



Язык Паскаль

<https://infourok.ru/-13260.html>

<http://ppt4web.ru/informatika/pascal0.html>

```
fp
File Edit Search Run Compile Debug Tools Options Window Help 16:42:53
test.pp 1-[?]
Uses sysutils, dos;

Begin
writeln('- SYSUTILS -');
write('Tamaño del disco actual: ', sysutils.DiskSize(0));
writeln(' (= ', sysutils.disksize(0) div 1024, ' k)');
write('Espacio libre en el disco actual: ', sysutils.diskfree(0));
writeln(' (= ', sysutils.diskfree(0) div 1024, ' k)');

writeln;

writeln('- DOS -');
write('Tamaño del disco actual: ', dos.DiskSize(0));
writeln(' (= ', dos.disksize(0) div 1024, ' k)');
write('Espacio libre en el disco actual: ', dos.diskfree(0));
writeln(' (= ', dos.diskfree(0) div 1024, ' k)');
End.

* 5:36 *
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```



Структура программы на Паскале



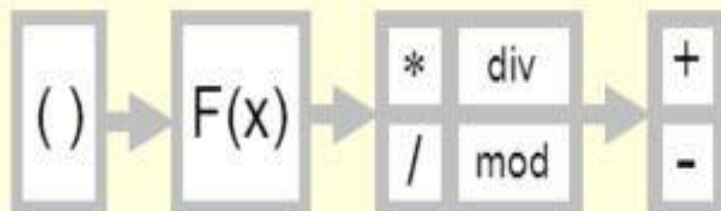
Арифметические операции

+	сложение
-	вычитание
*	умножение
/	деление
div	деление нацело
mod	Остаток от деления

Функции F (x)

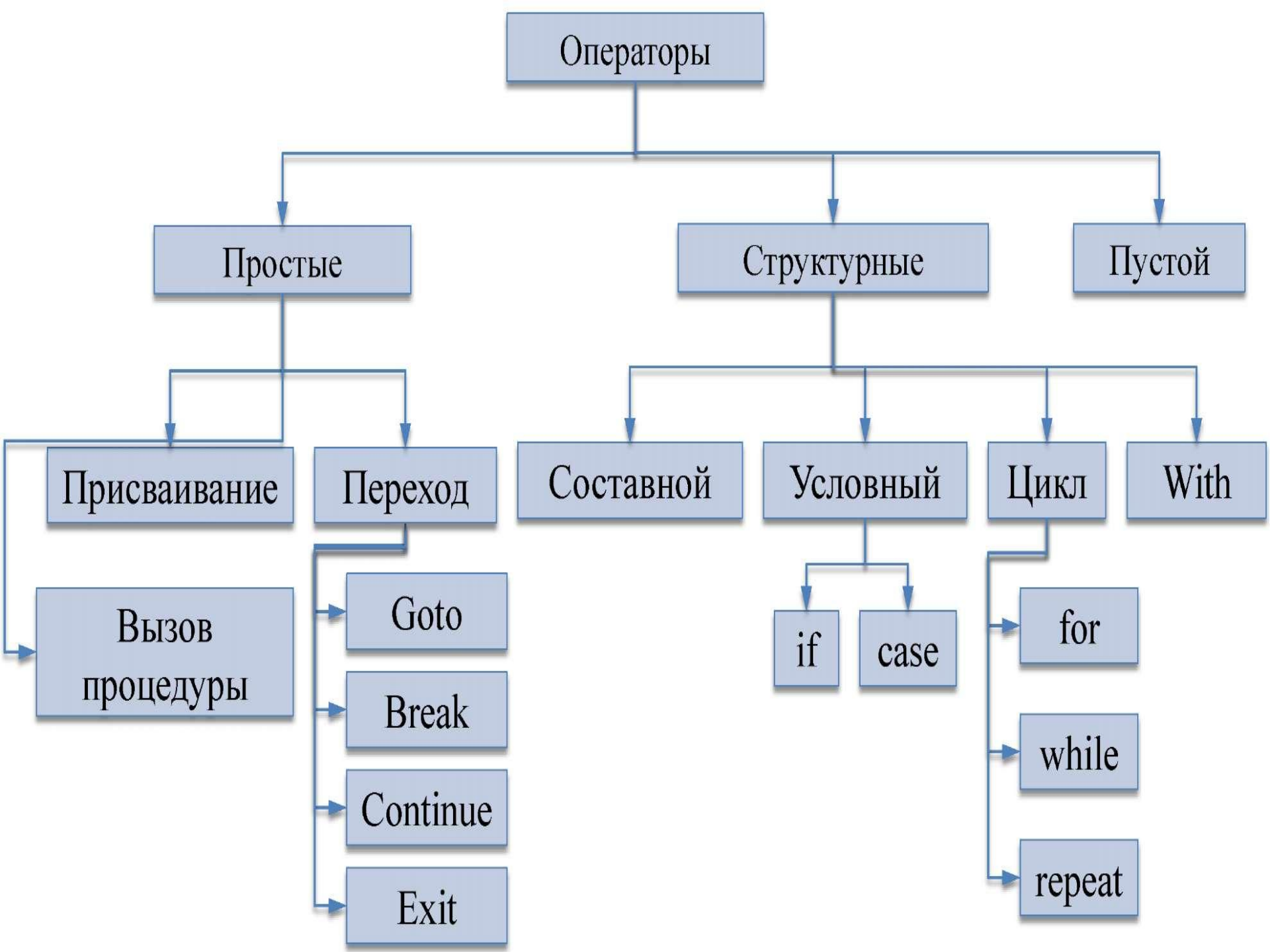
ABS (x)	x
SQR (x)	x^2
SQRT (x)	\sqrt{x}
EXP (x)	e^x
LN (x)	$\ln x$
SIN (x)	$\sin x$
COS (x)	$\cos x$

Приоритеты выполнения операций

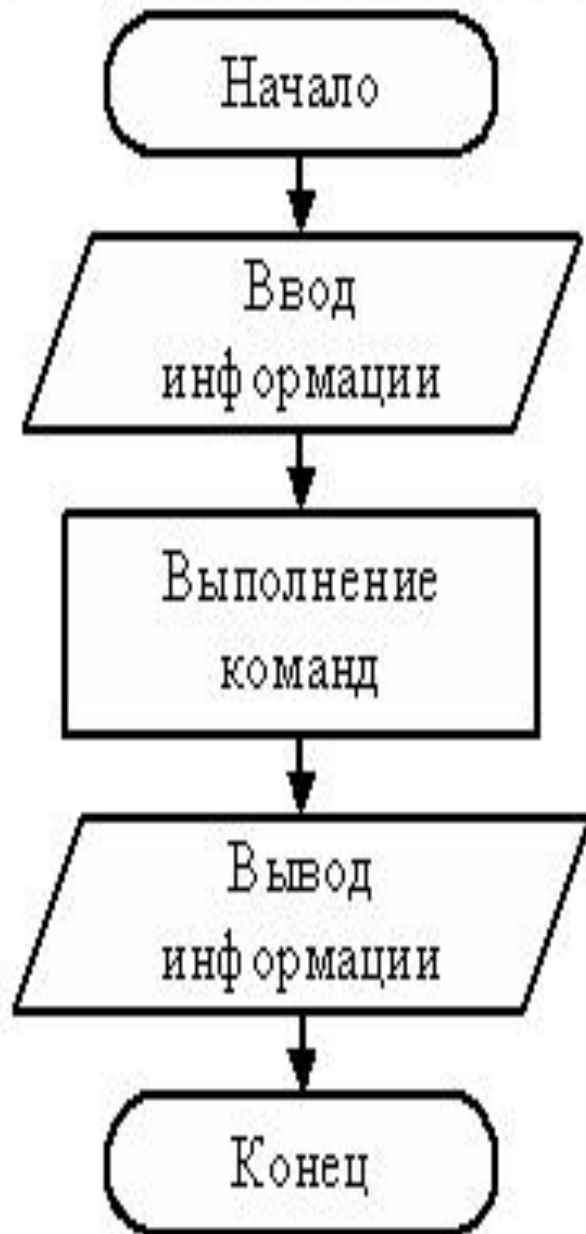


