

# ЦИКЛЫ В PASCAL

## ABC NET

# Циклические алгоритмы

- *Цикл* представляет собой последовательность операторов, которая выполняется *неоднократно*.
- В языке программирования Pascal имеется три разновидности цикла:
  1. Цикл со счётчиком (**For**)
  2. Цикл с предусловием (**while**)
  3. Цикл с постусловием (**repeat**)

# О циклах!!

- Подавляющее большинство задач с циклами можно решить разными способами, используя при этом любой из трёх операторов цикла;
- В некоторых случаях предпочтительнее использовать какой-то один из операторов;
- Самым универсальным из всех операторов цикла считается *while*, поэтому в случае затруднений с выбором можно отдать предпочтение ему;
- Цикл *for* обеспечивает удобную запись циклов с *заранее известным числом повторений*;
- При неумелом использовании циклов любого типа возможна ситуация, когда компьютер не сможет нормально закончить цикл ( в таком случае говорят, что программа «зациклилась»). При работе в среде Pascal ABC NET для выхода из подобной ситуации используется комбинация клавиш <Ctrl>+<Break>.

# ЦИКЛ СО СЧЕТЧИКОМ

X – управляющая переменная

A – начальное значение

B – конечное значение

Если  $A < B$ :

для **X** от A до B делай оператор

**For** X:=A **To** B **Do** оператор;

Если  $A > B$ :

**For** X:=A **DownTo** B **Do** оператор;

# Оператор FOR

- Этот вид оператора цикла называют **циклом со счётчиком** или **цикл с параметром**. В нём важную роль играет *переменная-параметр*, которая на каждом шаге цикла автоматически изменяет своё значение ровно *на единицу* – поэтому её и называют *счётчиком*.

- Инструкцию **for** можно реализовать двумя способами:

## Вариант 1

**For** счётчик:=НачальноеЗначение **to** КонечноеЗначение **do**

**Begin**

{инструкции}

**End;**

Ключевые слова **for**, **do** обозначают «для», «выполняй» соответственно. Строка, содержащая **for . . . do**, называется *заголовком цикла*, оператор, стоящий после **do** образует его *тело*. Очень часто тело цикла – составной оператор. Если тело цикла представлено одиночным оператором, то **begin** и **end** не пишутся.

## Вариант 2

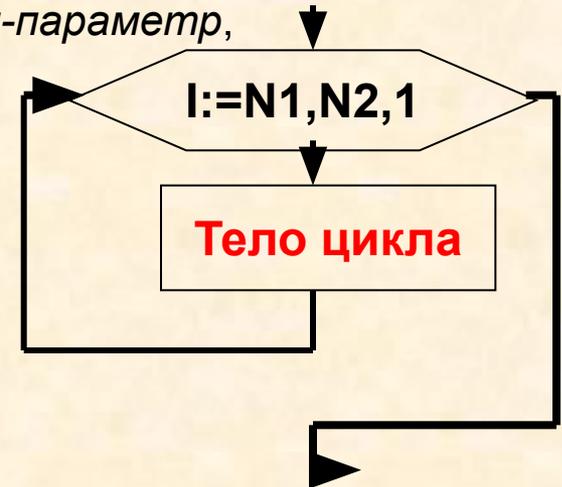
**For** Счётчик:=НачальноеЗначение **downto** КонечноеЗначение **do**

