Объявление переменных

Инструкция вида x = 5 сохраняет значение 5 в переменной x. После такого присвоения можно использовать переменную x вместо константы 5, пока переменная x не изменит своего значения.

Прежде чем использовать в программе новую переменную, ее необходимо *объявить*. Например:

```
int x;
x = 5;
```

или

```
int y;
y = 10;
```

Объявлять переменные можно в любом месте программы, но обязательно перед их использованием.

Объявление разных типов переменных

В C++ каждая переменная может хранить значение только одного типа, т.е. тип уже объявленной переменной изменить нельзя. Причиной тому является большая разница в размерах памяти, необходимой для хранения значений переменных разных типов.

В C++ тип int определяет множество целых чисел. Изученная ранее программа, преобразующая температуру (из Цельсия в Фаренгейты), могла работать только с целыми значениями температуры, производя усечение дробной части температурных значений. При этом компилятор, не выводя никаких предупреждающих сообщений, просто отбрасывает (а не округляет) дробную часть числа.

Ограничения, налагаемые на целые числа

1. Округление до целых значений

Требуется вычислить среднее трех чисел: a, b и c. Среднее значение вычисляется по формуле:

(a + b + c)/3

Например, если a=1; b=2 и c=2, то среднее будет равно $1\frac{2}{3}$ или 1,666... Полученный результат приводится к целому значению, просто отбрасывая его дробную часть. При этом 1,666 утратит свой “дьявольский” остаток и превратится в 1.

Даже если попытаться решить эту задачу иным способом:

a/3 + b/3 + c/3 = 0 + 0 + 0 = 0

2. Ограничения диапазона

Максимальным значением обычной целочисленной переменной является число 2 147 483 647, минимальным – - 2 147 483 647, т.е. общий диапазон (от -2^{32}до $+2^{32}-1$) – около 4 млрд. чисел (зависит от компилятора). Число 2 млрд. может оказаться недостаточным, например, для тактовой частоты, превышающей 2 ГГц (приставка Г – Гига обозначает миллиард).

С++ позволяет объявлять целые числа как беззнаковые, что означает, что они не могут быть отрицательными. Целое число типа unsigned int может принимать значения от 0 до +4 294 967 295. Можно объявить переменную просто как unsigned, опустив объявление int, которое подразумевается неявно.

Вещественные (действительные) числа

Число с плавающей запятой (или число с плавающей точкой) — форма представления вещественных чисел, в которой число хранится в форме мантиссы и показателя степени.

Название «плавающая запятая» происходит от того, что запятая в позиционном представлении числа может быть помещена где угодно относительно цифр в строке. Представление числа в форме с плавающей запятой может рассматриваться как компьютерная реализация экспоненциальной записи чисел.

Число с плавающей запятой состоит из:

- Знака мантиссы (указывающего на отрицательность или положительность числа);
- Мантиссы (выражающей значение числа без учёта порядка);
- Знака порядка;
- Порядка (выражающего степень основания числа, на которое умножается мантисса).

В вычислительных машинах показатель степени принято отделять от мантиссы буквой «E» (exponent). Например, число $1{,}528535047 \cdot 10^{-25}$ в большинстве языков программирования высокого уровня записывается как 1.528535047E-25.

Десятичные числа могут не иметь дробной части, оставаясь при этом действительными (например, действительное число 5.0).

Примеры:

1 000 000 = $1.0 \cdot 10^6$ = $1.0E6$

0,000001 (одна миллионная) = $1.0 \cdot 10^{-6}$=$1.0E-6$

0.00345 = $345.0 \cdot 10^{-5}$ = $34.5 \cdot 10^{-4}$= $3.45E-3$

Решение проблемы с отбрасыванием дробного числа:

Требуется вычислить среднее (d) трех вещественных чисел: a, b и c.

```
double d;
d = 1.0/3.0 + 2.0/3.0 + 2.0/3.0;
```

Это эквивалентно выражению

```
d = 0.333... + 0.666... + 0.666...;
```

Которое дает значение

```
d = 1.666...
```

1,(6) – “один и шесть в периоде”

Количество цифр «6» ограничено пределом переменной типа double.

Ограничения, налагаемые на числа с плавающей точкой

1. Вещественные переменные не могут использоваться для перечисления. C++ требует использовать при перечислении только целые значения.
2. Процессор компьютера выполняет операции с целыми числами гораздо быстрее, чем с действительными. Для сложения 1000 целых чисел процессору может потребоваться столько же времени, сколько для выполнения лишь 200 вычислений с плавающей точкой.
3. Вещественные числа тоже страдают от ошибок округления. Если усреднить числа 1.0, 2.0, 2.0, то получится не математически точное 1,(6), а приблизительное значение 1.666667.

Символьный тип и строка символов

char – символьный тип. Значением переменных может быть символ алфавита, цифра, знак препинания или знак арифметической операции.
Пример:

```
char c;
```

string – строка символов, составляющая предложение.
Пример:

```
string “this is a string”;
```

Логические выражения

C++ предоставляет в распоряжение программиста логический тип bool. Название этого типа происходит от имени Буля, автора символьной логики. Булева переменная может иметь только одно из двух значений – true или false.
Например, выражение “x равно y” может иметь значение true или false.

“true” – это строка

true – это булева переменная

TRUE – это может быть хоть чем (чем объявит программист)

Объявление переменной типа int (или любого другого типа) и ее инициализация:

```
int x;
x = 5;
```

Объявить и инициализировать переменную можно одним оператором:

```
int x = 5;
```

Значениями переменных, используемых в качестве символьных констант, могут быть и непечатаемые символы.

Специальные символы

Символьные константы	Обозначение действий
\n	Новая строка
\t	Табуляция
\\	Обратная косая черта

\n – позволяет разделить строку на две части
\t – перемещает выводимую информацию к следующей позиции табуляции

Выражения смешанного типа

C++ позволяет использовать в одном выражении переменные разных типов. Например, можно складывать целые и вещественные переменные:

```
int x = 5;
double y = x + 1.0;
// в этом выражении перед выполнением операции сложения
// значение x преобразуется к типу double
```

Тип генерируемого в результате значения будет соответствовать более мощному типу операнда.

По тому же принципу выражение одного типа может быть присвоено переменной другого типа, например:

```
double y = 1.0;
int x;
x = y;
// в этом выражении целая часть y сохраняется в x
```

Если переменная в левой стороне равенства относится к типу менее мощному, чем переменная справа, то при таком присвоении можно потерять точность значений.

Преобразование типа большего размера в меньший называется понижающим приведением (demotion), а обратное преобразование – повышающим приведением (promotion).