

При решении многих задач вычислительный процесс имеет циклический характер.

Это означает, что часть операторов многократно выполняется при различных значениях переменных.

Применение циклов в программе позволяет эффективно использовать компьютер, приводит к уменьшению длины программы и сокращению времени на ее составление и отладку.

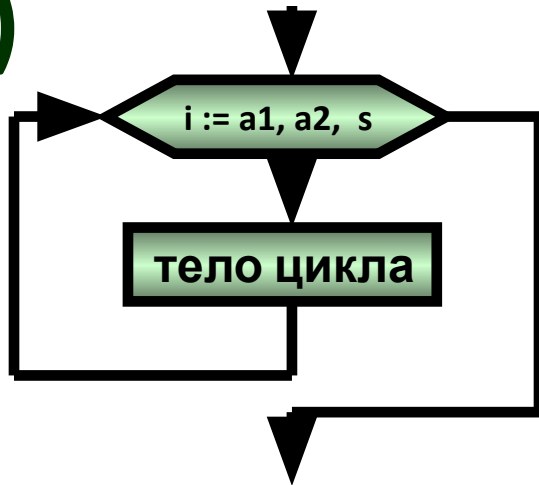
В языке Паскаль имеется три вида операторов цикла:

- Оператор цикла с параметром;
- Оператор цикла с предусловием;
- Оператор цикла с постусловием.

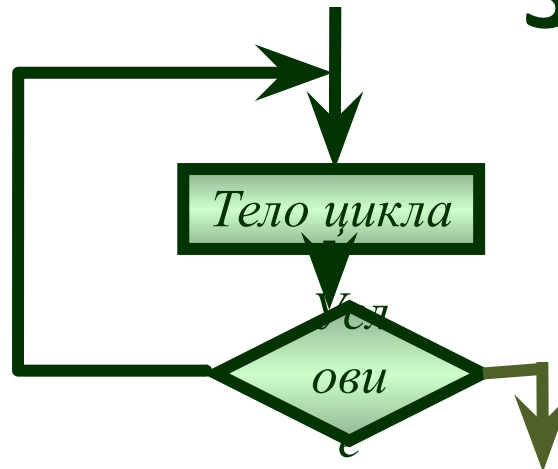
Операторы для записи циклов являются составными, так как в их состав входят другие операторы.

Привести в соответствие схемы и их обозначения.

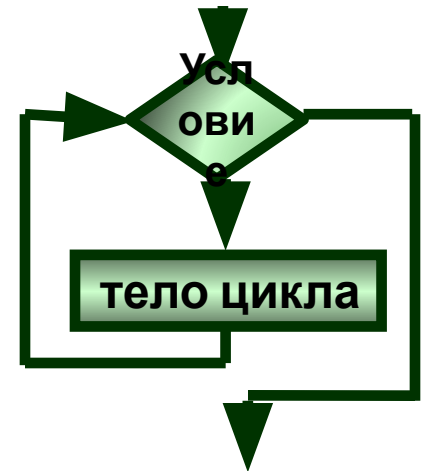
1)



2)



3)



а) Цикл с предусловием

б) Цикл со счетчиком

в) Цикл с постусловием

В языке программирования *Turbo Pascal* они выполняются с помощью разных операторов.

Для всех операторов цикла характерна следующая особенность:

1. Повторяющиеся вычисления записываются всего лишь один раз.
2. Вход в цикл возможен только через его начало.
3. Переменные оператора цикла должны быть определены до входа в циклическую часть.
4. Необходимо предусмотреть выход из цикла: или по естественному его окончанию, или по оператору перехода.

Если этого не предусмотреть, то циклические вычисления будут повторяться бесконечно. В этом случае говорят, что произошло «зацикливание» выполнения программы.

Оператор цикла с параметром

используется в тех случаях, когда заранее известно, сколько раз должна повториться циклическая часть программы.

