

Вывод в языке C++ через cout

Использование объекта cout

Для стандартного вывода (вывода на экран) в языке C++ используется объект `cout` и операторы `<<`. Например, для вывода значения переменной `a` можно написать:

```
cout << a;
```

Для вывода значений переменных `a` и `b` через пробел так:

```
cout << a << " " << b;
```

В этом примере мы даем команду сначала вывести значение переменной `a`, затем строку из одного пробела, затем вывести значение переменной `b`.

Для перехода на следующую строку нужно либо вывести специальный символ `'\n'`, либо использовать объект `endl`. Приведем примеры использования обоих способов для вывода значений переменных в отдельных строках:

```
cout << a << "\n" << b << endl;
```

Через `cout` можно выводить все стандартные типы данных - целые и действительные числа, символы, строки (как C-строки, так и объекты `string`), заботиться о задании типов данных при этом не нужно, компилятор автоматически распознает необходимые типы.

Для использования стандартного ввода-вывода в языке C++ необходимо подключить заголовочный файл `iostream`:

```
#include<iostream>
```

Задание ширины поля вывода

Есть два способа управления выводом - через вызовы метода объекта `cout` (например, `cout.width(20)`) или при помощи "манипуляторов" (`cout << setw(10)`).

Второй способ удобней, поэтому рассмотрим его. Для использования манипуляторов необходимо подключить заголовочный файл `iomanip`:

```
#include<iomanip>
```

Для задания ширины поля вывода одного значения используется манипулятор `setw` с одним параметром - шириной поля вывода. Например:

```
cout << setw(10) << a;
```

выведет значение переменной *a* в поле шириной 10 символов. Аналогично правилам вывода в языке C, по умолчанию осуществляется выравнивание выводимого значения по правому краю (перед выводимым значением добавляются пробелы), если же число содержит больше символов, чем ширина поля, то будут выведены все символы.

Манипулятор *setw* влияет только на значение ширины поля вывода для следующего выводимого значения, если нужно задать значение ширины поля вывода для двух чисел, то это нужно делать перед выводом каждого значения.

Манипулятор *setfill* с одним параметром типа *char* позволяет задать символ-"заполнитель", который будет использоваться вместо пробела для заполнения поля. Например, вывести время в формате HH:MM (ровно по две цифры для вывода часов и минут) можно следующим образом:

```
cout << setfill('0') << setw(2) << h << ":" << setw(2) << m << endl;
```

Заметим, что манипулятор *setfill* действует постоянно (до следующего вызова *setfill*), не нужно устанавливать его перед выводом каждого значения.

Также как и в языке C можно задавать тип выравнивания внутри поля. Выравнивание бывает левым, правым и внутренним. Внутреннее выравнивание означает, что сначала выводится знак числа (если число отрицательное), затем - пробелы, затем - цифры числа. Для задания выравнивания используются манипуляторы *left*, *right*, *internal*.

Примеры использования различных типов выравнивания:

Пример кода

```
cout << left << setw(7) << -123 << "*" << endl;  
cout << right << setw(7) << -123 << "*" << endl;  
cout << internal << setw(7) << -123 << "*" << endl;
```

Вывод программы

```
-123 *  
-123*  
- 123*
```

Вывод действительных чисел

Есть два способа вывода действительных чисел - с фиксированной точкой (соответствует форматной строке *"%f"* функции *printf*) и с плавающей точкой (соответствует *"%e"*). Для переключения режима вывода действительных чисел в формат с фиксированной точкой используется манипулятор *fixed*, для вывода с плавающей точкой - манипулятор *scientific*.

По умолчанию числа выводятся с точностью в 6 знаков после точки, это значение можно изменить при помощи манипулятора `setprecision` с параметром - шириной поля вывода.

Например:

```
cout << scientific << setprecision(15) << x << endl;
```

Данные манипуляторы после включения действуют постоянно, до следующего задания значения, то есть их не нужно задавать перед выводом каждой переменной.

По умолчанию (если не задать явно манипулятор `fixed` или `scientific`) число может выводиться или с плавающей точкой, или с фиксированной, в зависимости от величины числа. Если был установлен один из манипуляторов `fixed` или `scientific`, то вернуть "автоматический" выбор формы представления числа можно при помощи манипулятора

```
cout << resetiosflags(ios_base::floatfield);
```