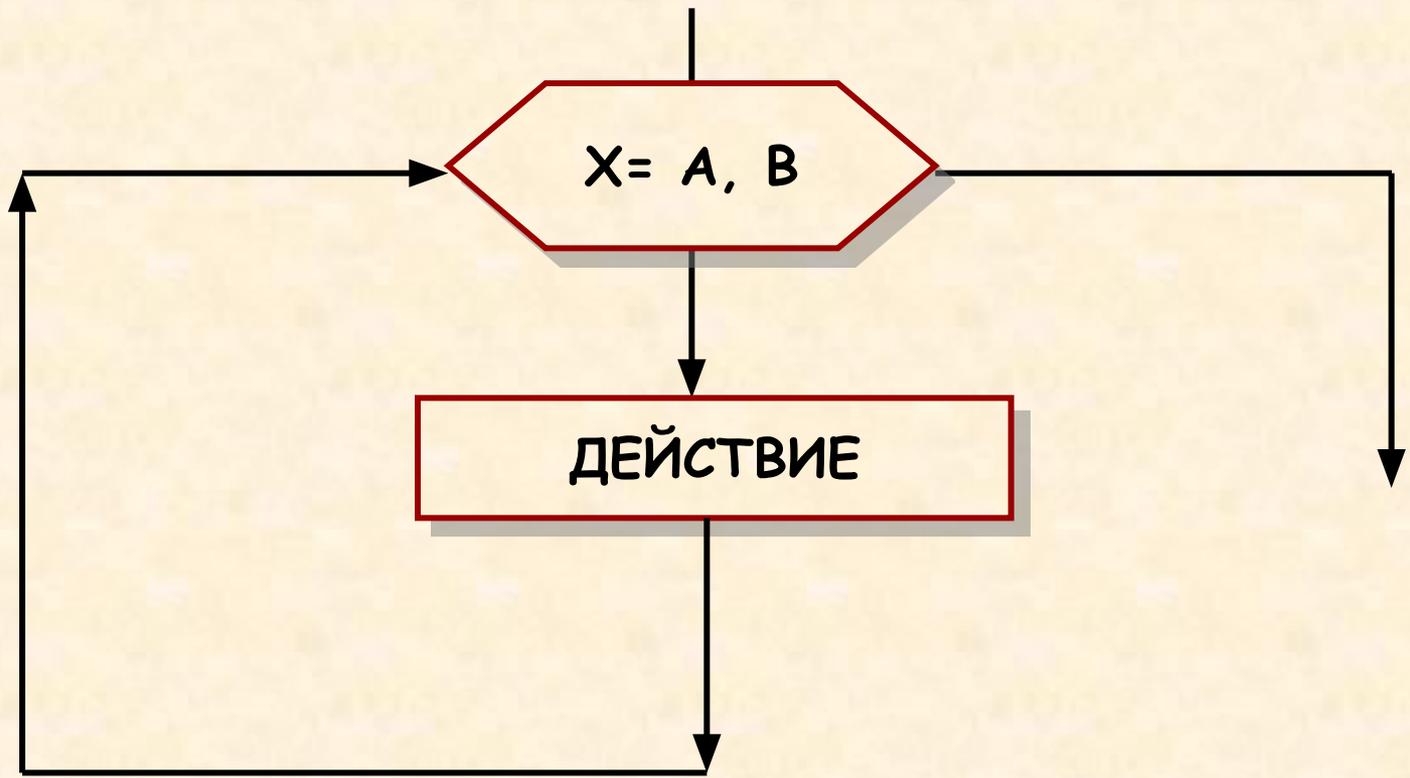
**Begin**

{операторы}

**End;**

Инструкции между **begin** и **end** выполняются столько раз, сколько определяет выражение  $[(\text{НачальноеЗначение} - \text{КонечноеЗначение}) + 1]$





# ПРОГРАММА:

**Задача 1: Надо вычислить сумму  $S=1+1/2+1/3+\dots+1/25$**

```
Program sum;
VAR S: REAL;
    I: INTEGER;
BEGIN
S:=0;
FOR I:=1 TO 25 DO
    S:=S+1/I;
WRITELN(' S=',S);
END.
```

**Задача 2: Необходимо возвести 2 в 3-ую степень**

```
Program stp;
VAR f, I: INTEGER;
BEGIN
f:=1;
for i:=1 to 3 do
begin
f:=f*2;
end;
WRITELN ('2^3 = ',F);
End.
```

# Задание 1

Составить программу, которая выводит на экран натуральные числа от **1** до **30**

