

# Программирование на языке Паскаль

# УРОК 1





**Никлаус Вирт  
(Niklaus Wirth)**



**Блез Паскаль**

В **1970** году в мире программирования произошли по крайней мере два великих события- появились операционная **система UNIX** и **новый язык программирования**, созданный профессором **Никлаусом Виртом** из швейцарского Федерального Технологического Института в Цюрихе . **Вирт** назвал его в честь великого французского математика и философа XVII века **Блеза Паскаля**.



**Язык Паскаль** удобен для  
начального обучения  
программированию, не только  
потому, что учит как правильно  
написать программу, но и тому,  
как правильно  
разрабатывать методы решения  
задач программирования



# Алфавит языка

- прописные и строчные буквы латинского алфавита от **A до z**, а также символ подчеркивания ( **\_** ), который тоже считается буквой.
- Прописные и строчные буквы взаимозаменяемы (равнозначны);
- арабские цифры **0 1 2 3 4 5 6 7 8 9**;
- специальные одиночные знаки: **+ - \* / = < > . , : ; ^ \$ # @**;
- специальные парные знаки: **[ ] ( ) { } ;**
- составные знаки : **< = > = < > .. ( \*\* ) ( .. )**.

# Структура программы

Программа на языке Паскаль состоит из:

*// Заголовок (необязательная часть)*

*// Блок описаний*

*// Тело программы*

# Блок описаний

- *Описание меток;*
- *Определение констант;*
- *Определение типов;*
- *Описание переменных;*
- *Описание процедур и функций.*

# Тело программы

- `begin` { начало программы }
- { тело программы }
- `end.` { конец программы }



# Структура программы

**Program** имя программы; {заголовок программы}

**Uses** ...; {раздел подключения модулей}

**Label** ...; {раздел описания меток}

**Const** ...; {раздел описания констант}

**Type** ...; {раздел определения типов}

**Var** ...; {раздел описания переменных}

**Function** ...; **Procedure** ...; {раздел описания функций и процедур}

**BEGIN**

... {раздел операторов}

**END.**

# Из чего состоит программа?

---

**Константа** – постоянная величина, имеющая имя.

**Переменная** – изменяющаяся величина, имеющая имя (ячейка памяти).

**Процедура** – вспомогательный алгоритм, описывающий некоторые действия (рисование окружности).

**Функция** – вспомогательный алгоритм для выполнения вычислений (вычисление квадратного корня, **sin**).

# Константы

**const**

**i2 = 45; { целое число }**

**pi = 3.14; { вещественное число }**

целая и дробная часть отделяются точкой

**q = 'Вася'; { строка символов }**

можно использовать русские буквы!

**L = True; { логическая величина }**

может принимать два значения:

- True (истина, «да»)
- False (ложь, «нет»)



# Переменные

---

## Типы переменных:

- integer { *целая* }
- real { *вещественная* }
- char { *один символ* }
- string { *символьная строка* }
- boolean { *логическая* }

## Объявление переменных (выделение памяти):

*variable* – переменная

тип – целые

```
var a, b, c: integer;
```

список имен  
переменных

# Пример программы

**Begin**

```
write('Это моя первая программа!');
```

**End.**

Запустите программу на исполнение  
просмотрите результат.

# Оператор вывода

**Write(‘текст’);** - оператор вывода текста и переменных на экран (оставляет курсор на текущей строке);

**WriteIn(‘текст’);** - оператор вывода текста и переменных на экран (переносит курсор на новую строку);



# Задание

- Выведите следующий текст на экран:

**Привет всем!**

**Я лучший программист НТСК!**

Обратите внимание на то, что текст  
выводится в две разные строчки

# УРОК 2

# Вычисление математических выражений в АВС Pascal



- **Приоритет выполнения операций в АВС Pascal такой же, как и в математике.**
- **Достаточно только научиться правильно записывать математические выражения на языке Pascal**

# Арифметические действия

<i>Операция</i>	<i>Наименование</i>
<b>A+B</b>	<b>Сложение</b>
<b>A-B</b>	<b>Вычитание</b>
<b>A*B</b>	<b>Умножение</b>
<b>A/B</b>	Деление (только для вещественного типа)
<b>A div B</b>	Вычисление целой части (неполного частного)
<b>A mod B</b>	Вычисление остатка

**Вычисли:**

**10 div 3**

**18 mod 4**

# Примеры

Математическая запись	Запись на языке Pascal
$37(25+87,5)-17(4,6+1,9)$	$37*(25+87.5)-17*(4.6+1.9)$
$\frac{ab}{c} + d^4$	$a*b/c+d*d*d*d$
$\frac{a+2b-3c}{5a+4}$	$(a+2*b-3*c)/(5*a+4)$
$\frac{a+b}{a-b} + \frac{ab}{3,14}$	$(a+b)/(a-b)+a*b/3.14$

Десятичная запятая в Pascal  
обозначается точкой



# Стандартные функции

Функция Паскаля	Математическая запись	Название
Abs(x)	$ X $	Абсолютная величина числа X (модуль)
Sqr(x)	$X^2$	Возведение числа в квадрат
Exp(x)	$e^x$	Экспонента
Sqrt (x)	$\sqrt{x}$	Вычисление квадратного корня
Frac (x)		Вычисляет дробную часть числа
Round (x)		Округляет до ближайшего целого числа
Trunc (x)		Отсекает дробную часть