Оператор цикла
имеет вид:

FOR I:=m1 TO m2 DO

Begin

m1<m2

Операторы циклической
части программы;

End;

**FOR - для
TO - до
DO - выполнить**

Возможен следующий
вид:

FOR I:=m1 DOWNTO m2 DO

Begin

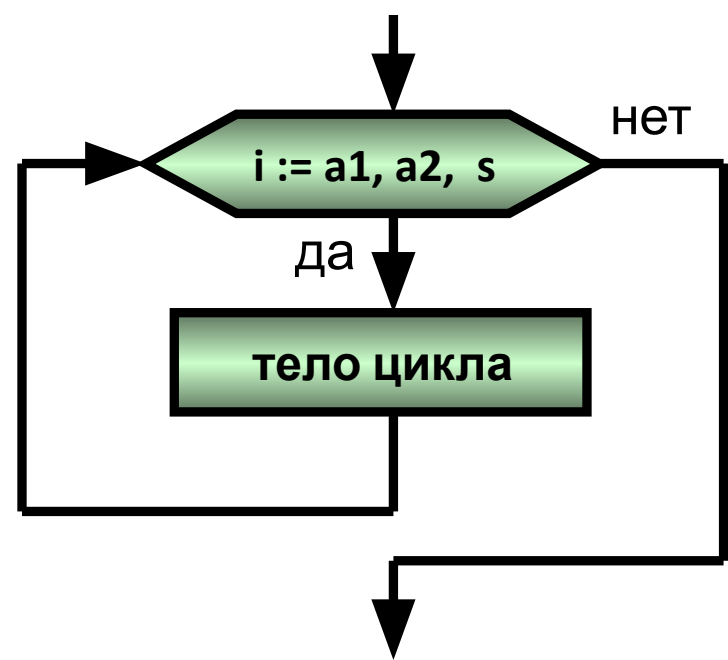
m1>m2

Операторы циклической
части программы;

End;

I – параметр цикла,
m1,m2 – начальное и конечное
значения параметра цикла

**Цикл со
счетчиком
(с известным
числом
повторений)**



```
For i:= a1 to a2 do  
begin  
    тело цикла  
end;
```

**Шаг изменения
счетчика +1**

i - параметр цикла,
выполняет роль счетчика;
 $a1$ - начальное значение
счетчика;
 $a2$ - конечное значение
счетчика.

Назад

Найти сумму 4 произвольных чисел.

...

S := 0;

For i := 1 to 4 do

Begin

Write('x=');

Readln(x);

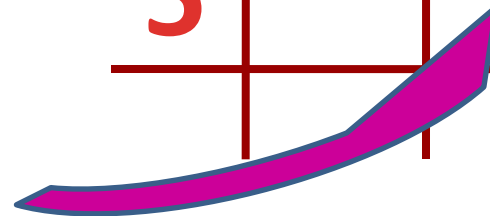
s := s + x;

End;

...

i	x	s	?
1	7	0	$1 \leq 4$
2	13	20	$2 \leq 4$
3	41	61	$3 \leq 4$
4	8	69	$4 \leq 4$
5			$5 \leq 4$

Ответ



ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ КОМАНД

ОПЕРАТОРА *FOR ...TO ...DO...*

1. Определяются значения переменных $a1$ и $a2$;
2. Переменной i присваивается значение переменной $a1$;
3. Проверяется условие выполнения тела цикла $i \leq a2$;
4. Если это условие истинно, то выполняется тело цикла;
5. Увеличивается значение переменной i на 1;

В операторе *For ...* значение счетчика может не только увеличиваться, но и уменьшаться.

В этом случае оператор выглядит так:

```
For i:= a2 downto a1 do  
begin  
  тело цикла  
end;  
end;
```

Шаг изменения
счетчика -1

Условие выполнения тела цикла: $i \geq a1$.

Тело цикла будет выполнено $a2 - a1 + 1$ раз.