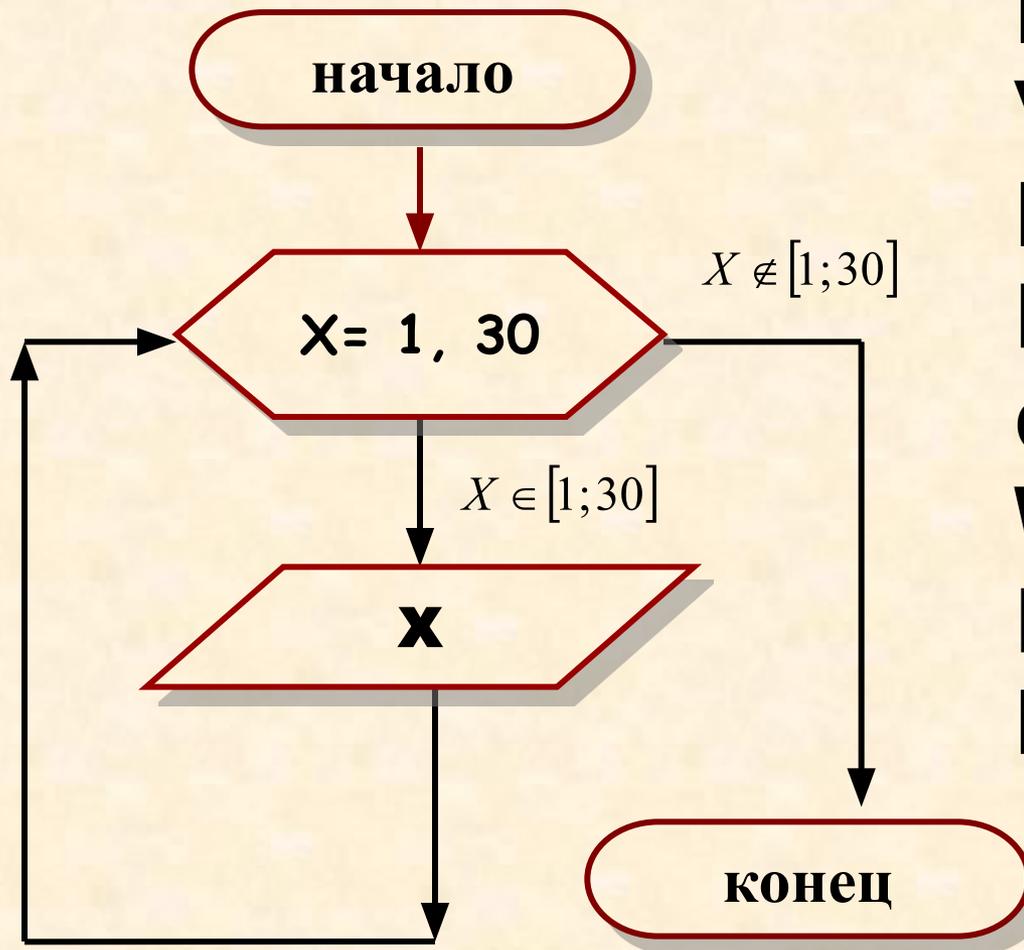
**1 2 3 4 5 ...30**

**Решение:**

X – управляющая переменная

**A = 1**

**B = 30**



```
Program z1;  
Var x:integer;  
Begin  
For x:=1 to 30  
do  
WriteLn(x);  
ReadLn;  
End.
```

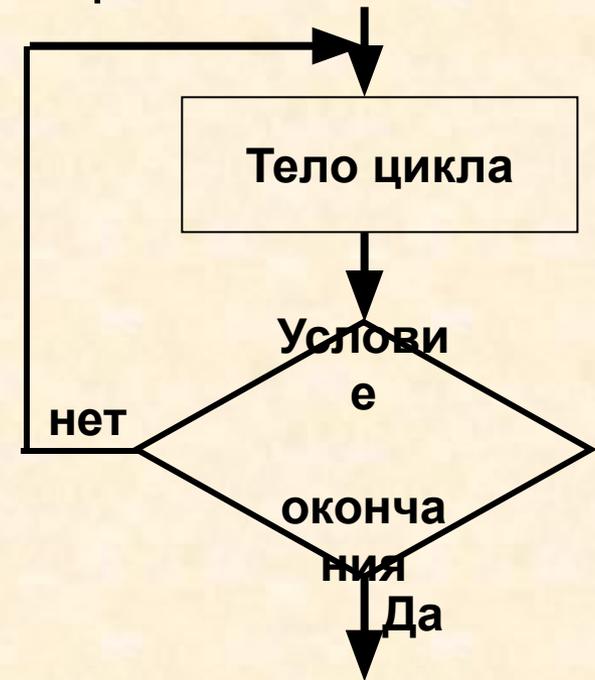
# Оператор REPEAT

- Оператор повтора `repeat` состоит из заголовка (`repeat`), тела и условия окончания (`until`). Ключевые слова `repeat`, `until` обозначают «повторяй» и «пока» соответственно.