Определите результат:

Frac(16.68); Round(16.68); Trunc(16.68);

# Стандартные функции

Функция Паскаля	Математическая запись	Название
Sin(x)	$\sin x$	Вычисление синуса
Cos(x)	$\cos x$	Вычисление косинуса
Int(x)	$[x]$	Целая часть числа
Ln(x)	$\ln x$	Вычисление натурального логарифма
Exp(y*ln(x))	$x^y$	Возведение числа в степень
Random(x)		Возвращает случайное число в диапазоне от 0 до X
Pi	$\pi$	Число $\pi$

$$\operatorname{tg} X = \frac{\sin x}{\cos x}$$

$$\operatorname{ctg} X = \frac{\cos x}{\sin x}$$

# Градусы в Pascal нужно перевести в радианы

$$30^\circ = 30 * \pi / 180$$

Математическая запись

Запись на языке Pascal



# Операторы ЯП Pascal ABC

# Оператор присваивания

**<имя переменной> := <выражение>;**

Арифметическое выражение может включать

- константы
- имена переменных
- знаки арифметических операций:

+   -   \*   /   div   mod

умножение

деление

деление  
нацело

остаток от  
деления

- вызовы функций
- круглые скобки ( )

# Какие операторы неправильные?

```
program qq;  
var a, b: integer;  
    x, y: real;  
begin  
    a := 5;  
    10 := x;  
    y := 7,8;  
    b := 2.5;  
    x := 2*(a + y);  
    a := b + x;  
end.
```

имя переменной должно  
быть слева от знака :=

целая и дробная часть  
отделяются **точкой**

нельзя записывать  
вещественное значение в  
целую переменную



# Оператор ввода

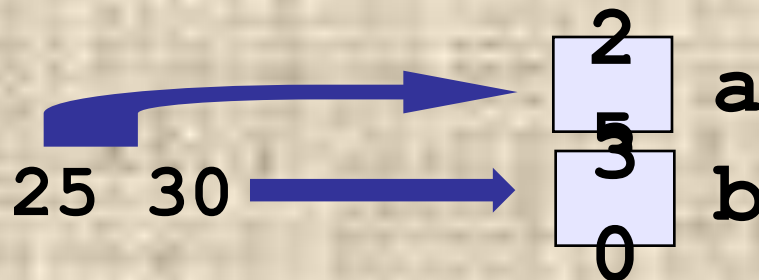
---

```
read ( a );      { ввод значения  
                  переменной a }
```

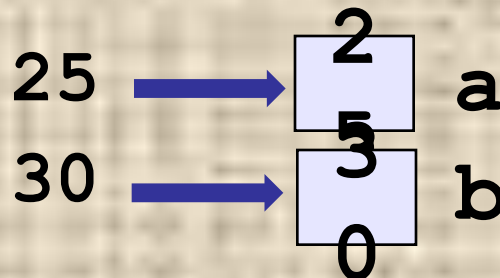
```
read ( a, b );  { ввод значений  
                  переменных a и b }
```

## Как вводить два числа?

через пробел:



через *Enter*:



# Оператор вывода

---

`write ( a );`      { вывод значения  
                          переменной `a` }

`writeln ( a );`    { вывод значения  
                          переменной `a` и переход  
                          на новую строку }

`writeln ( 'Привет!' );`    { вывод текста }

`writeln ( 'Ответ: ', c );`    { вывод  
                          текста и значения переменной `c` }

`writeln ( a, '+', b, '=', c );`

# Сложение двух чисел

---

**Задача.** Ввести два целых числа и вывести на экран их сумму.

**Простейшее решение:**

```
program qq;  
var a, b, c: integer;  
begin  
    read ( a, b );  
    c := a + b;  
    writeln ( c );  
end.
```

# Полное решение

```
program qq;  
var a, b, c: integer;  
begin  
    writeln('Введите два целых числа');  
    read ( a, b );  
    c := a + b;  
    writeln ( a, '+', b, '=', c );  
end.
```

Протокол:

компьютер

Введите два целых числа

25 30

пользователь

25+30=55



# УРОК 3

**ВЫЧИСЛИТЬ:**

$$12 \operatorname{div} 4 =$$

$$19 \operatorname{div} 5 =$$

$$12 \bmod 3 =$$

$$136 \bmod 10 =$$

**На Паскале**

**На языке математики**

**Sqrt(x)**

*Модуль числа  $x$*

**Abs(x)**

*Возведение в квадрат числа  $x$*

**Trunc(x)**

$e^x$

**Sqr(x)**

*Вычисление дробной части*

**Round(x)**

*Корень квадратный из  $x$*

**Frac(x)**

*Округляет до ближайшего целого числ*

**Exp(x)**

*Отсекает дробную часть*

**Random(x)**

*Синус  $x$*

**Sin(x)**

*Возвращает случайное число от 0 до  $x$*

**Exp(y\*ln(x))**

*Косинус  $x$*

**Cos(x)**

$[x]$

**Ln(x)**

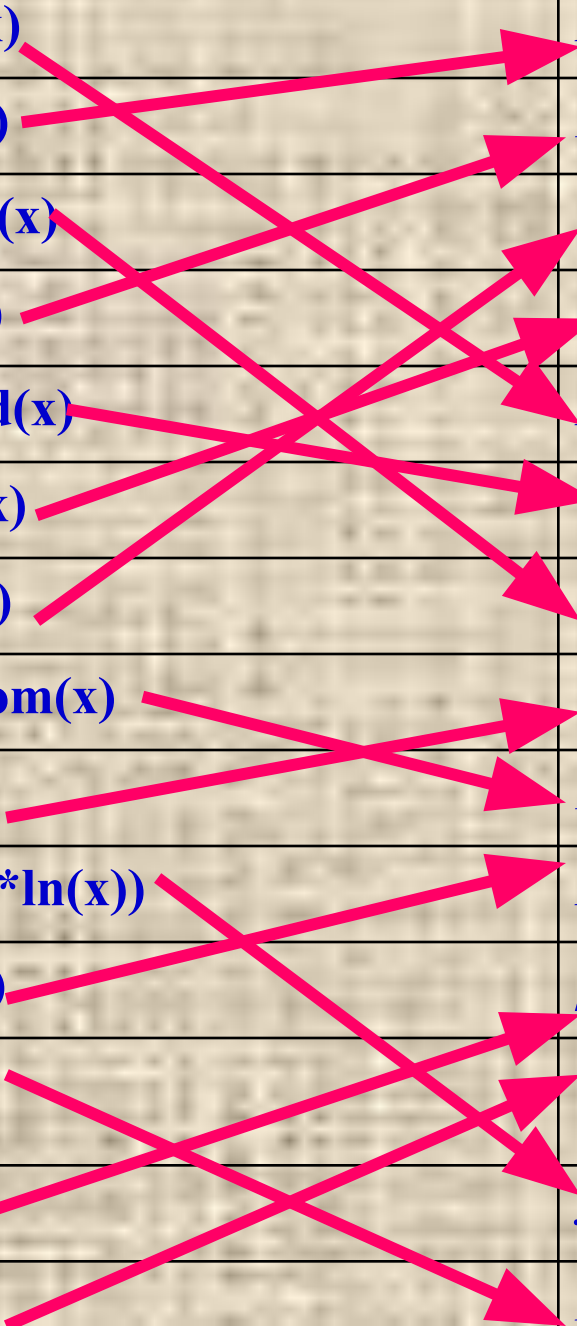
$\pi$

**Int(x)**

$x^y$

**Pi**

*Ln  $x$*



# *Домашнее задание*

**1. Вычислить длину окружности и площадь круга по заданному радиусу**

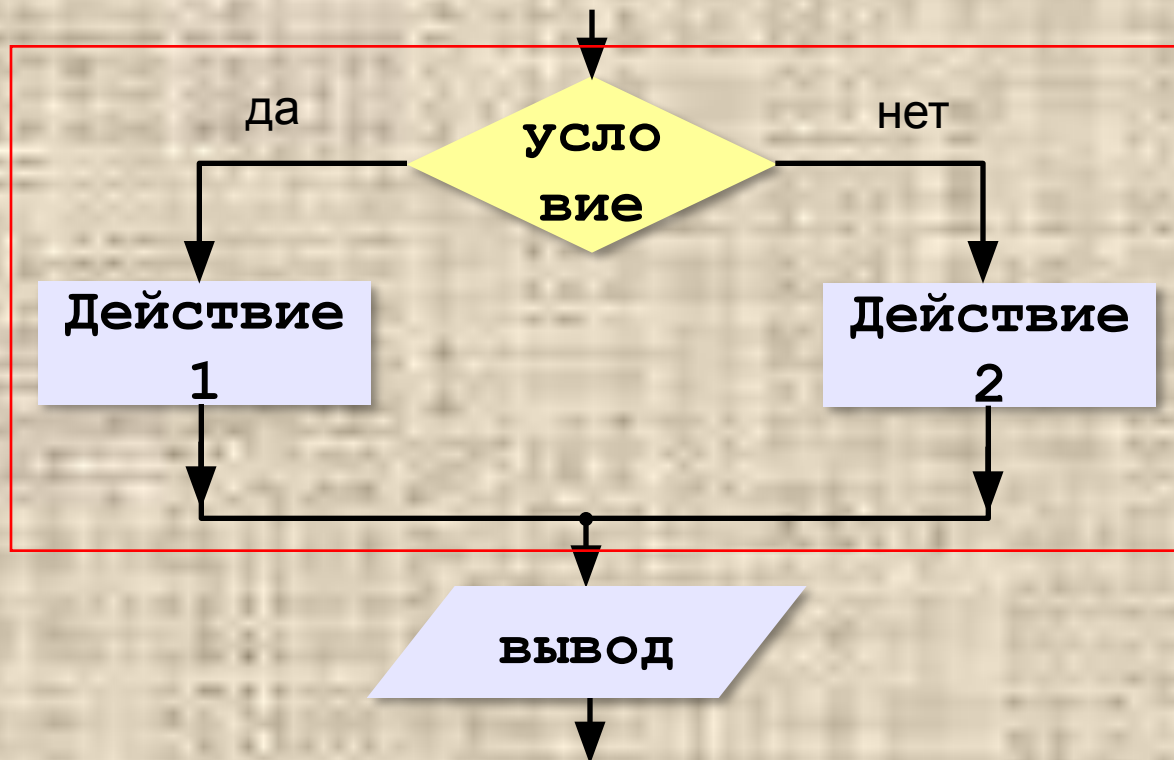
**2. Вычислить периметр прямоугольного треугольника по его катетам**