```
PROGRAM ИМЯ ПРОГРАММЫ ;  
VAR список одностипных переменных : ТИП ;  
BEGIN ОПЕРАТОРЫ END .
```

```
PROGRAM ИМЯ ПРОГРАММЫ ;  
VAR список одностипных переменных : INTEGER ;  
    список одностипных переменных : REAL ;  
BEGIN  
    READ ( список ввода ) ;  
    READLN ( список ввода ) ;  
    переменная := арифметическое выражение ;  
    WRITE ( список вывода ) ;  
    WRITELN ( список вывода )  
END .
```



Блок-схема простейшей программы



```
Program <имя>;  
var <описания>;  
begin
```

```
readln (<список переменных>);
```

```
<операторы>;
```

```
writeln(<список переменных>)
```

```
end.
```

Переменные

Переменная – это величина, имеющая имя, тип и значение. Значение переменной можно изменять во время работы программы.

Типы переменных:

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

Объявление переменных (выделение памяти):

```
var a, b: integer;  
    Q: real;  
    s1, s2: string;
```

Типы значений переменных

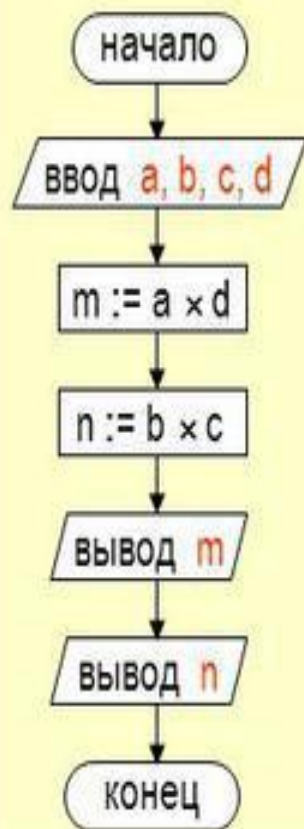
Тип	Перевод	Диапазон принимаемых значений
Integer	целый	целые числа от - 32 768 до 32 767
LongInt	длинное целое	целые числа от - 2 147 483 648 до 2 147 483 647
Byte		целые числа от 0 до 255
Real	Вещественный	целые и дробные числа

Линейные алгоритмы на Паскале



Задача: разделить одну простую дробь на другую - получить результат

$$\frac{a}{b} : \frac{c}{d} = \frac{a \cdot d}{b \cdot c} = \frac{m}{n}$$



алг Деление дробей

цел a, b, c, d, m, n

нач

ВВОД a, b, c, d

$m := a \times d$

$n := b \times c$

ВЫВОД m

ВЫВОД n

конец