## Repeat

{операторы}

- **Until** (условие выхода из цикла)  
число повторений заранее неизвестно и определяется в ходе программы;
- После слова **until** записывается условие **завершения** цикла
- **Условие** – это выражение логического типа: простое или сложное логическое выражение
- цикл **repeat** удобно использовать в тех случаях, когда тело цикла гарантированно должно *выполниться хотя бы один раз*;



# Программа:

Определить число  $n$ , при котором сумма квадратов натурального ряда чисел от 1 до  $n$  не превысит величину  $K$ , введенную с клавиатуры. Т. е.

$$S \geq K, \quad \text{где } S = \sum_{i=1}^n i^2 = 1^2 + 2^2 + \dots + n^2$$

**repeat**

**WriteLn('Введите  
положительное  
число');**

**ReadLn(x);**

**until x > 0;**

```
program sum_sq; {Сумма квадратов  
натурального ряда }
```

```
uses crt;
```

```
var k, s, n : integer;
```

```
begin
```

```
clrscr ;
```

```
writeln( 'Введите K' );
```

```
readln (k);
```

```
s:=0; n:=1;
```

```
repeat
```

```
    s :=s+n*n;
```

```
    n := n+1;
```

```
until s > k;
```

```
writeln ('N= ', n : 3, ' s= ', s : 5 );
```

```
readln;
```

```
end.
```

# Оператор WHILE

- Оператор повтора `while` состоит из *заголовка* и *тела цикла*. Ключевые слова *while* и *do* обозначают «до тех пор, пока» и «выполняй» соответственно

**While** Условие выполнения цикла **do**

{инструкции}

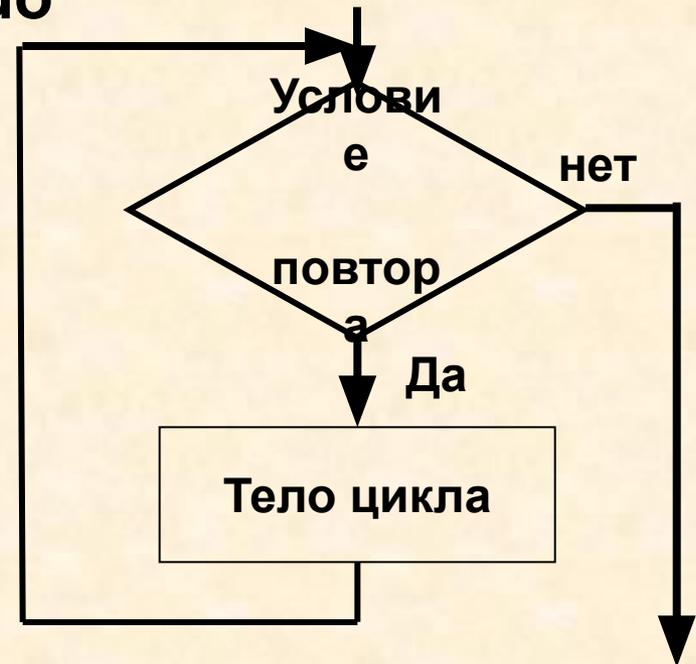
**End;**

□ **Число повторений** операторов (инструкций) цикла **while** определяется в ходе работы программы;

□ После слова **while** записывается условие **продолжения выполнения** инструкций цикла. В этом отличие цикла **while** от цикла **repeat**;

□ **условие** – это выражение логического типа, которое может принимать одно из двух значений: `true` или `false`;

□ Цикл **while** – это цикл с предусловием, т.е. инструкции тела цикла *вообще могут быть не выполнены*, если проверяемое условие ложно с самого начала;

