

# Арифметические действия в языке Си

Математические вычисления не ограничиваются лишь арифметическими действиями. Кроме них, можно ещё встретить корни, модули, логарифмы, тригонометрические функции и пр. Научимся же использовать подобные функции в своих программах.

Для использования математических функций нужно подключить заголовочный файл `math.h`. В ней определено много различных функций, но мы пока рассмотрим следующие:

Некоторые математические функции

<b><code>fabs(x)</code></b>	модуль числа <code>x</code>
<b><code>sqrt(x)</code></b>	квадратный корень из числа <code>x</code>
<b><code>sin(x)</code></b>	синус числа <code>x</code> ( <code>x</code> в радианах)
<b><code>cos(x)</code></b>	косинус числа <code>x</code> ( <code>x</code> в радианах)
<b><code>pow(x, y)</code></b>	вычисление <code>x</code> в степени <code>y</code>
<b><code>exp(x)</code></b>	вычисление <code>e</code> в степени <code>x</code>
<b><code>log(x)</code></b>	натуральный логарифм числа <code>x</code>
<b><code>log10(x)</code></b>	десятичный логарифм числа <code>x</code>

Два важных момента.

- Все функции возвращают значение типа **`double`**.
- Параметры функций – вещественные числа (**`double`**), но можно передавать и целые числа. При этом произойдёт неявное преобразование типа. Компилятор из целого числа, например **`3`**, сделает вещественное **`3.0`**.

Примеры.

Даны длины катетов прямоугольного треугольника. Вычислить длину гипотенузы. Простая задачка на знание теоремы Пифагора.

Листинг 1.

```
#include <stdio.h>
#include <math.h> // подключаем math.h
int main (void) {
    int a, b, c2;

    scanf ("%d", &a);
    scanf ("%d", &b);

    c2 = a*a + b*b;
    printf ("c = %.2f\n", sqrt(c2));

    return 0;
}
```

}

Вычислить синус угла введённого с клавиатуры. Угол вводится в градусах.

Листинг 2.

```
#include <stdio.h>
#include <math.h> // подключаем math.h
int main (void) {
    double alpha, sin_a, pi = 3.1415926;

    scanf("%lf",&alpha);
    alpha = alpha*pi/180;
    sin_a = sin(alpha);
    printf("%.2f\n", sin_a);

    return 0;
}
```

В этой программе есть о чём поговорить. Тригонометрические функции, которые определены в **math.h** работают с радианной мерой угла. Людям же привычнее работать с градусами. Поэтому в данной программе мы предварительно перевели значение из градусов в радианы. Если этого не сделать, результат получится неправильным. Проверьте это самостоятельно.

## Неявное преобразование типов

При явном преобразовании типа мы в скобках перед значением указывали тип, к которому нужно привести данное значение. В неявном преобразовании этого делать не нужно. Компилятор автоматически подберёт необходимый тип.

Неявное преобразование типов осуществляется в следующих случаях:

1. перед передачей аргументов в функцию (как в нашем примере с корнем. Листинг 1.)
2. выполнение арифметических операций с разными типами аргументов
3. перед выполнением присваивания

### Правила неявного преобразования типов

если выполняются арифметические операции с разными типами аргументов. Оба аргумента приводятся к большему типу. Порядок типов: **int < float < double**

при присваивании. Значение справа от оператора присваивания приводится к типу переменной слева от оператора присваивания. При этом, если больший тип присваивается меньшему, то может произойти потеря точности.

Примеры:

<code>int+float</code>	будет автоматически преобразовано к <code>float+float</code>
<code>float/int</code>	будет автоматически преобразовано к <code>float/float</code>
<code>double*float</code>	будет преобразовано к <code>double*double</code>
<code>int = double</code>	<code>double</code> будет преобразовано к <code>int</code> с потерей дробной части
<code>float = int</code>	<code>int</code> будет преобразовано к <code>float</code>

## Практика.

1. Решите предложенные [задачи с автоматической проверкой решения](#).

Интернет версия: [http://youngcoder.ru/lessons/4/math\\_function.php](http://youngcoder.ru/lessons/4/math_function.php)