Задача 1. Написать программу, которая вычисляет сумму положительных из n введенных с клавиатуры целых чисел.

```
Program summa;  
Var i, n, x, s: integer;  
Begin  
  Writeln ('ввести количество чисел');  
  Readln(n);  
  s := 0;  
  For i := 1 to n do  
    Begin  
      Writeln ('ввести число');  
      Readln(x);  
      If x > 0 then s := s + x;  
    End;  
  Writeln ('сумма >0 =', s);  
  Readln;  
End
```

- ✓ Начальное значение суммы должно быть
- ✓ Переменная-счетчик должна быть целого типа,
- ✓ Не рекомендуется в теле цикла изменять значение переменной-счетчика.

Задача 2. Написать программу, выводящую на экран степени числа 2 (от 0 до 10) в виде

Program Степени2.p;.

Var i, x: integer;

Begin

Writeln ('Tabliza');

Writeln ('-----');

x := 1;

For i := 0 to 10 do

Begin

Writeln ('2^', i:2, '= ', x:5);

*x := x*2;*

Writeln;

End;

Readln;

End.

```
C:\ Borland Pascal
Borland Pascal
Tabliza
-----
2^ 0= 1
2^ 1= 2
2^ 2= 4
2^ 3= 8
2^ 4= 16
2^ 5= 32
2^ 6= 64
2^ 7= 128
2^ 8= 256
2^ 9= 512
2^10=1024
```

Вопросы:

1. Сколько раз будет выполнено тело цикла?

1. for i:= -5 to 5 do тело цикла;

2. for i:= 6 to 2 do тело цикла;

3. for i:= 3 to 3 do тело цикла;

4. for i:= 3 downto 4 do тело цикла;

5. for i:= 3 downto 1 do тело цикла;

2. Какого типа должны быть параметр цикла, его начальное и конечное значения?

3. Можно ли изменять конечное значение переменной-счетчика в теле цикла?

Верно:

1. 11 раз

2. 0 раз

3. 1 раз

4. 0 раз

5. 3 раза

Вопрос

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

1. Можно ли в теле цикла использовать цикл?

Такой цикл

```
...  
For i := 1 to 4 do  
  Begin  
    For j := 1 to 4 do  
      Write( i*j, ' ');  
    Writeln;  
  End;  
...
```

2. Что до результата фрагмента программы в этом случае?

3. Какой цикл является внешним, а какой – внутренним?

Сколько раз выполняется цикл?

```
a := 1;  
for i := 1 to 3 do a := a + 1;
```

a = 4

```
a := 1;  
for i := 3 to 1 do a := a + 1;
```

a = 1

```
a := 1;  
for i := 1 downto 3 do a := a + 1;
```

a = 1

```
a := 1;  
for i := 3 downto 1 do a := a + 1;
```

a = 4

Задача: Вычислить $S=1+2+3+ \dots$
 $+n$

```
Program zadacha;  
  var n,s,i:integer;  
Begin  
  cls;  
  write('введите натуральное число  
n=');  
  readln(n);  
  s:=0;  
  for i:=1 to n do s:=s+i;  
  writeln('сумма=',s)  
End.
```

Оператор цикла с предусловием

используется в тех случаях, когда заранее неизвестно число повторений цикла.

Форма записи оператора цикла с предусловием:

WHILE логическое выражение **DO**

Begin

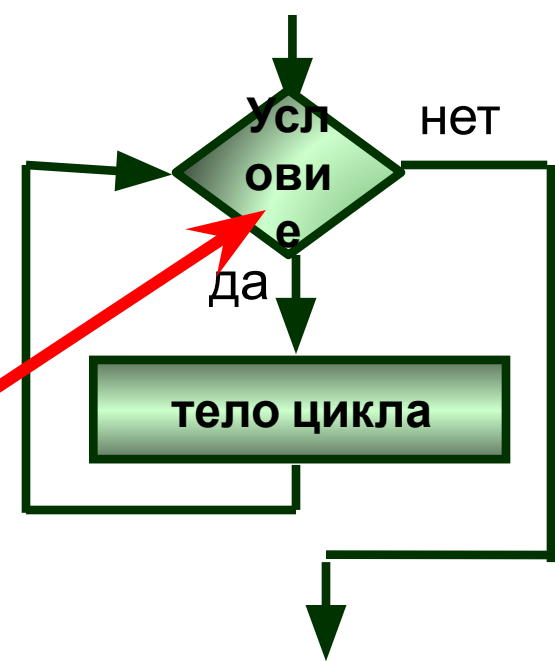
операторы циклической части программы;