```
PROGRAM Division ;
```

```
VAR a, b, c, d, m, n : INTEGER ;
```

```
BEGIN
```

```
  READLN ( a, b, c, d ) ;
```

```
  m := a x d ;
```

```
  n := b x c ;
```

```
  WRITELN ( m ) ;
```

```
  WRITELN ( n )
```

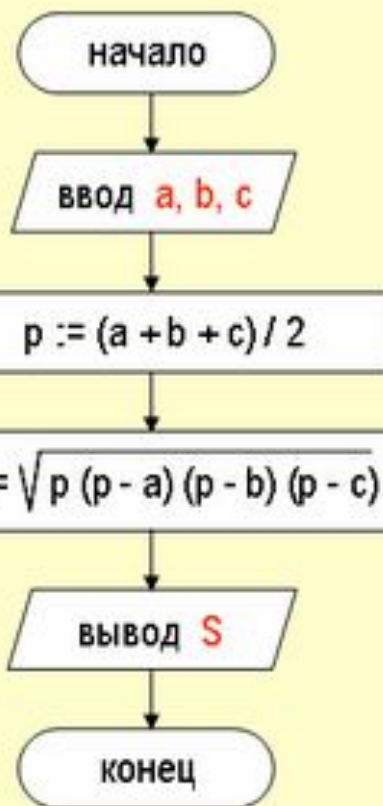
```
END .
```

Пример линейного алгоритма на Паскале

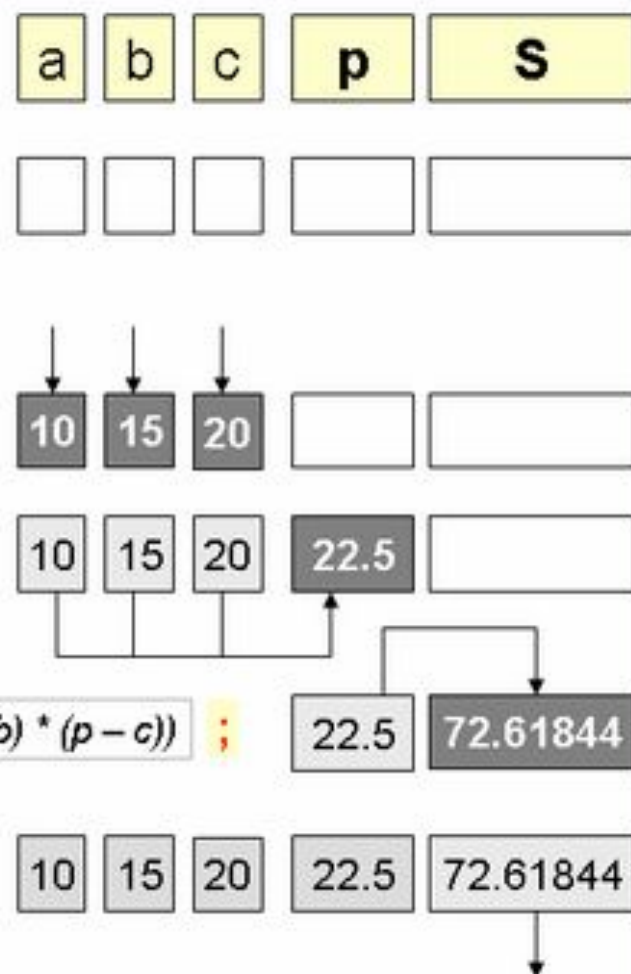


По трем сторонам треугольника (a, b, c) вычислить площадь треугольника (S)

Формула Герона



```
PROGRAM GERON ;  
VAR a, b, c : INTEGER ;  
    p, S : REAL ;  
BEGIN  
    READLN ( a, b, c ) ;  
    p := (a + b + c) / 2 ;  
    S := SQRT ( p * (p - a) * (p - b) * (p - c) ) ;  
    WRITELN ( S )  
END .
```



Структура Pascal-программы