# УРОК 4

# Реализация разветвляющихся алгоритмов на Паскале

# Условный оператор (полная форма)



полная  
форма  
ветвления

# Условный оператор (полная форма)

---

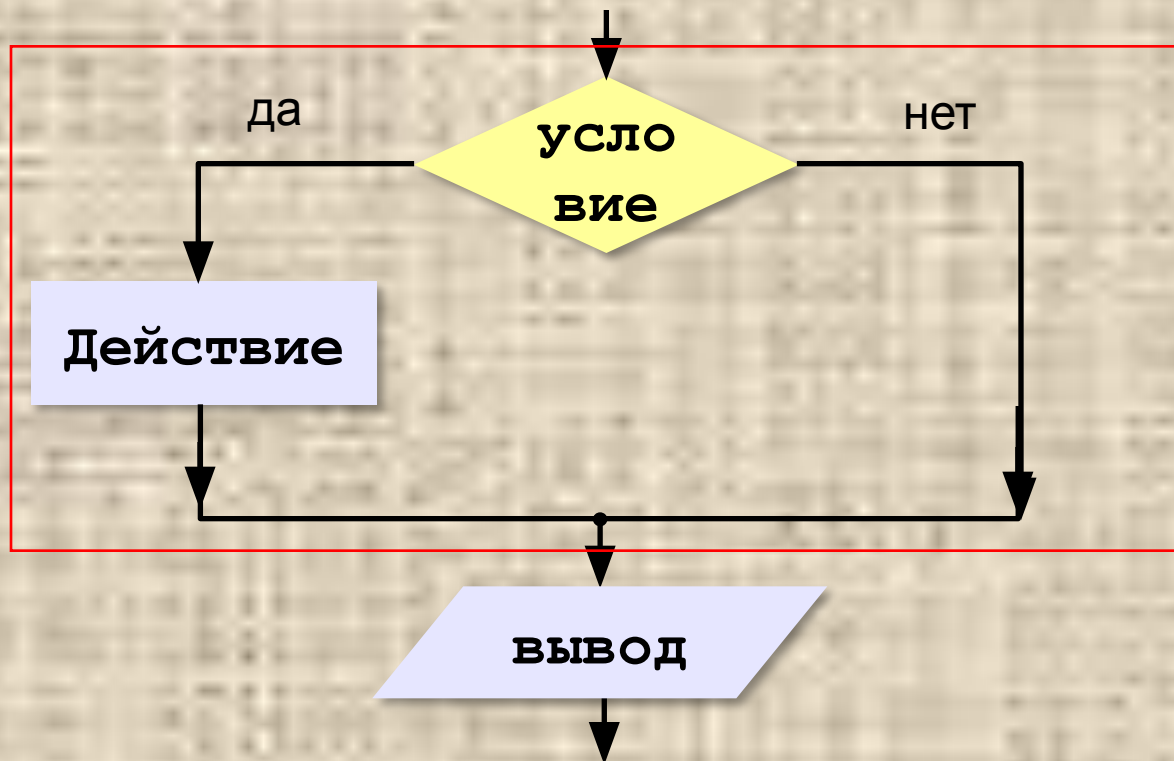
```
if <условие> then begin
    {что делать, если условие верно}
end
else begin
    {что делать, если условие неверно}
end;
```

## Особенности:

- перед ***else*** **НЕ** ставится точка с запятой
- если в блоке один оператор, можно убрать слова ***begin*** и ***end***



# Условный оператор (неполная форма)



неполная  
форма  
ветвления

# Условный оператор (неполная форма)

---

```
if <условие> then begin
    {что делать, если условие верно}
end;
```

## Особенности:

- если в блоке один оператор, можно убрать слова *begin* и *end*

---

**Вычислить значение функции:**

$$y = \begin{cases} x^2 + 5, & x > 3 \\ x - 8, & x \leq 3 \end{cases}$$

# Домашнее задание

1. Составьте программу, которая уменьшает первое число в пять раз, если оно больше второго.
2. Запишите программу, в которой значение переменной  $c$  вычисляется по формуле:  $a+b$ , если  $a$  нечетное и  $a*b$ , если  $a$  четное.



# УРОК 5

# Домашнее задание

## 1. Написать программу решения задачи:

а). Известно значение  $x$ . Вычислить  $y$ , если

$$y = \begin{cases} x + 1, & \text{если } x > 0, \\ x - 1, & \text{если } x \leq 0. \end{cases}$$

б). Даны координаты точки. Выяснить лежит ли эта точка в 3 координатной четверти?