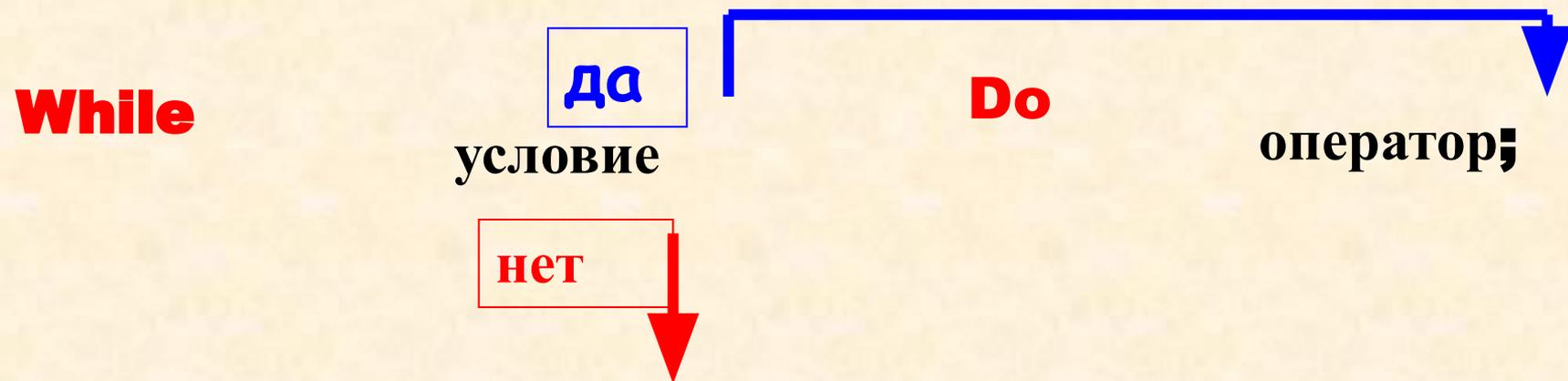


# Программа:

Задание 3: **Надо вычислить сумму  $S=1+1/2+1/3+\dots+1/30$**

```
Program sm;  
VAR S: REAL;  
    N:INTEGER;  
BEGIN  
    S:=0; N:=1;  
    WHILE N<=30 DO  
        BEGIN S:=S+1/N; N:=N+1; END;  
    WRITELN(' S=',S);  
    END.
```

# ЦИКЛ С ПРЕДУСЛОВИЕМ



# Задачи на циклы

- Например, выполнение цикла-фрагмента программы:

```
For i:=14 downto 10 do write(i:3);
```

Выведет на экран последовательность чисел в виде:

```
14 13 12 11 10
```

Если переменная-счётчик имеет символьный char тип, то оператор

```
For ch:='a' to 'e' do write (ch:2);
```

