

## Типы данных

**Целочисленные типы** данных представляют собой значения, которые могут использоваться в арифметических выражениях. Как правило, диапазон целых чисел определяется количеством байтов в памяти компьютера, отводимых под одну переменную. В языке Pascal целые числа занимают в памяти от 1 до 4 байт.

Тип	Диапазон	память (байт)
byte	0..255	1
shortint	-128..127	1
integer	-32768..32767	2
word	0..65535	2
longint	-2147483648..2147483647	4

При работе с целочисленными типами следует руководствоваться «вложенностью» типов, то есть везде, где можно использовать тип word, допускается использование типа byte (но не наоборот), в longint «входит» integer, который, в свою очередь, включает в себя shortint.

При работе с целыми числами в Pascal допускаются следующие арифметические операции:

+	сложение
-	вычитание
*	умножение
div	целочисленное деление
mod	остаток от целочисленного деления
+	унарная операция сохранения знака
-	унарная операция изменения знака

**Целочисленное деление (div)** отличается от обычной операции деления тем, что возвращает целую часть частного, дробная часть отбрасывается. Если делимое меньше делителя, то результат целочисленного деления равен нулю.

Выражение	Результат
14 div 5	2
10 div 3	3
2 div 3	0

**Остаток от целочисленного деления (mod)** восстанавливает остаток, полученный при выполнении целочисленного деления.

Выражение	Результат
11 mod 5	1
34 mod 22	12

Обращение	Функция
abs(x)	Модуль аргумента.
dec(x,n)	Уменьшает значение целочисленной переменной x на n. При отсутствии второго параметра значение x уменьшается на 1.
inc(x,n)	Увеличивает значение целочисленной переменной x на n. При отсутствии второго параметра значение x увеличивается на 1.
odd(x)	Возвращает значение True, если аргумент – нечетное число, и False, если x – четное.
random(x)	Генерирует значение псевдослучайного числа из диапазона [0..x].
randomize	Инициализация генератора псевдослучайных чисел (обновление базы случайных чисел).
sq(x)	Возведение аргумента в квадрат.

**Вещественные типы** данных представляют собой действительные значения, которые используются в арифметических выражениях. В языке Pascal целые числа занимают в памяти от 4 до 10 байт.

При работе с вещественными числами в Pascal допускаются следующие арифметические операции:

Тип	Диапазон	байт
real	2.9*10E-39.. 1.7*10E38	6
single	1.5*10E-45 ..3.4*10E38	4
double	5.0*10E-324 ..1.7*10E308	8
extended	1.9*10E-4951 ..1.1*10E4932	10
comp	-2 <sup>63</sup> ..2 <sup>63</sup> -1	8

+	сложение
-	вычитание
*	умножение
/	деление
+	унарная оп-ция сохранения знака
-	унарная оп-ция изменения знака

Для работы с вещественными данными могут использоваться стандартные математические функции:

Обращение	Тип арг.	Тип рез-та	Функция
abs(x)	I, R	I, R	Модуль аргумента.
arctan(x)	I, R	R	Арктангенс аргумента (результат в радианах).
cos(x)	I, R	R	Косинус (угол в радианах).
exp(x)	I, R	R	Экспонента e <sup>x</sup> .
frac(x)	I, R	R	Дробная часть аргумента.
int(x)	I, R	R	Целая часть аргумента.
ln(x)	I, R	R	Натуральный логарифм, т.е. логарифм по основанию e (e = 2,718282).
Pi		R	Возвращает значение числа Пи (3,141592653897932385).
Random		R	Генерирует значение псевдослучайного числа из диапазона [0..1).
randomize			Инициализация генератора псевдослучайных чисел.

round(x)	R	I	Возвращает значение x, округленное до ближайшего целого числа.
sin(x)	I, R	R	Синус (угол в радианах).
sq(x)	I, R	I, R	Возведение аргумента в квадрат.
sqrt(x)	I, R	R	Вычисление квадратного корня из аргумента.
trunc(x)	R	I	Возвращает ближайшее целое число, меньше или равное x, если x >= 0, и больше или равное x, если x < 0.

Примеры.

Математическое выражение

Выражение на Паскале

$$x^2 - 7x + 6$$

$$\text{sq}(x) - 7*x + 6$$

$$|x| - |y|$$

$$(\text{abs}(x) - \text{abs}(y)) / (1 + \text{abs}(x*y))$$

$$1 + |xy|$$

$$\ln \left( (y - \sqrt{|x|}) \left( x - \frac{y}{z + x^2} \right) \right)$$

$$\ln(\text{abs}((y - \text{sqrt}(\text{abs}(x))) * (x - y / (z + \text{sqrt}(x))))))$$

**Литерный (символьный) тип** определяет множество значений кодовой таблицы. В программе значения переменных и констант типа char должны быть заключены в апострофы. Например, 'A', ' ', ' ', ' ' – символ A, пробел, пустой символ, запятая.

Тип	Диапазон	Байт
char	Кодовая таблица	1

**Булевский тип** применяется в логических выражениях и имеет только 2 значения: True и False.

Тип	Диапазон	Байт
boolean	True, False	1

**Структура программы** выглядит следующим образом:

Program <имя>;

} заголовок программы

```

Uses < имя_модуля 1, имя_модуля 2, ... >;
Label ...;
Const ...;
Type ...;
Var ...;
Procedure < имя_процед уры >;
  < тело процедуры >
Function < имя_функци и >;
  < тело функции >

```

описательная часть блока

```

Begin
  < операторы >
End.

```

исполнительная часть блока

Label – раздел описания меток

Const – раздел описания констант

Type – раздел описания типов

Var – раздел описания переменных

**Условный оператор** часто называют оператором выбора или альтернативой, так как в зависимости от выполнения определенного условия выбирается тот или иной путь решения задачи. Оператор *if* разрешает выбор одной из двух альтернатив и может принимать одну из следующих форм:

**полную** if <условие> then <действие1>  
else <действие2>;

**или сокращенную** if <условие> then <действие>;

здесь ключевые слова *if*, *then*, *else* означают соответственно "если", "то", "иначе".

1. Условие – это выражение булевского типа, оно может быть простым или сложным. Простое условие состоит из двух операндов и оператора сравнения. В Pascal предусмотрено 6 операторов сравнения: = (равно); > (больше); < (меньше); >= (больше или равно); <= (меньше или равно); <> (не равно).
2. Сложные условия образуются с помощью логических операций:
  - **and** (логическое И);
  - **or** (логическое ИЛИ);
  - **not** (логическое НЕ);
  - **xor** (логическое исключающее ИЛИ).

**Важно.** Входящие в составные условия, простые условия обязательно заключают в скобки.

**Пример** if (a >= 0) and (a <= 24) then ...

3. Если <действие> предполагает выполнение нескольких операторов, то эти операторы необходимо заключить в операторные скобки **begin ...end**