   Begin
      y:= x *x;
      WriteLn ('x=',x,'  y=',y);
      x:= x+1;
    End; ReadLn;
End.
```

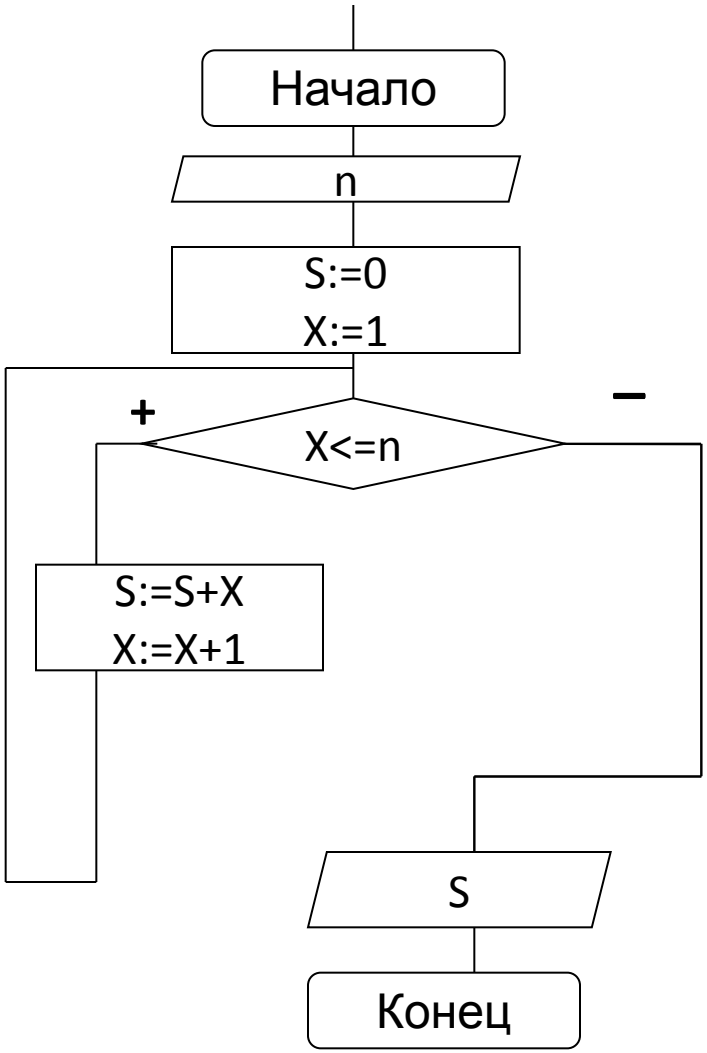
К условию

Задача

- Найти сумму чисел натурального ряда:
  $S=1+2+3+4+\ldots+n$
- Решение

Решение:

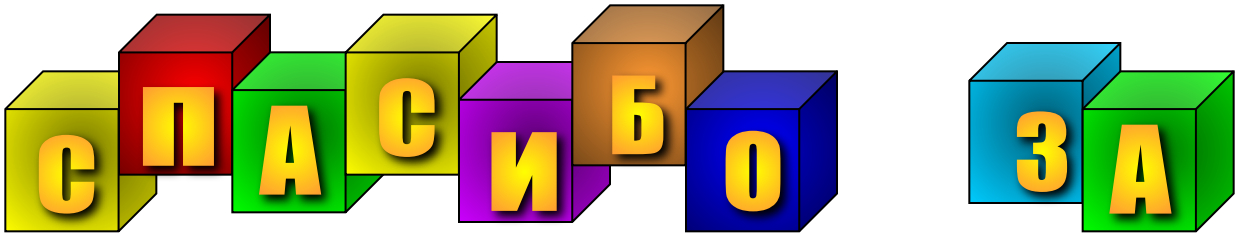
- Блок- схема



- Паскаль

```
Program Task;
Uses Crt;
Var     n, x, s : integer;
Begin
Write ('n='); ReadLn (n);
  S:=0;
  X:=1;
  While x<=n do
    Begin
      S:=S+X;
      X:=X+1;
    End;
  WriteLn ('Сумма=', S);
End;
```

К условию



УРОК