

# Арифметика целых чисел

## Переменные

Программы оперируют с данными. Данные хранятся в переменных — специальных областях памяти, отведенных для хранения данных. Каждая переменная имеет свой **тип** и должна быть объявлена перед использованием. При использовании в программе ранее не объявленной переменной компилятор выдаст сообщение об ошибке вроде

```
test.cpp: In function 'int main()':
test.cpp:5: 'c' undeclared (first use this function)
test.cpp:5: (Each undeclared identifier is reported only once
for each function it appears in.)
```

В этом сообщении указывается имя файла и номер строки, в которой обнаружена ошибка (test.cpp:5), а также описание ошибки ('c' undeclared — объект с именем 'c' не объявлен).

Объявление переменной имеет следующий вид:

<тип переменной> <один или несколько идентификаторов переменных через запятую>;

Например, переменные `n` и `m` типа `int` можно объявить такой строкой:

```
int n, m;
```

Переменную `x` типа `double` можно объявить такой строкой:

```
double x;
```

Значения переменных сразу после объявления не определены и являются произвольными. Например, если написать следующую программу:

```
int a;
cout << a;
```

то на экран будет выведено произвольное целое число, например, `-1208725516` (так называемый "мусор").

Идентификатор переменной — это ее имя, которое должно быть последовательностью букв латинского алфавита, символа подчеркивания и цифр, начинающейся с буквы. Примеры правильных идентификаторов: `a`, `i`, `Year`, `school179`. Имена переменных чувствительны к регистру букв, то есть `Number`, `number`, `NUMBER` и `nUMbeR` — это четыре разных переменных.

## Арифметические операторы

Арифметическая инструкция — это некоторое выражение, состоящее из констант, идентификаторов переменных и арифметических операторов, которая завершается точкой с запятой. Самый главный арифметический оператор — это

оператор присваивания '=', который присваивает одной переменной, идентификатор которой указывается слева от оператора '=' значение некоторого выражения, которое стоит справа. Например:

```
a = 2;  
b = a + 1;
```

Мы начнем с рассмотрения только целочисленных величин. Для хранения в памяти целочисленных величин как правило используются переменные типа `int`. С переменными типа `int` можно делать следующие арифметические действия:

- + Сложение
- Вычитание
- \* Умножение
- / Деление (целочисленное)
- % Остаток от деления

Особого внимания заслуживают операторы деления и остатка. Деление производится нацело, с отбрасыванием целой части. Например,  $10/3$  это 3,  $179/10$  это 17,  $2/5$  это 0. Оператор % возвращает остаток от деления того, что записано справа на то, что записано слева. Например,  $10\%3$  это 1,  $179\%10$  это 9,  $2\%5$  это 2.

В арифметическом выражении сначала выполняются слева направо все операторы умножения и деления, затем слева направо все операторы сложения и вычитания, затем справа налево все операторы присваивания. При необходимости порядок действий можно изменить при помощи скобок.

## Ввод-вывод

Для того, чтобы вывести на экран значение переменной или текстовой строки нужно использовать объект "cout" и оператор "<<", который в данном случае следует называть "Поместить в". cout — объект, связанный со стандартным выводом программы, как правило, это терминал. Для того, чтобы перейти при печати на новую строку необходимо поместить в cout стандартный объект endl.

Текстовые строки при выводе на экран необходимо заключать в двойные кавычки. Если хочется вывести на экран несколько объектов (переменных, текстовых строк и т.д.), то их нужно разделять между собой оператором '<<'.

Для того, чтобы считать значение переменной нужно использовать объект "cin" и оператор '>>', который надо называть "Извлечь из". При этом считывание данных будет производиться со стандартного ввода программы, как правило, являющегося клавиатурой. Если хочется за одну операцию считать несколько переменных, то их идентификаторы нужно разделять между собой оператором '>>'.

## Более сложный пример

Обобщим все изложенное выше в более сложной программе, которая находит сумму двух введенных чисел.

```
#include <iostream>

using namespace std;

int main()
{
    int a, b, s;
    cout << "Введите два числа: ";
    cin >> a >> b;
    s = a + b;
    cout << a << "+" << b << "=" << s << endl;
    return 0;
}
```