```
program First;
  const Pi = 3.14;
  var r: real;
  S,C: real;
begin
  write('Введите радиус окружности: ');
  readln(r);
  S := Pi*r*r;
  C := 2*Pi*r;
  writeln('Длина окружности равна C=',C);
  writeln('Площадь круга равна S=',S);
end.
```

} **Имя программы Заголовок**

} **Секция описаний**

} **Начало блока операторов**

} **Операторы (блок)**

} **Конец блока операторов**

Внимание!!! Операторы языка отделяются знаком ;

Внимание!!! Программа заканчивается end.

Математическая запись

Запись на языке Pascal

$$37(25+87,5)-17(4,6+1,9)$$

Арифметические действия

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

$$\frac{ab}{c} + d^4$$

$$\frac{a+2b-3c}{5a+4}$$

$$\frac{a+b}{a-b} + \frac{ab}{3,14}$$

$$(a+2*b-3*c)/(5*a+4)$$

$$(a+b)/(a-b)+a*b/3.14$$

Стандартные функции

Математическая
запись

$$\sqrt{15} + \frac{|-n| + 3^{15}}{e^3}$$

$$\frac{\sqrt{ab}}{|c|} + d^4$$

$$\operatorname{tg} 60^\circ * \pi$$

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

Sin(x)	sinx	Вычисление синуса
Cos(x)	cosx	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	π	Число π

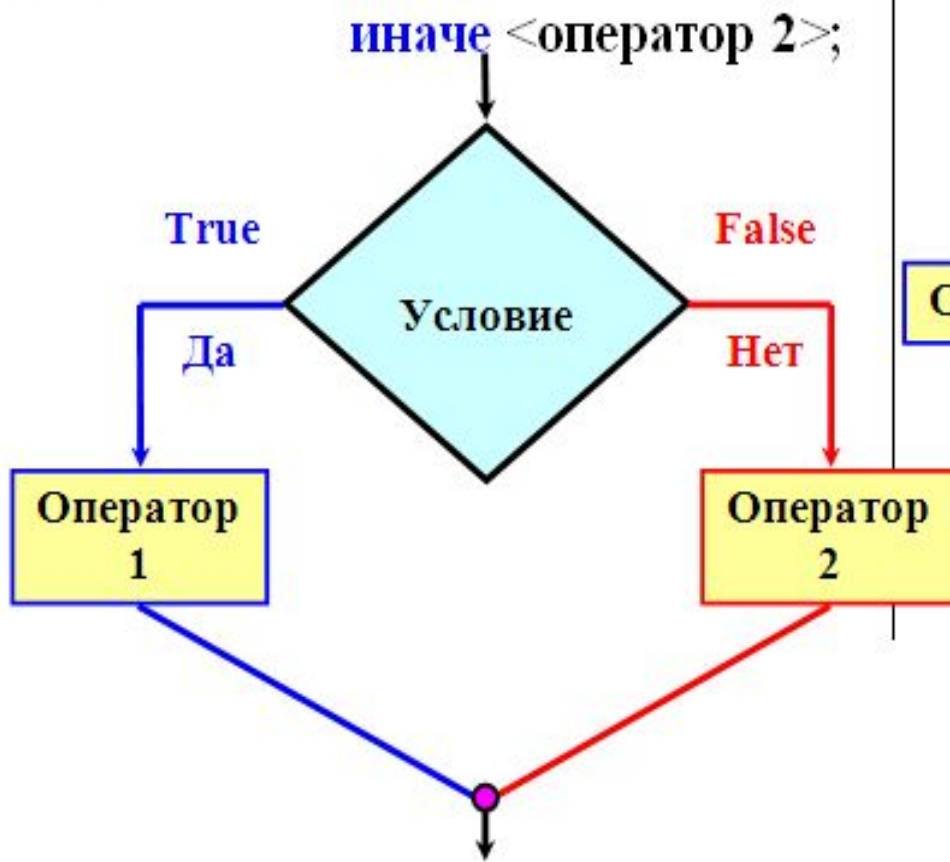
$$\operatorname{tg} X = \frac{\sin x}{\cos x} \quad \operatorname{ctg} X = \frac{\cos x}{\sin x}$$



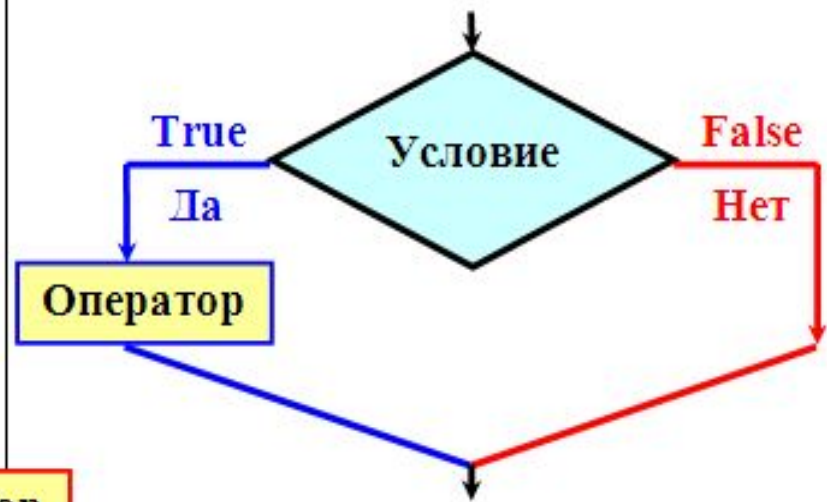
Условный оператор



If <условие> then <оператор 1>
 else <оператор 2>;
 если <условие> то <оператор 1>
 иначе <оператор 2>;



If <условие> then <оператор>;
 если <условие> то <оператор>;



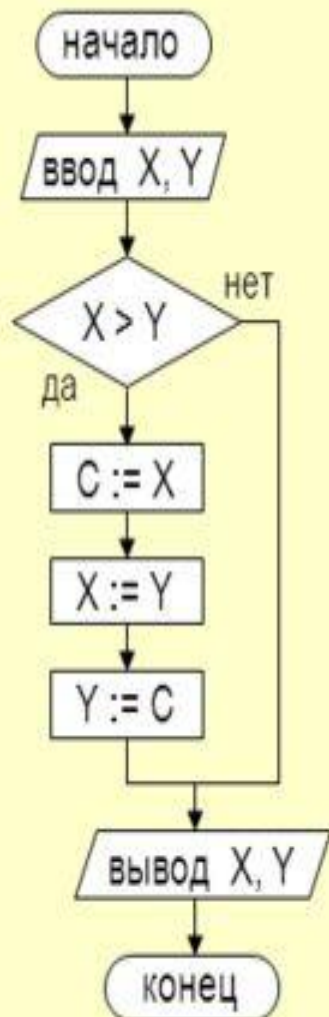
Условия записываются, как правило, в виде логических выражений. Простой формой логического выражения является **операция отношения**.

В Паскале допускаются все виды отношений:

- > - больше;
- < - меньше;
- = - равно;
- <= - меньше или равно;
- >= - больше или равно;
- <> - не равно.

Алгоритм "Сортировка" на Паскале

Задача: составить алгоритм упорядочивания значений двух переменных



```
алг СОРТИРОВКА
вещ X, Y, C
нач

  ввод X, Y

  если X > Y
  то C := X
      X := Y
      Y := C
  кв

  вывод X, Y

кон
```

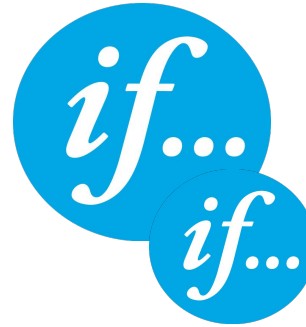
```
PROGRAM SORTING ;
VAR x, y, c : REAL ;
BEGIN
  READLN ( x, y ) ;

  IF x > y
  THEN BEGIN
        c := x ;
        x := y ;
        y := c
      END ;