Выведет на экран последовательность букв в виде:

```
a b c d e
```

Оператор

```
For ch:='e' to 'a' downto write (ch:2);
```

Выведет на экран последовательность букв в виде:

```
e d c b a
```

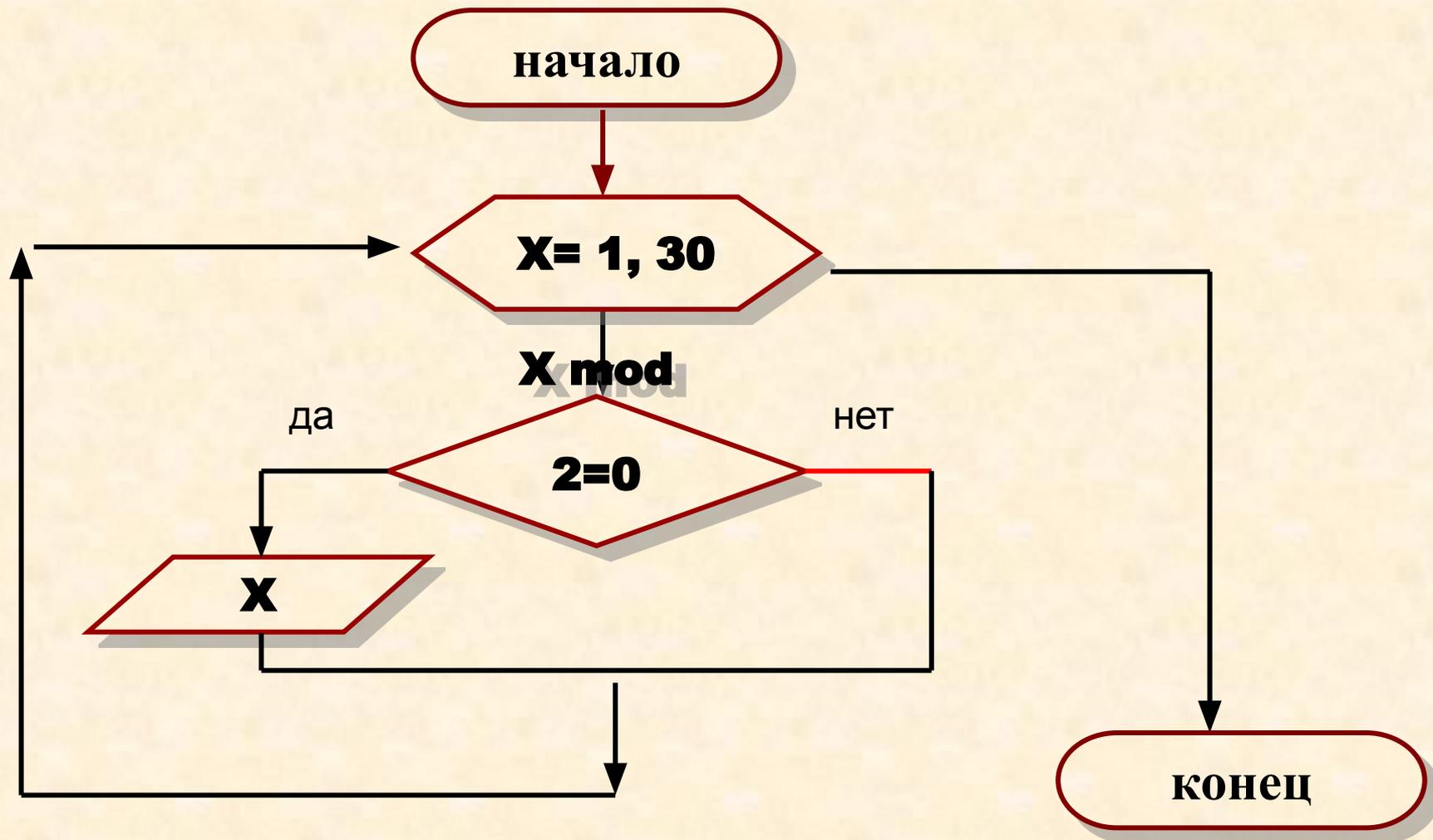
## Задание 3

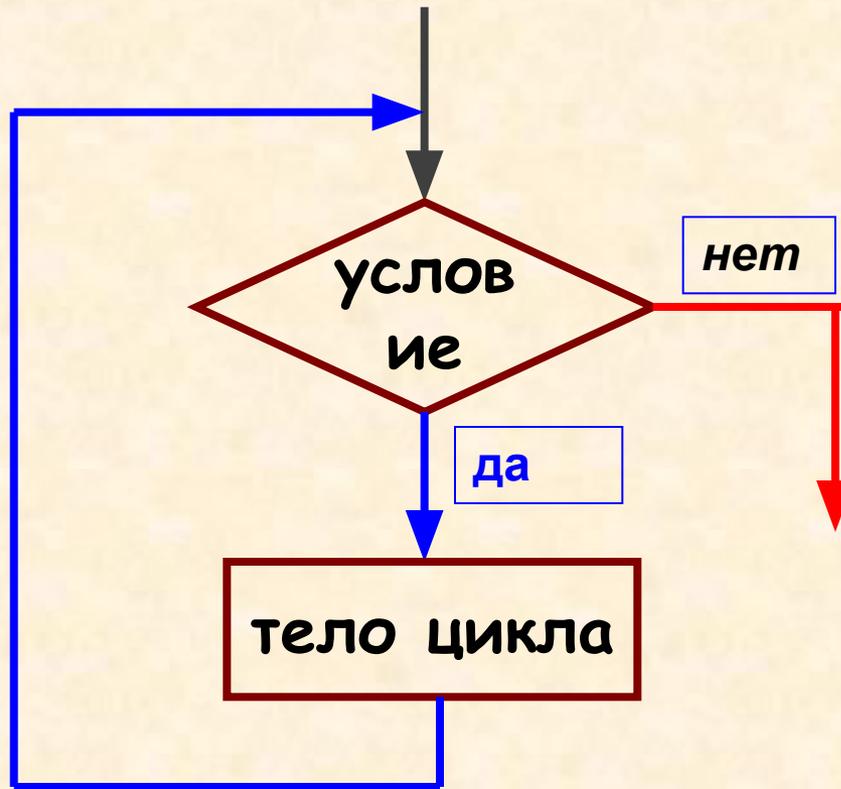
Составить программу, которая выводит на экран четные натуральные числа от **1** до **30**

