

2 Арифметика

2.1 Переменные

Все переменные в языке C++ должны быть объявлены 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. Предположение о том, что все переменные первоначально имеют нулевые значения, является ошибочным.

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.

2.2 Типы данных

При объявлении переменной мы должны указать ее тип. Существует несколько стандартных типов, кроме того, программист может создавать собственные типы данных. Для представления целых чисел существуют следующие стандартные типы:

В языках C, C++ есть следующие стандартные типы данных для представления целых чисел.

Тип	Описание	Размер	Диапазон	Синоним
int	Целые числа	4 байта	$-2^{31}..2^{31}-1$	
unsigned int	Беззнаковые целые	4 байта	$0..2^{32}-1$	unsigned
long int	Длинные целые	4 байта	$-2^{31}..2^{31}-1$	long
unsigned long	Беззнаковые длинные	4 байта	$0..2^{32}-1$	
short int	Короткие целые	2 байта	$-2^{15}..2^{15}-1$	short
unsigned short	Беззнаковые короткие	2 байта	$0..2^{16}-1$	

При этом в стандарте языка C++ не оговаривается конкретный размер каждого из вышеперечисленных типов, для каждого компилятора они могут быть своими. Приведенные выше числа верны для компиляторов GCC, MS VC++, а вот для Borland C++ версии 3.1 размер переменной типа `int` — 2 байта.

Действительные числа можно записывать в виде десятичных дробей как с фиксированной точкой (например, 3.1415926, 100.001, -10000000.0), так и с плавающей точкой. В последнем случае число имеет вид $\langle f \rangle e \langle p \rangle$, где $\langle f \rangle$ — дробное число (положительное или отрицательное), называемое *мантиссой*, а $\langle p \rangle$ — целое число (положительное или отрицательное), называемое *порядком*. Число, записанное таким образом, равно $f \times 10^p$. Фактически, порядок означает, на какое число позиций нужно сдвинуть вправо десятичную точку в записи числа $\langle f \rangle$. Если же порядок меньше нуля, то сдвиг десятичной точки осуществляется влево. Примеры записи чисел с плавающей точкой:

3.14e1 означает 31.4

3.14e5 означает 314000

3.14e-3 означает 0.00314

-1e6 означает -1000000

-1e-6 означает -0.000001

Для представления действительных чисел существует два стандартных типа:

Имя типа	Размер
float	4 байта
double	8 байт

Как правило, для хранения целых чисел следует использовать тип `int`, а для действительных чисел — `double`.

2.3 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;
```

В последней строке встретился оператор сложения '+'. Кроме оператора сложения, есть еще операторы вычитания '-', умножения '*', деления '/' и взятия остатка от деления целых чисел '%'.

Особого внимания заслуживает оператор деления. Если оба его аргумента имеют целочисленный тип (то есть один из типов, перечисленных в первой таблице или целочисленные константы), то этот оператор рассматривается, как оператор деления целых чисел с остатком. Если же хотя бы один из операторов будет иметь дробный тип, то оператор деления выполняется, как оператор деления десятичных дробей.

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.

2.4 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 '>>'.

2.5 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;
}
```