## 2. Определите значение переменной $c$ после выполнения следующего фрагмента программы:

```
a := 6;  
b := 15;  
a := b - a*2;  
if a > b then  
    c := a + b  
else c := b - a;
```

# УРОК 6

# Реализация циклических алгоритмов на Паскале

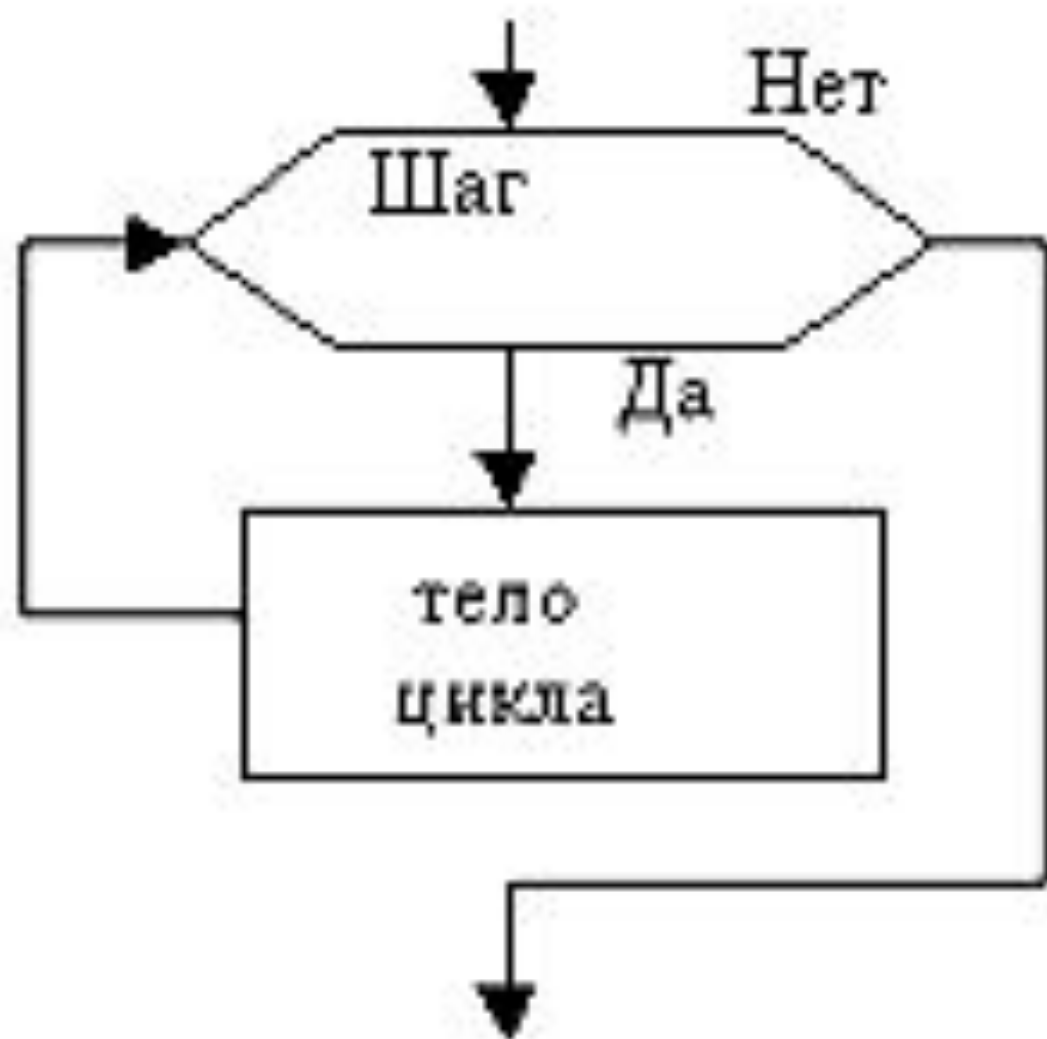


---

**Цикл** – это многократное  
выполнение одинаковой  
последовательности действий.

- **ЦИКЛ С ИЗВЕСТНЫМ ЧИСЛОМ ШАГОВ**  
*(цикл с параметром)*
- **ЦИКЛ С НЕИЗВЕСТНЫМ ЧИСЛОМ ШАГОВ**  
*(цикл с условием)*

# Циклы с параметром



---

## Увеличение переменной на 1 (*шаг 1*):

```
for <переменная> := <начальное значение> to  
    <конечное значение> do begin  
    {тело цикла}  
end;
```

## Уменьшение переменной на 1 (*шаг -1*):

```
for <переменная> := <начальное значение>  
    downto  
    <конечное значение> do begin  
    {тело цикла}  
end;
```



# Особенности:

---

- переменная цикла может быть только целой (`integer`)
- шаг изменения переменной цикла всегда равен 1 (`to`) или -1 (`downto`)
- если в теле цикла только один оператор, слова `begin` и `end` можно не писать:

```
for i:=1 to 8 do  
    writeln( 'Привет' ) ;
```

- если конечное значение меньше начального, цикл (`to`) не выполняется ни разу (проверка условия в начале цикла, цикл с предусловием)

- 
- в теле цикла не разрешается изменять переменную цикла
  - при изменении начального и конечного значения внутри цикла количество шагов не изменится:

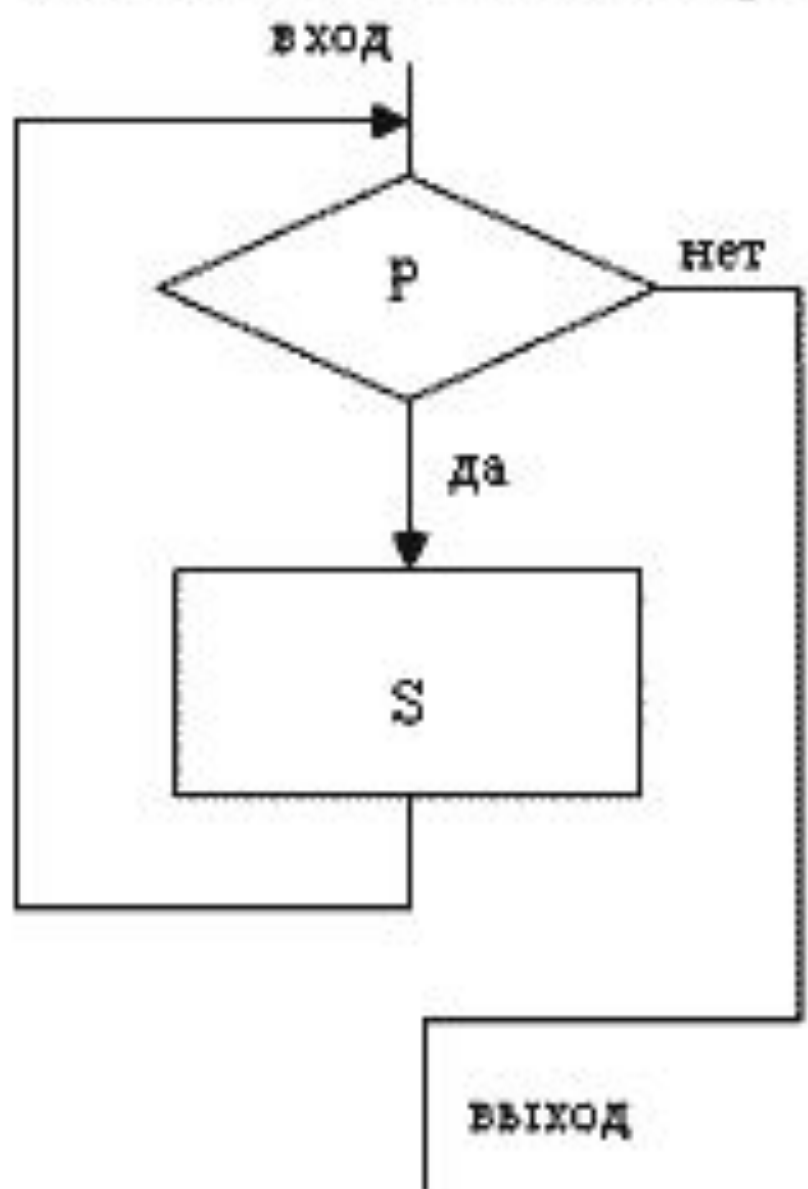
```
n := 8;  
for i:=1 to n do begin  
  writeln('Привет');  
  n := n + 1;  
end;
```

нет  
зацикливания

# Циклы с предусловием

# Цикл – пока

(может не выполняться ни разу)





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

## Особенности:

- МОЖНО ИСПОЛЬЗОВАТЬ СЛОЖНЫЕ УСЛОВИЯ:

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

- если в теле цикла только один оператор, слова **begin** и **end** можно не писать:

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

- 
- условие пересчитывается каждый раз при входе в цикл
  - если условие на входе в цикл ложно, цикл не выполняется ни разу

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

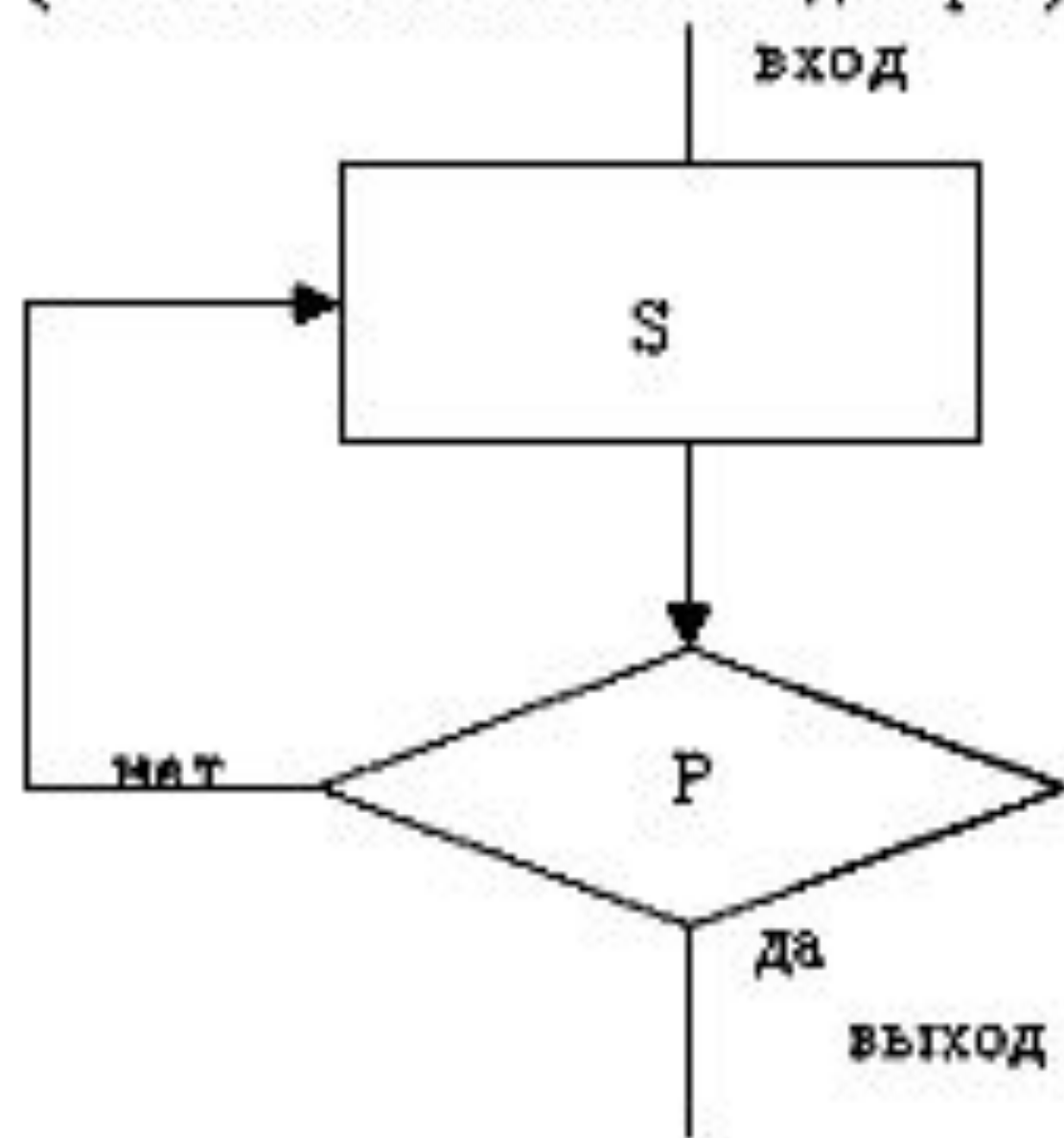
- если условие никогда не станет ложным, программа зацикливается