Решение:

Условие четности числа –  
остаток от деления на **2 = 0**

**$X \bmod 2 = 0$**





## Задание 7

Составить программу, которая вычисляет значения функции  $y=x^2-1$ , где  $x$   $[-2 ; 2]$  с шагом **0,5**

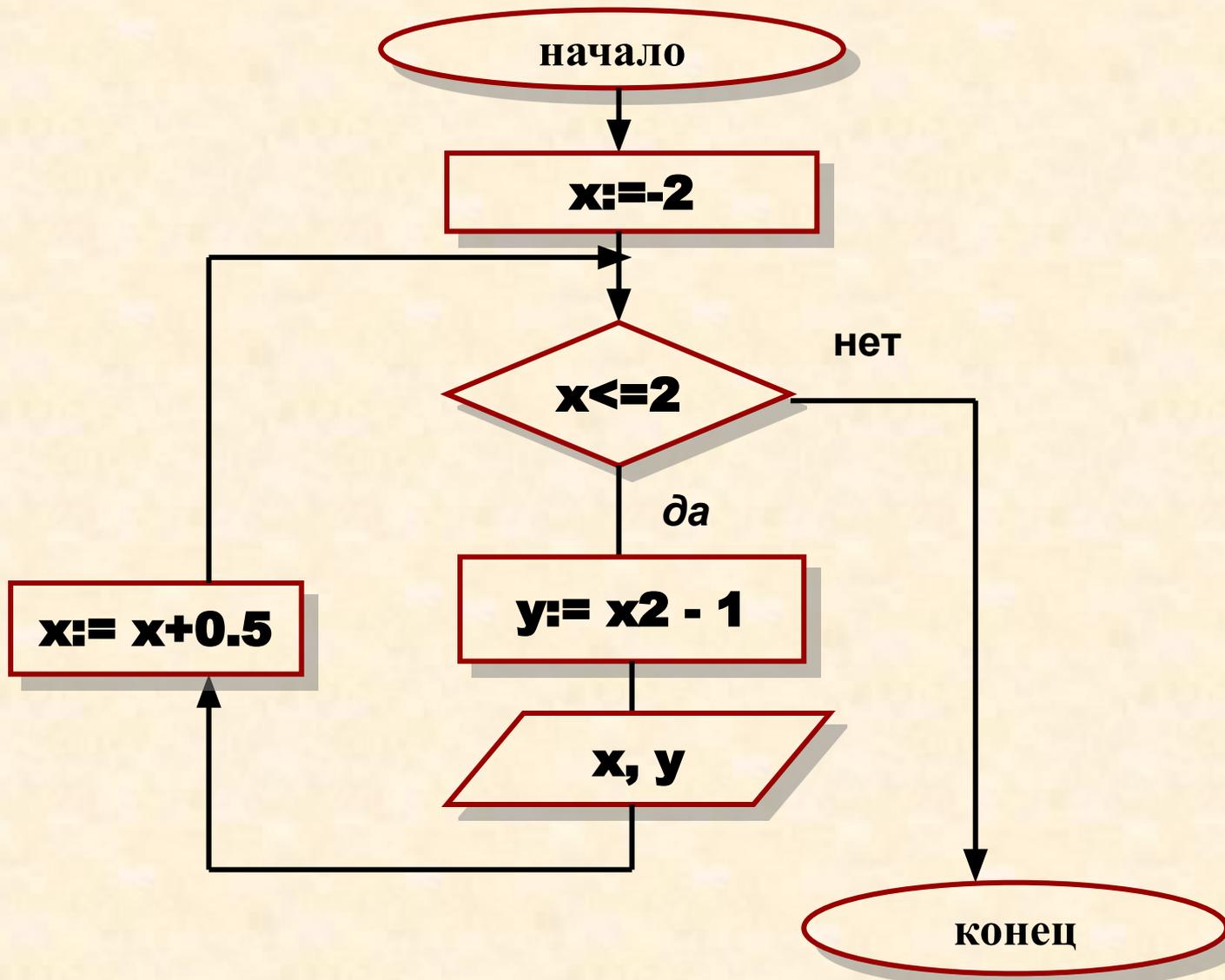
$$x = -2 \quad y=3$$

$$x = -1,5 \quad y= 1,25$$

$$x = -1 \quad y = 0$$

...

$$x=2 \quad y=3$$