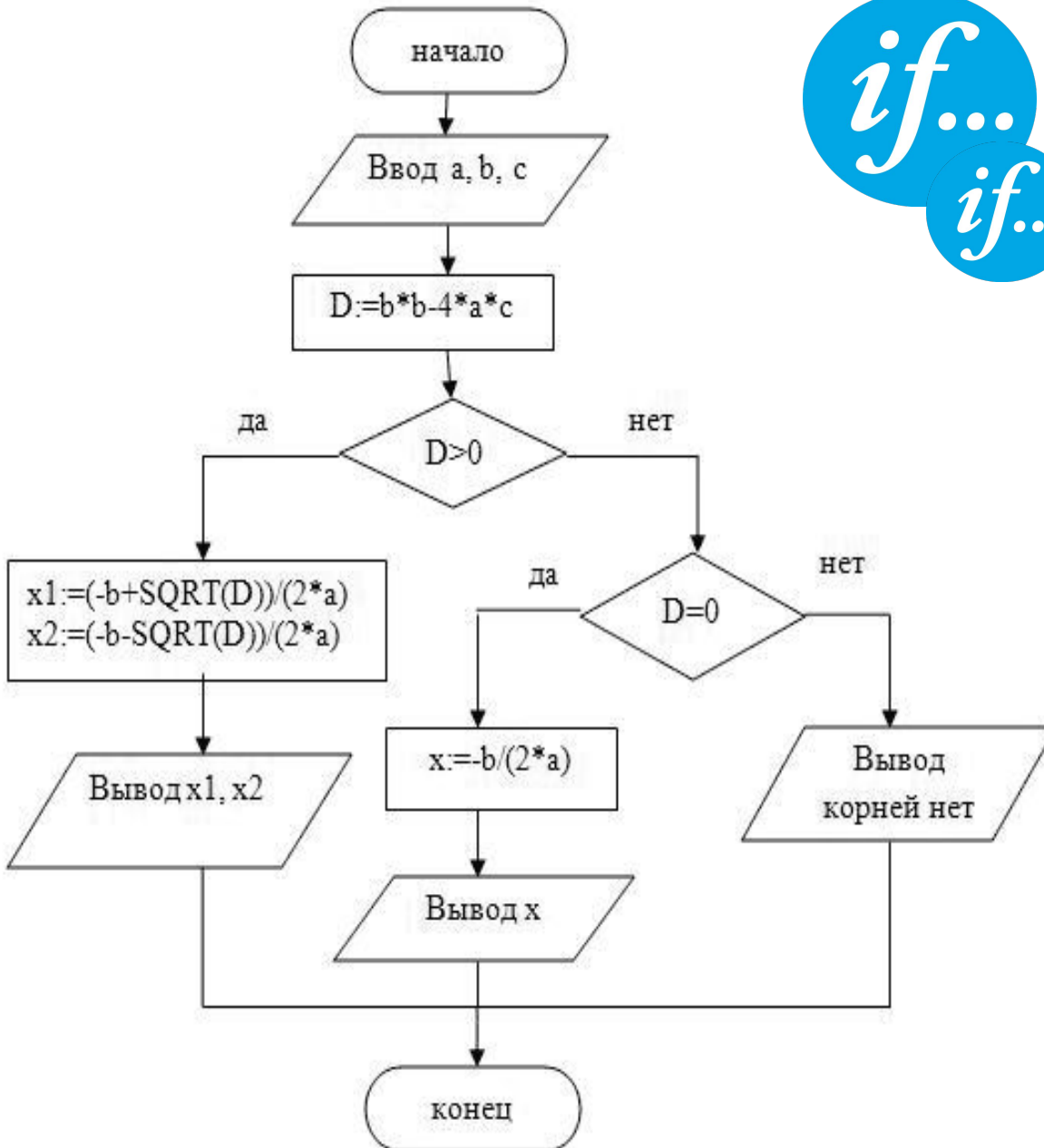
  WRITELN ( x, y )

END .
```

Программа решения квадратного уравнения



```
program z 2;  
var a,b,c,d,x,x1,x2:real;  
begin  
  readln(a,b, c);  
  d:= b*b - 4*a*c;  
  if d>0 then  
  begin  
    x1:=(-b +sqrt(d))/(2*a);  
    x2:=(-b -sqrt(d))/(2*a);  
    writeln ('x1= ',x1,'x2= ',x2);  
  end  
  else  
  if d=0 then  
  begin  
    x := -b/(2*a);  
    writeln ('x = ',x);  
  end  
  else  
  writeln ('корней нет');  
end.
```



В Паскале различают циклы:

- 1) с неопределенным числом повторений; здесь различают цикл с пред-условием (цикл “Пока”) и цикл с пост-условием (цикл “До”).
- 2) с определённым числом повторений (цикл с параметром, или цикл со счетчиком, или цикл “Для”);

3. Цикл с параметром.

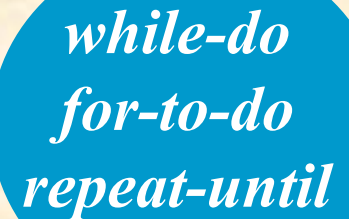
Синтаксическая запись цикла с параметром (цикл “Для”) имеет два варианта:

1) с возрастанием переменной цикла:

```
For <переменная>:=<начальное значение> to <конечное значение> do  
Begin  
<оператор 1>;  
<оператор 2>;  
<оператор 3>;  
...  
End;
```

2) с убыванием переменной цикла:

```
For <переменная>:=<конечное значение> downto <начальное значение> do  
Begin  
<оператор 1>;  
<оператор 2>;  
<оператор 3>;  
...  
End;
```

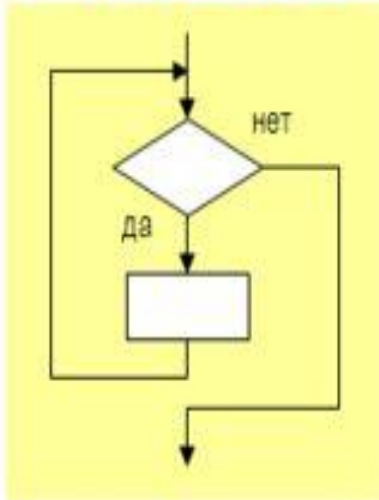


while-do
for-to-do
repeat-until

Циклические операторы на Паскале



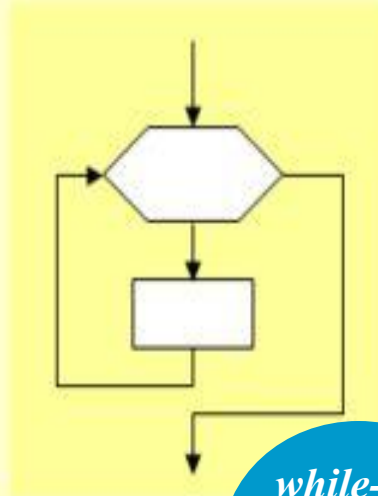
Оператор цикла с предусловием



WHILE `условие` DO

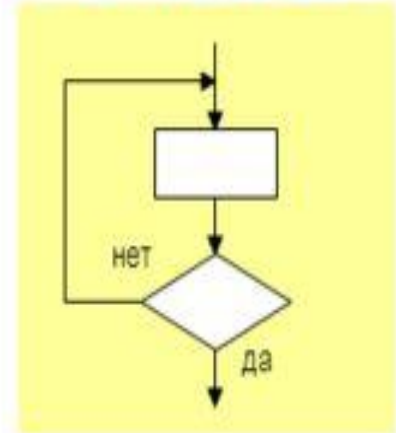
`оператор`

Оператор цикла с параметром



*while-do
for-to-do
repeat-until*

Оператор цикла с постусловием



REPEAT

`оператор`

UNTIL `условие`

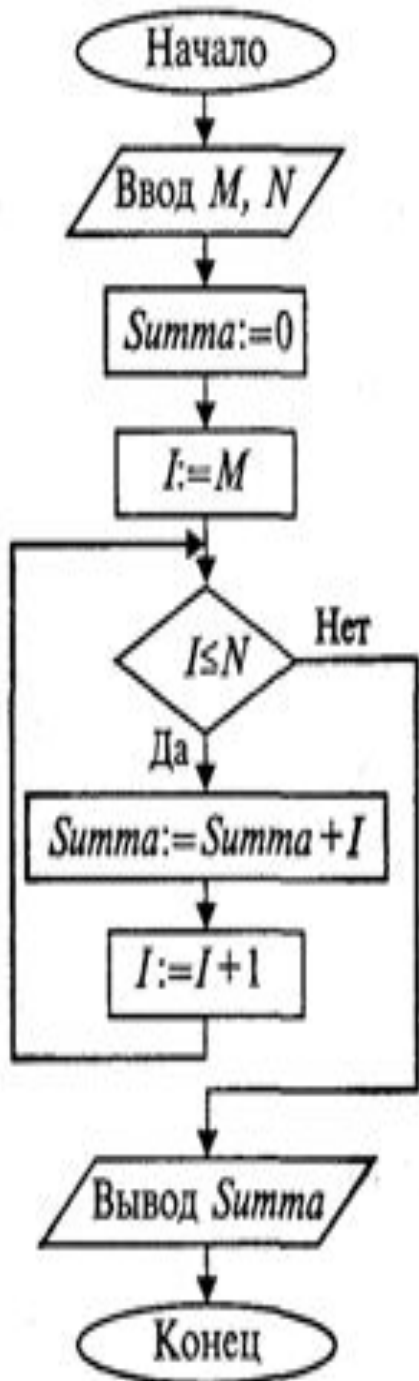
параметр цикла ↑ FOR `переменная` := `выражение 1` TO `выражение 2` DO `оператор`

параметр цикла ↓ FOR `переменная` := `выражение 1` DOWNTO `выражение 2` DO `оператор`

Имя параметра (счетчика) цикла

Начальное значение счетчика цикла

Конечное значение счетчика цикла



```

Program Adding;
Var I,M,N,Summa: Integer;
Begin Write('M=');
      ReadLn(M);
      Write('N=');
      ReadLn(N);
      Summa:=0;
      I:=M;
      While I<=N Do
      Begin
        Summa:=Summa+I;
        I:=Succ(I) (или I:=I+1)
      End;
      WriteLn('Сумма равна',
              Summa)
End.
  
```

