



# ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ

## НАЧАЛА ПРОГРАММИРОВАНИЯ

9 класс

# Ключевые слова

- **вещественный тип данных**
- **целочисленный тип данных**
- **символьный тип данных**
- **строковый тип данных**
- **логический тип данных**



# Числовые типы данных

Стандартные функции языка Паскаль:

| Функция                 | Назначение                          | Тип аргумента | Тип результата            |
|-------------------------|-------------------------------------|---------------|---------------------------|
| <code>abs (x)</code>    | Модуль $x$                          | integer, real | Такой же, как у аргумента |
| <code>sqr (x)</code>    | Квадрат $x$                         | integer, real | Такой же, как у аргумента |
| <code>sqrt (x)</code>   | Квадратный корень из $x$            | integer, real | real                      |
| <code>round (x)</code>  | Округление $x$ до ближайшего целого | real          |                           |
| <code>frac (x)</code>   | Целая часть $x$                     | real          |                           |
| <code>int (x)</code>    | Дробная часть $x$                   | real          |                           |
| <code>random</code>     | Случайное число от 0 до 1           | -             | real                      |
| <code>random (x)</code> | Случайное число от 0 до $x$         | integer       | integer                   |

# Исследование функций **round**, **int** и **frac**

```
program n_3;  
  var x: real;  
begin  
  writeln ('Исследование функций round, int, frac');  
  write ('Введите x>>');  
  readln (x);  
  writeln ('Округление - ', round(x));  
  writeln ('Целая часть - ', int(x));  
  writeln ('Дробная часть - ', frac(x))  
end.
```

Выполните программу несколько раз для

$x \in \{10,2; 10,8; -10,2; -10,8\}$ .

Какой будет тип результата каждой из этих функций?



# Целочисленный тип данных

Операции над целыми числами в языке Паскаль:

| Операция                         | Обозначение | Тип результата |
|----------------------------------|-------------|----------------|
| Сложение                         | +           | integer        |
| Вычитание                        | -           | integer        |
| Умножение                        | *           | integer        |
| Получение целого частного        | div         | integer        |
| Получение целого остатка деления | mod         | integer        |
| Деление                          | /           | real           |

# Операции **div** и **mod**

Трёхзначное число можно представить в виде следующей суммы:  $x = a \cdot 100 + b \cdot 10 + c$ , где  $a, b, c$  - цифры числа.

Программа нахождения суммы цифр вводимого с клавиатуры целого трёхзначного числа.

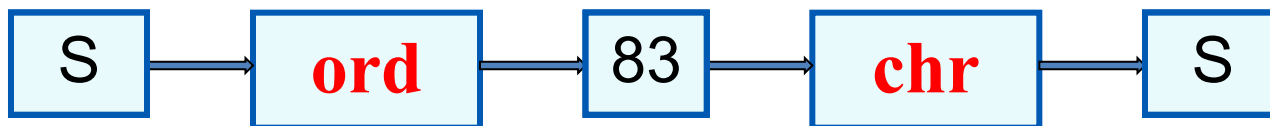
```
program n_4;  
  var x, a, b, c, s: integer;  
begin  
  writeln ('Нахождение суммы цифр трёхзначного числа');  
  write ('Введите исходное число>>');  
  readln (x);  
  a:=x div 100;  
  b:=x mod 100 div 10;  
  c:=x mod 10;  
  s:=a+b+c ;  
  writeln ('s= ', s)  
end.
```

# Символьный и строковый ТИПЫ ДАННЫХ

**Символы** – это все буквы и значки, которые есть на клавиатуре. Для ввода в программу символьных переменных необходимо указать для них символьный тип данных **char**.

Функция **ord** преобразовывает букву в её числовой код.

Функция **chr** преобразовывает числовой код символа в сам СИМВОЛ.



Значением строковой величины (тип **string**) является произвольная последовательность символов, заключенная в апострофы.

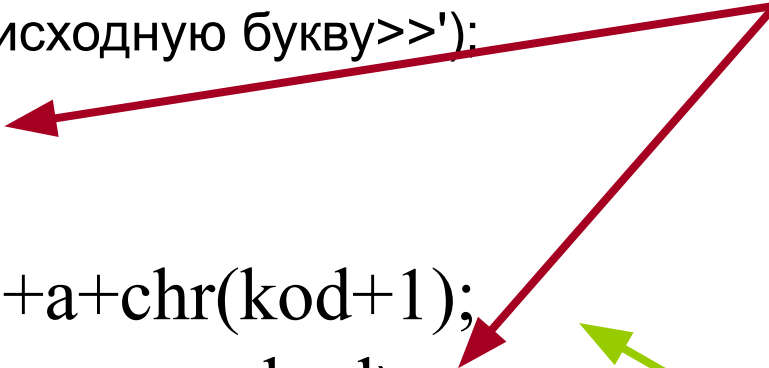
```
var c: string  
c:= chr(52)+chr(37)
```




# Символьный и строковый ТИПЫ ДАННЫХ

```
program n_5;  
  var a: char; kod: integer; b: string;  
begin  
  writeln ('Код и строка');  
  write ('Введите исходную букву>>');  
  readln (a);  
  kod:=ord(a);  
  b:=chr(kod-1)+a+chr(kod+1);  
  writeln ('Код буквы ', a, '-', kod);  
  writeln ('Строка: ', b)  
end.
```

Вывод на экран  
кода буквы  
введённой с  
клавиатуры



Вывод на экран  
строки из трёх  
букв.  
Каких?





# Логический тип данных

Величины логического типа принимают всего два значения:

**false** и **true**;

**false** < **true**.

Логические значения получаются в результате выполнения операций сравнения числовых, символьных, строковых и логических выражений.

В Паскале логической переменной можно присваивать результат операции сравнения.

# Логический тип данных

Пусть **ans** - логическая переменная,  
*n* - целая переменная.

В результате выполнения оператора присваивания  
**ans:=n mod 2=0**

переменной **ans** будет присвоено значение **true** при любом чётном *n* и **false** в противном случае.

```
program n_6;  
  var n: integer; ans: boolean;  
begin  
  writeln ('Определение истинности высказывания о чётности числа');  
  write ('Введите исходное число>>');  
  readln (n);  
  ans:=n mod 2=0;  
  writeln ('Число ', n, ' является четным - ', ans)  
end.
```

# Логический тип данных

Логическим переменным можно присваивать значения логических выражений, построенных с помощью логических функций и (**and**), или (**or**), не (**not**).

| Логическая операция в Паскале | Название операции                    |
|-------------------------------|--------------------------------------|
| and                           | конъюнкция<br>(логическое умножение) |
| or                            | дизъюнкция<br>(логическое сложение)  |
| not                           | отрицание (инверсия)                 |

# Логический тип данных

```
program n_7;  
  var a, b, c: integer; ans: boolean;  
begin  
  writeln ('Определение истинности высказывания  
           о равнобедренном треугольнике');  
  write ('Введите значения a, b, c>>');  
  readln (a, b, c);  
  ans:=(a=b) or (a=c) or (b=c);  
  writeln ('Треугольник с длинами сторон ', a, ', ', b,  
          ', ', c, ' является равнобедренным - ', ans)  
end.
```

# Самое главное

*Типы данных* в языке Паскаль:

- вещественный
- целочисленный
- символьный
- строковый
- логический и другие.

Для них определены соответствующие операции и функции.





# Опорный конспект

**Типы данных** в языке Паскаль: вещественный, целочисленный, символьный, строковый, логический.