End;

WHILE - пока
DO - выполнить

Оператор выполняется до тех пор пока логическое выражение (условие) имеет значение истина, прекращает выполняться, если логическое выражение принимает значение ложь.

Цикл с предусловием



...

While условие **do**

Begin

тело цикла

End;

...

Тело цикла выполняется, пока условие **ИСТИННО**.

Чтобы не было закливания, переменные, входящие в условие, должны изменяться в теле цикла.

Задача 3. Найти количество цифр в введенном с клавиатуры числе ($n > 0$).

```
Write('n=');
```

```
Readln(n);
```

```
m := n; k := 0;
```

```
While m > 0 do
```

```
Begin
```

```
    m := m div 10;
```

```
    k := k + 1;
```

```
End;
```

```
Writeln ('количество цифр в числе', n, '=', k);
```

```
Readln;
```

```
End.
```

n	m	k
5372	537	0
	2	

Ответ:

Задача: Известна сумма S_1 , положенная в банк и годовая процентная ставка P .

Через сколько лет накопится сумма S_2 ?

```
program bank;  
var  
  s1,s2,p:real;  
  g:integer;  
begin  
  write('Введите начальную сумму '); readln(s1);  
  write('Введите конечную сумму '); readln(s2);  
  write('Введите процентную ставку '); readln(p);  
  g:=0;  
  while s1<s2 do  
  begin  
    s1:=s1 + s1 * p / 100;  
    g:=g + 1;  
  end;  
  writeln('До накопления суммы пройдёт ',g,' лет');  
end.
```

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 a := a - 1;
```

зацикливание

Цикл с последующим условием используется в тех случаях, когда заранее неизвестно число повторений цикла.

Оператор цикла с постусловием имеет вид:

REPEAT

Операторы циклической части программы

Операторы циклической части выполняются повторно (по крайней мере 1 раз), до тех пор, пока значение логического выражения ложно.

Условием прекращения циклических вычислений является истинное значение логического выражения.

REPEAT –
повторить

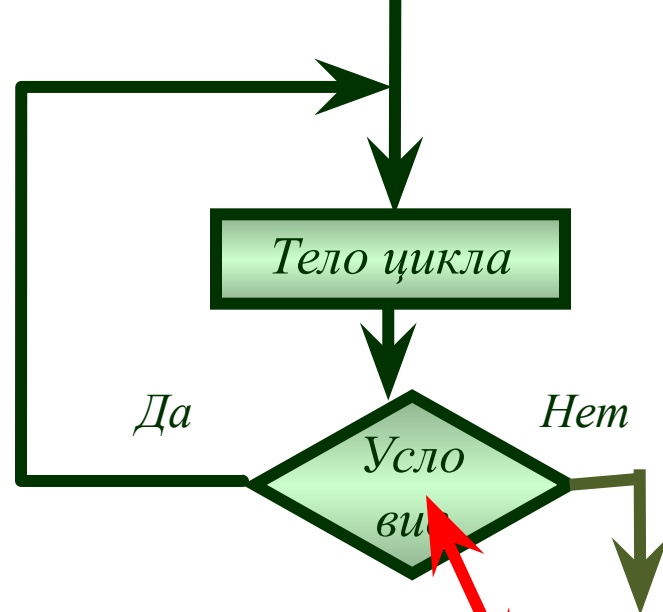
UNTIL – до тех пор

ИТАК, сначала выполняется циклическая часть, а затем проверяется условие.

Цикл

с

постусловием



...

Repeat тело цикла until условие;

...

Тело цикла выполняется пока условие ЛОЖНО.

Задача 4. Найти количество цифр в введенном с клавиатуры числе ($n > 0$).

Решим эту задачу с оператором Repeat