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

# Циклы с постусловием

# Цикл - до

(выполняется хотя бы один раз)





**Цикл с постусловием** – это цикл, в котором проверка условия выполняется в конце цикла.

```
repeat  
    <тело цикла>  
until <условие>
```

# **ЗАДАЧА**

**Найти сумму квадратов всех натуральных чисел от 1 до 100.**

***Решим эту задачу с использованием всех трех видов циклов.***

# *С использованием цикла "Пока".*

```
Program qq;  
var a, s : integer;  
begin  
  a:=1;  
  s:=0;  
  while a<=100 do begin  
    s:=s+a*a;  
    a:=a+1;  
  end;  
  writeln(s);  
end.
```

# *С использованием цикла*

**"До"**

**Program qq;**

**var a, s : integer;**

**begin**

**a:=1;**

**s:=0;**

**Repeat**

**s:=s+a\*a;**

**a:=a+1;**

**until a >100;**

**writeln(s);**

**end.**



# ***С использованием цикла "С параметром"***

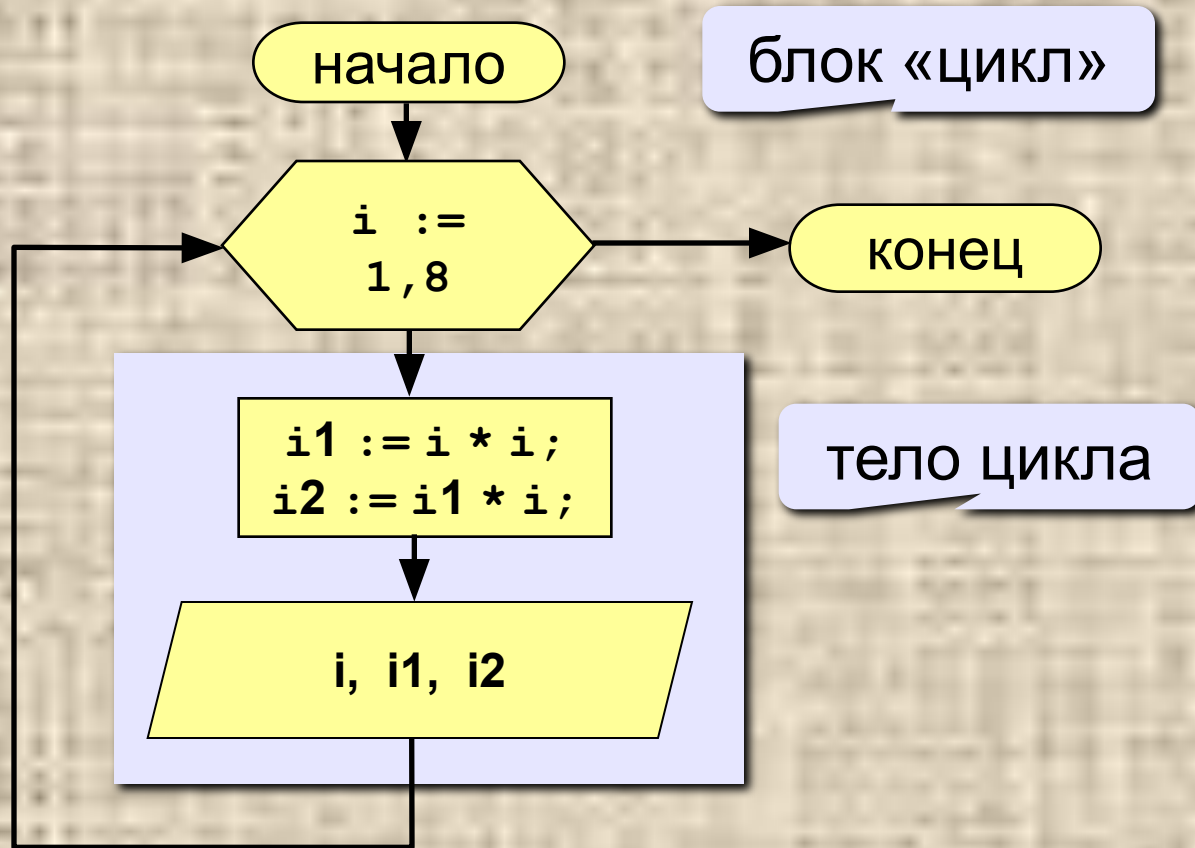
```
Program qq;  
var a, s : integer;  
begin  
    s:=0;  
    for a:=1 to 100 do  
        s:=s+a*a;  
        writeln (s);  
    end.
```