```

Program Summing_2;
Var I,M,N,Summa: Integer;
Begin
  Write('M=');
  ReadLn(M);
  Write('N=');
  ReadLn(N);
  Summa:=0;
  For I:=M To N Do
    Summa:=Summa+I;
  WriteLn('Сумма равна',
          Summa)
End.
  
```

*while-do
for-to-do
repeat-until*

**ФУНКЦИЯ -
возвращает следующее
по порядку значение**

Пример задачи

*while-do
for-to-do
repeat-until*

Пример.

Найти сумму квадратов всех натуральных чисел от 1 до 100.

Решим эту задачу с использованием всех трех видов циклов

I. С использованием цикла "Пока".

Program Ex1;

Var

A : Integer;

S : Longint;

Begin

A:=1; S:=0;

While A<=100 Do

Begin

S:=S+A*A;

A:=A+1

End;

Writeln(S)

End.

II. С использованием цикла "До".

Program Ex2;

Var

A : Integer;

S : Longint;

Begin

A:=1; S:=0;

Repeat

S:=S+A*A;

A:=A+1

Until A>100;

Writeln(S)

End.

III. С использованием цикла "С параметром".

Program Ex3;

Var

A : Integer;

S : Longint;

Begin

S:=0;

For A:=1 To 100 Do

S:=S+A*A;

Writeln(S)

End

Сравнение циклов While и Repeat

Цикл ПОКА	Цикл ДО
<code>while</code> условие <code>do</code>	<code>repeat</code>
<code><оператор>;</code>	<code><операторы></code> <code>until</code> условие

while-do
for-to-do
repeat-until

1. В цикле `While` проверка условия выполнения цикла находится в начале цикла, а в `Repeat` – в конце.

Цикл `Repeat` всегда выполняется хотя бы один раз, а цикл `While` может не выполняться ни разу.

2. В цикле `While` выход из цикла осуществляется, если условие ложно, а в `Repeat` – если условие истинно.

3. Между словами `Repeat` и `Until` можно размещать несколько операторов без `Begin` и `End`, а цикл `While...do` может содержать только один оператор тела цикла; поэтому нужно использовать `Begin` и `End`.

Определить k – количество трехзначных натуральных чисел, сумма цифр которых равна n ($1 < n < 27$).

Операции деления ($/$, div и mod) не использовать.

```
var d1, d2, d3, k, n : integer;
begin
writeln('Введите число n, с которым будем сравнивать сумму цифр числа');
readln(n);
k:=0;
{d1 - левая, d2 - средняя, d3 - правая цифры числа}
for d1:=1 to 9 do
for d2:=0 to 9 do
for d3:=0 to 9 do
if d1+d2+d3=n then begin
k:=k+1; write(d1,d2,d3, ' ');
end;
writeln('Количество искомым чисел равно -', k);
end.
```

Формат описания подпрограммы

procedure имя процедуры (формальные параметры);

раздел описаний процедуры

begin

исполняемая часть процедуры

end;

В Паскале различают два вида подпрограмм: **процедуры** и **функции**. Основное различие между ними заключается в том, что процедура получает в результате своей работы любое количество данных, а функция — только одно значение.

function имя функции (формальные параметры):тип результата;

раздел описаний функции

begin

исполняемая часть функции

имя функции := результат;

end;

Формальные параметры в заголовке процедур и функций записываются в виде:

var имя параметра: имя типа

Вызов процедуры производится оператором, имеющим следующий формат:

имя процедуры (список фактических параметров);

При вызове процедур и функций необходимо соблюдать следующие правила:

**количество фактических параметров должно совпадать с количеством формальных;
соответствующие фактические и формальные параметры должны совпадать
по порядку следования и по типу**

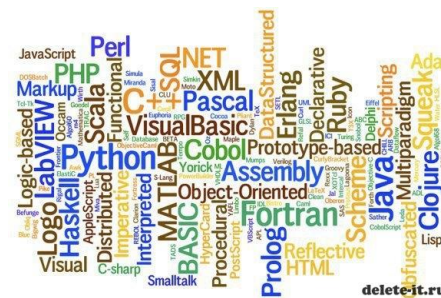
Поиск максимума из двух целых чисел.

```
var x,y,m,n: integer;  
{-----}  
procedure MaxNumber(a,b: integer;  
var max: integer);  
begin if a>b then max:=a else max:=b; end;  
  
{-----}  
begin  
readln(x,y);  
MaxNumber(x,y,m);  
MaxNumber(2,x+y,n);  
writeln('m=',m,'n=',n);  
end.
```

```
Procedure A({ Получаемые переменные });  
var  
{ Блок объявления локальных переменных }  
begin  
{ Тело процедуры }  
end;
```

```
var x,y,m,n: integer;  
{-----}  
function MaxNumber(a,b: integer): integer;  
var max: integer;  
begin if a>b then max:=a else max:=b;  
MaxNumber := max; end;  
  
{-----}  
begin  
readln(x,y);  
m := MaxNumber(x,y);  
n := MaxNumber(2,x+y);  
writeln('m=',m,'n=',n);  
end.
```