```
...  
...  
Write('n=');  
Readln(n);  
    m := n; k := 0;  
  
Repeat  
    m := m div 10;  
    k := k + 1;  
Until m = 0;  
WriteLn ('количество цифр в числе', n, '= ', k);  
Readln;  
End.
```

1. Операторные скобки (begin ... end) здесь не нужны;

2. Тело цикла выполнится хотя бы раз независимо от условия.

Сколько раз выполняется цикл?

```
a := 4; b := 6;  
repeat a := a + 1 until a > b;
```

3 раза
a = 7

```
a := 4; b := 6;  
repeat a := a + b until a > b;
```

1 раз
a = 10

```
a := 4; b := 6;  
repeat a := a + b until a < b;
```

зацикливание

```
a := 4; b := 6;  
repeat b := a - b until a < b;
```

2 раза
b = 6

```
a := 4; b := 6;  
repeat a := a + 2 until a < b;
```

зацикливание

Алгоритм Евклида

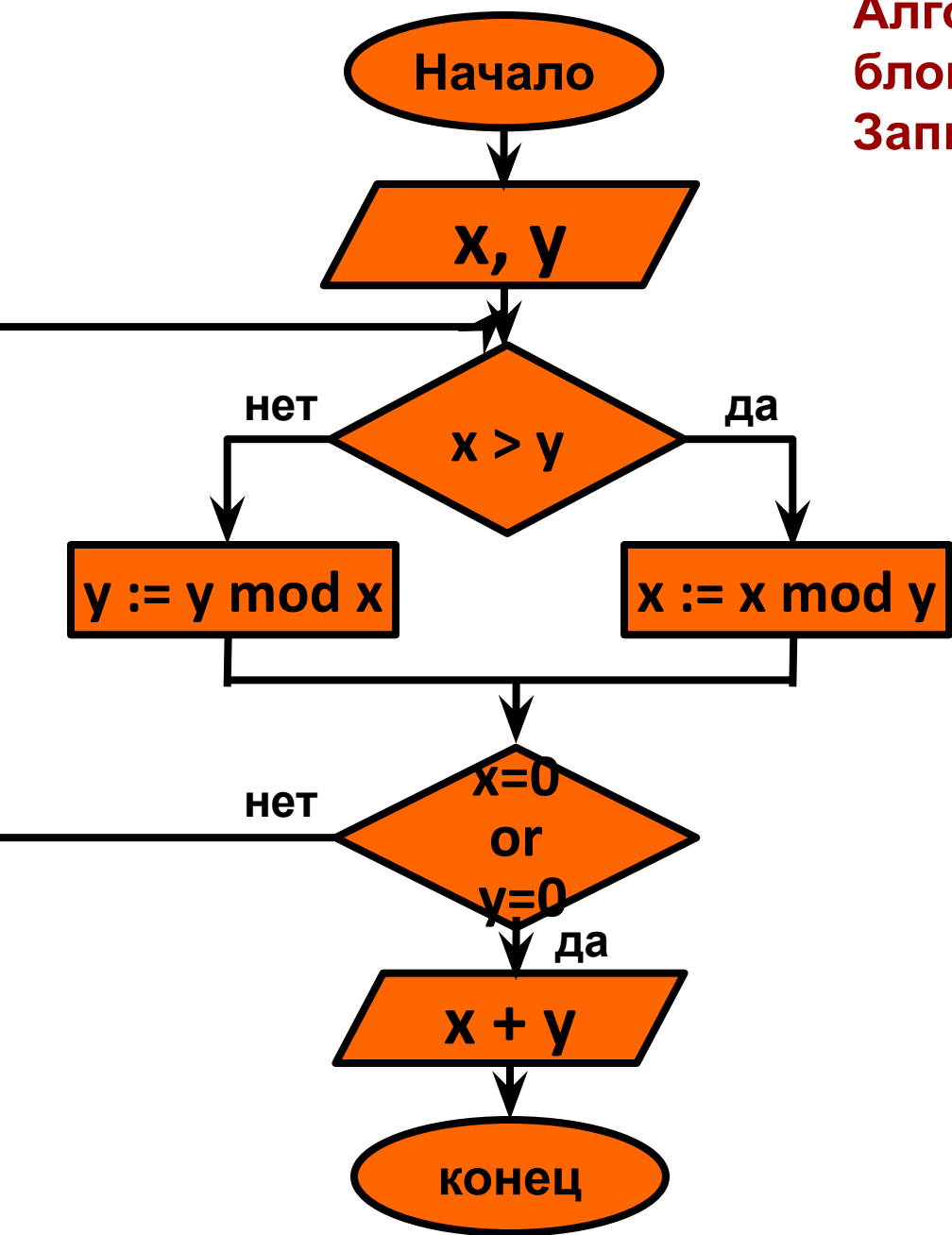
Алгоритм Евклида – это алгоритм нахождения наибольшего общего делителя (НОД) двух целых неотрицательных чисел.

Пусть x и y одновременно не равные нулю целые неотрицательные числа и пусть $x \geq y$.
Если $y = 0$, то $\text{НОД}(x, y) = x$, а если $y \neq 0$,
то для чисел x, y , и r , где r – остаток от деления x на y , выполняется равенство
 $\text{НОД}(x, y) = \text{НОД}(y, r)$.

Например, пусть $x = 48$, а $y = 18$.

$\text{НОД}(48, 18) = \text{НОД}(18, 12) = \text{НОД}(12, 6) =$

Алгоритм Евклида изображен
блок-схемой «цикл с постусловием»
Запишите его на языке Turbo Pascal.