**Задача.**

**Вывести на экран квадраты и кубы  
целых чисел от 1 до 8.**

**Особенность:**

**одинаковые действия выполняются 8  
раз.**



# Программа

---

```
program qq;
```

```
var i, i1, i2: integer;
```

```
begin
```

начальное значение

переменная  
цикла

конечное значение

```
  for i:=1 to 8 do begin
```

```
    i1 := i*i;
```

```
    i2 := i1*i;
```

```
    writeln(i, i1, i2);
```

```
  end;
```

```
end.
```



# Цикл с уменьшением переменной

---

**Задача.** Вывести на экран квадраты и кубы целых чисел от 8 до 1 (в обратном порядке).

**Особенность:** переменная цикла должна уменьшаться.

**Решение:**

```
for i:=8 downto 1 do begin
    i1 := i*i;
    i2 := i1*i;
    writeln(i, i1, i2);
end;
```

# УРОК 7

# Массивы на Паскале

---

**Массив** – это группа однотипных элементов, имеющих общее имя и расположенных в памяти рядом.

**Особенности:**

- все элементы имеют один тип
- весь массив имеет одно имя
- все элементы расположены в памяти рядом

**Примеры:**

- список учеников в классе
- квартиры в доме
- школы в городе
- данные о температуре воздуха за год



**НОМЕР**  
элемента массива  
(ИНДЕКС)

**A**

массив

1

2

3

4

5

5

10

15

20

25

A[1]

A[2]

**ЗНАЧЕНИЕ**  
элемента массива

A[5]

**НОМЕР (ИНДЕКС)**  
элемента массива: 2

A[2]

**ЗНАЧЕНИЕ**  
элемента массива: 10

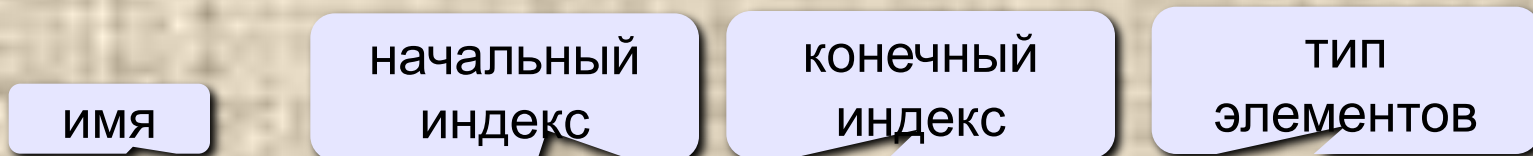
# Объявление массивов

---

## Зачем объявлять?

- определить **ИМЯ** массива
- определить **ТИП** массива
- определить **ЧИСЛО ЭЛЕМЕНТОВ**
- выделить **место в памяти**

## Массив целых чисел:



```
var A : array[ 1 .. 5 ] of integer ;
```

## Размер через константу:

```
const N=5;  
var A : array[1..N] of integer;
```

# Что неправильно?

```
var a: array [1..1  
             0] of integer;
```

...

```
A[5] := 4.5;
```

```
var a: array ['a'..'z'  
            ] of integer;
```

...

```
A['b'  
 ] := 15;
```

```
var a: array [0..9] of integer;
```

...

```
A[10] := 'x';
```

## Объявление:

```
const N = 5;  
var a: array[1..N] of integer;  
    i: integer;
```

## Ввод с клавиатуры.

```
for i:=1 to N do begin  
    write('a[', i, ']=');  
    read ( a[i] );  
end;
```

```
a[1] = 5  
a[2] = 12  
a[3] = 34  
a[4] = 56  
a[5] = 13
```



Почему  
write?

## По:

```
for i:=1 to N do a[i]:=a[i]*2;
```

```
writeln('Массив A:');  
for i:=1 to N do  
    write(a[i]:4);
```

```
Массив A:  
10 24 68 112 26
```



# Задача:

---

1. Ввести с клавиатуры массив из 5 элементов, найти среднее арифметическое всех элементов массива.

Пример:

Введите пять чисел:

4 15 3 10 14

среднее арифметическое 9.200

## РЕШЕНИЕ:

## Program qq;

```
var N: array[1..5] of integer;
```

```
i: integer;
```

```
c, k: real;
```

```
begin
```

```
  for i:=1 to 5 do begin
```

```
    write ('N[',i,']');
```

```
    read (N[i]);
```

```
  end;
```

```
  c:=0;
```

```
  for i:=1 to 5 do begin
```

```
    c:=c + N[i];
```

```
    k:=c/5;
```

```
  end;
```

```
    write('среднее', k:6:2);
```

```
end.
```