```
Function A ({ Получаемые переменные }):byte;  
var  
{ Блок объявления локальных переменных }  
begin  
{ Тело процедуры }  
end;
```



КВАДРАТИЧНАЯ ФУНКЦИЯ

СВОЙСТВА ФУНКЦИИ $f(x) = ax^2 + bx + c$

ВИД ГРАФИКА ФУНКЦИИ $f(x) = ax^2 + bx + c$

$a > 0$

$a < 0$

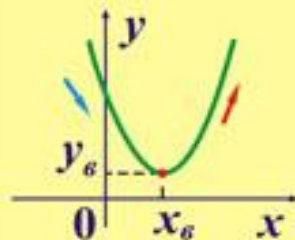
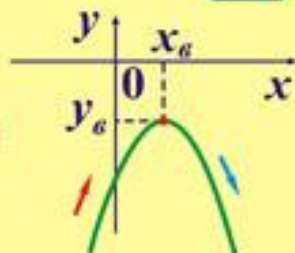


График - парабола



1. $D(f) = (-\infty; +\infty)$

2. $E(f) = [y_e; +\infty)$

2. $E(f) = [-\infty; y_e]$

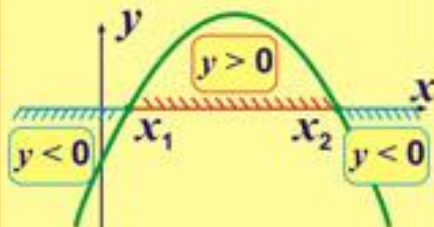
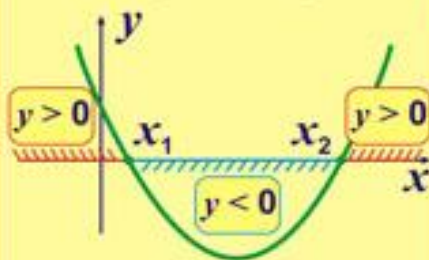
3. $f(x)$ убывает при $x \in (-\infty; x_e]$,

3. $f(x)$ возрастает при $x \in (-\infty; x_e]$,

$f(x)$ возрастает при $x \in [x_e; +\infty)$

$f(x)$ убывает при $x \in [x_e; +\infty)$

4. Промежутки знакопостоянства



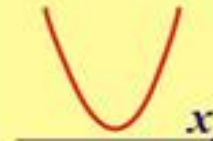
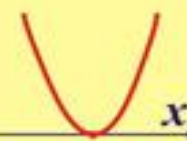
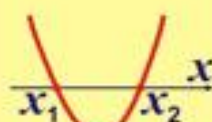
x_1 и x_2 - корни уравнения $ax^2 + bx + c = 0$

$D > 0$

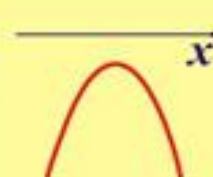
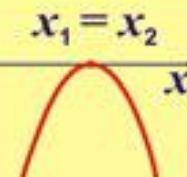
$D = 0$

$D < 0$

$a > 0$



$a < 0$



ПОСТРОЕНИЕ ПАРАБОЛЫ $y = ax^2 + bx + c$

$y = 0,5x^2 - 4x + 6$

1. Вершина параболы

$x_e = \frac{-b}{2a} \quad x_e = \frac{4}{1} = 4$

$y_e = 0,5 \cdot 4^2 - 4 \cdot 4 + 6 = -2$

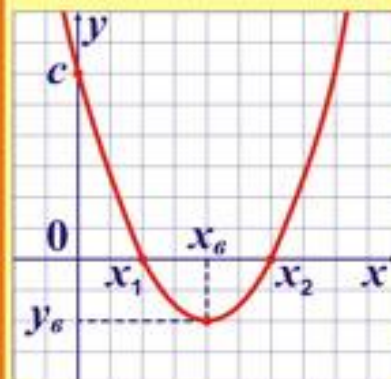
2. Пересечение с осью Oy

$x = 0 \Rightarrow y = c \quad y = 6$

3. Пересечение с осью Ox

$0,5x^2 - 4x + 6 = 0$

$x_1 = 2, \quad x_2 = 6$



Определение массива

Многие задачи, которые решаются с помощью ЭВМ, связаны с обработкой массивов.

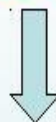
Массив - это упорядоченная совокупность однотипных данных, с каждым из которых связан упорядоченный набор целых чисел, называемых индексами. Массив характеризуется именем, размерностью и размером.



Обработка массива

Главной особенностью алгоритмов обработки одномерных массивов является то, что обработка массивов должна производиться **поэлементно**. По этой причине обработка осуществляется с помощью **циклического алгоритма**, а в качестве переменной цикла обычно используется **индекс элементов**.

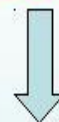
Массивы



ОДНОМЕРНЫЕ

элементы такого массива расположены в виде строки (столбца)

3 5 7 1 4 4 5 8



ДВУМЕРНЫЕ

элементы такого массива расположены в виде таблицы (несколько строк)

3 5 7 1

Структура массива

- Одномерный массив $A[i]$ типа **real** из 9 элементов

Имя массива	A								
Индексы(№)	1	2	3	4	5	6	7	8	9
Элементы	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]
Значения	1.2	0.9	-1.3	-9.1	0.02	45.1	-3	5.0	7
Тип данных	real	real	real	real	real	real	real	real	real

- Двумерный массив $B[i,j]$ типа **integer** из 4 строк и 5 колонок.


- Элемент массива $B[2,4]=34$

индексы	1	2	3	4	5
1	1	4	6	-7	9
2	78	12	45	34	21
3	34	13	6	45	0
4	-6	1	-8	56	-99

Номер строки

Номер столбца

Ввод, вывод элементов одномерных и двумерных массивов

	Одномерный массив <pre>var A : array[1 .. 5] of integer ;</pre> <pre>Const N=5;</pre> <pre>var A: array[1..N] of integer;</pre>	Двумерный массив <pre>Const n=10; m=12;</pre> <pre>Var a:array [1..n,1..m] of integer;</pre> Или <pre>Var B:array [1..10,1..12] of real;</pre>
ВВОД	<i>for i:=1 to n do readln (a(i));</i>	<i>for i:=1 to n do for j:=1 to m do readln (a(i,j));</i>
ВЫВОД	<i>for i:=1 to n do writeln (a(i));</i>	<i>for i:=1 to n do for j:=1 to m do writeln (a(i,j));</i>

Задания 19. Обработка массивов и матриц

[Алгоритмы, меняющие элементы массива местами](#)

[Алгебраические операции с элементами массива,
двумерные массивы](#)

[Алгоритмы с использованием условного оператора](#)

Задания 25 (С2). Алгоритмы обработки массивов

[Вычисление средних, суммы элементов](#)

[Поиск максимального элемента](#)

[Поиск минимального элемента](#)

Действия над элементами массива

1. Ввод и вывод;
2. Копирование массива;
3. Поиск элементов по некоторому условию;
4. Подсчет количества, суммы и произведения;
5. Нахождение максимума (минимума) и их индексов;
6. Сдвиг, вставка и удаление;
7. Сортировка.



Заполнение массива *случайными числами*

Число из интервала $[0, M-1]$

`random (M)`

Вещественное число от 0 до 1

`random`

Число из интервала $[a, b]$

`random (b-a+1)+a`

```
for i:=1 to 10 do
begin
a[i]:=random(101); {числа от 1 до 100}
end;
```