```
Program NOD;
```

```
Var x, y : integer;
```

```
Begin
```

```
Write('vvod x, y');
```

```
Readln (x, y);
```

```
Repeat
```

```
if x > y then x := x mod y
```

```
else y := y mod x
```

```
Until (x = 0) or (y = 0);
```

```
Writeln ('NOD=', x + y);
```

```
Readln;
```

```
End.
```

Пример 1: Распечатать числа Фибоначчи от 1 до n.
(Каждое число в последовательности, начиная с третьего, получается сложением двух предыдущих чисел. Например, 1,1,2,3,5,8,13,21,...).

```
program primer_1;  
  var a,b,c,n,i:integer;  
begin  
  cls;  
  writeln('ВВЕДИТЕ ЧИСЛО n');  
  readln(n);  
  a:=1; b:=1;  
  writeln (a,b);  
  i:=3;  
  repeat  
    c:=a+b;  
    writeln(c);  
    a:=b; b:=c;  
    i:=i+1  
  until i>n  
end.
```

Пример 2: Написать программу нахождения наибольшего общего делителя (НОД) двух натуральных чисел.

```
program prim_2;
var
  x,y:integer;
begin
  cls;
  writeln('Введите x и y '); readln(x,y);
  repeat
    if x>y then x:=x mod y
    else y:=y mod x
  until (x=0) or (y=0);
  writeln('НОД=',x+y);
end.
```

Задача: Ввести целое число и определить число цифр в нем.

Идея решения: Отсекаем последовательно последнюю цифру, увеличиваем счетчик.

n	count
123	0
12	1
1	2
0	3

Проблема: Неизвестно, сколько шагов надо сделать.

Решение: Надо остановиться, когда $n = 0$, т.е. надо делать «до тех пор, пока не $n \geq 0$ ».