```
Program z6;  
Var x,y:Real;  
Begin  
x:=-2;  
While x<=2 do  
  begin  
    y:=sqr(x)-1;  
    WriteLn('x=',x,' y=',y);  
    ReadLn;  
    x:=x+0.5  
  end;  
End.
```

# ЦИКЛ С ПОСТУСЛОВИЕМ

## Repeat

<оператор 1>;

<оператор 2>;

...

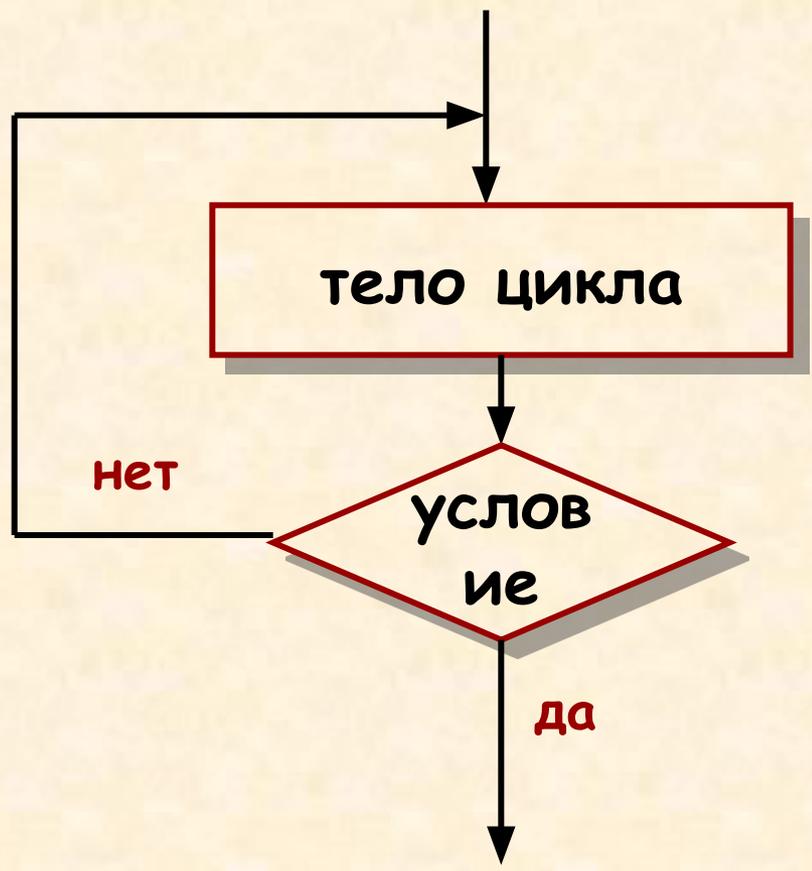
<оператор n>;

**Until** <условие>;

нет

да





**Спасибо за внимание!!!**