

## Поточный ввод-вывод в C++

Язык C++ / Поточный ввод-вывод в C++

Поточный ввод-вывод в C++ выполняется с помощью функций сторонних библиотек. В C++, как и в C, нет встроенных в язык средств ввода-вывода.

В C для этих целей используется библиотека `stdio.h`.

В C++ разработана новая библиотека ввода-вывода `iostream`, использующая концепцию объектно-ориентированного программирования:

```
#include <iostream>
```

Библиотека `iostream` определяет три стандартных потока:

- ❑ `cin` стандартный входной поток (`stdin` в C)
- ❑ `cout` стандартный выходной поток (`stdout` в C)
- ❑ `cerr` стандартный поток вывода сообщений об ошибках (`stderr` в C)

Для их использования в Microsoft Visual Studio необходимо прописать строку:

```
using namespace std;
```

Для выполнения операций ввода-вывода переопределены две операции поразрядного сдвига:

- ❑ `>>` получить из входного потока
- ❑ `<<` поместить в выходной поток

### Вывод информации

```
cout << значение;
```

Здесь значение преобразуется в последовательность символов и выводится в выходной поток:

```
cout << n;
```

Возможно многократное назначение потоков:

```
cout << 'значение1' << 'значение2' << ... << 'значение n';
```

```
int n;  
char j;  
cin >> n >> j;  
cout << "Значение n равно" << n << "j=" << j;
```

### Ввод информации

```
cin >> идентификатор;
```

При этом из входного потока читается последовательность символов до пробела, затем эта последовательность преобразуется к типу идентификатора, и получаемое значение помещается в идентификатор:

```
int n;  
cin >> n;
```

Возможно многократное назначение потоков:

```
cin >> переменная1 >> переменная2 >>...>> переменнаяn;
```

При наборе данных на клавиатуре значения для такого оператора должны быть разделены символами (пробел, `\n`, `\t`).

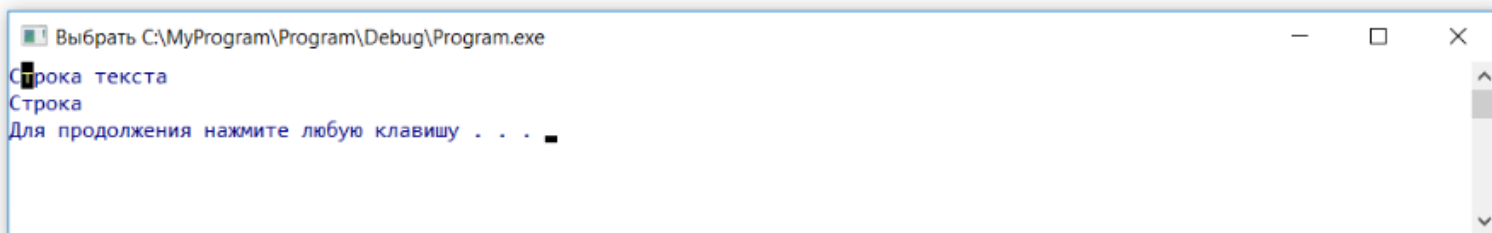
```
int n;  
char j;  
cin >> n >> j;
```

Особого внимания заслуживает ввод символьных строк. По умолчанию потоковый ввод `cin` вводит строку до пробела, символа табуляции или перевода строки.

### Пример

```
1  #include <iostream>  
2  using namespace std;  
3  int main()  
4  {  
5      char s[80];  
6      cin >> s;  
7      cout << s << endl;  
8      system("pause");  
9      return 0;  
10 }
```

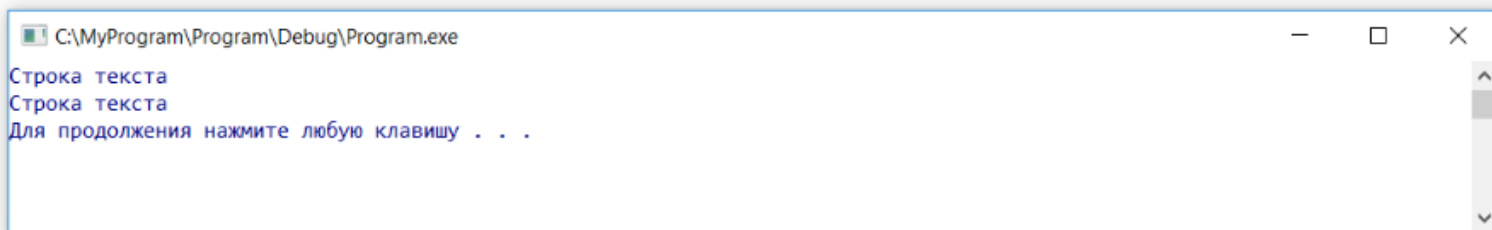
Результат выполнения



Для ввода текста до символа перевода строки используется манипулятор потока `getline()`:

```
1  #include <iostream>  
2  using namespace std;  
3  int main()  
4  {  
5      char s[80];  
6      cin.getline(s, 80);  
7      cout << s << endl;  
8      system("pause");  
9      return 0;  
10 }
```

Результат выполнения



Манипуляторы потока

Функцию - манипулятор потока можно включать в операции помещения в поток и извлечения из потока (<<, >>).

В C++ имеется ряд манипуляторов. Рассмотрим основные:

Манипулятор	Описание
endl	Помещение в выходной поток символа конца строки '\n'
dec	Установка основания 10-ой системы счисления
oct	Установка основания 8-ой системы счисления
hex	Установка основания 16-ой системы счисления
setbase	Вывод базовой системы счисления
width(ширина)	Устанавливает ширину поля вывода
fill('символ')	Заполняет пустые знакоместа значением символа
precision(точность)	Устанавливает количество значащих цифр в числе (или после запятой) в зависимости от использования fixed
fixed	Показывает, что установленная точность относится к количеству знаков после запятой
showpos	Показывает знак + для положительных чисел
scientific	Выводит число в экспоненциальной форме
get()	Ожидает ввода символа
getline(указатель, количество)	Ожидает ввода строки символов. Максимальное количество символов ограничено полем количество

**Пример** Программа ввода-вывода значения переменной в C++

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int n;
6      cout << "Введите n:";
7      cin >> n;
8      cout << "Значение n равно: " << n << endl;
9      cin.get(); cin.get();
10     return 0;
11 }
```

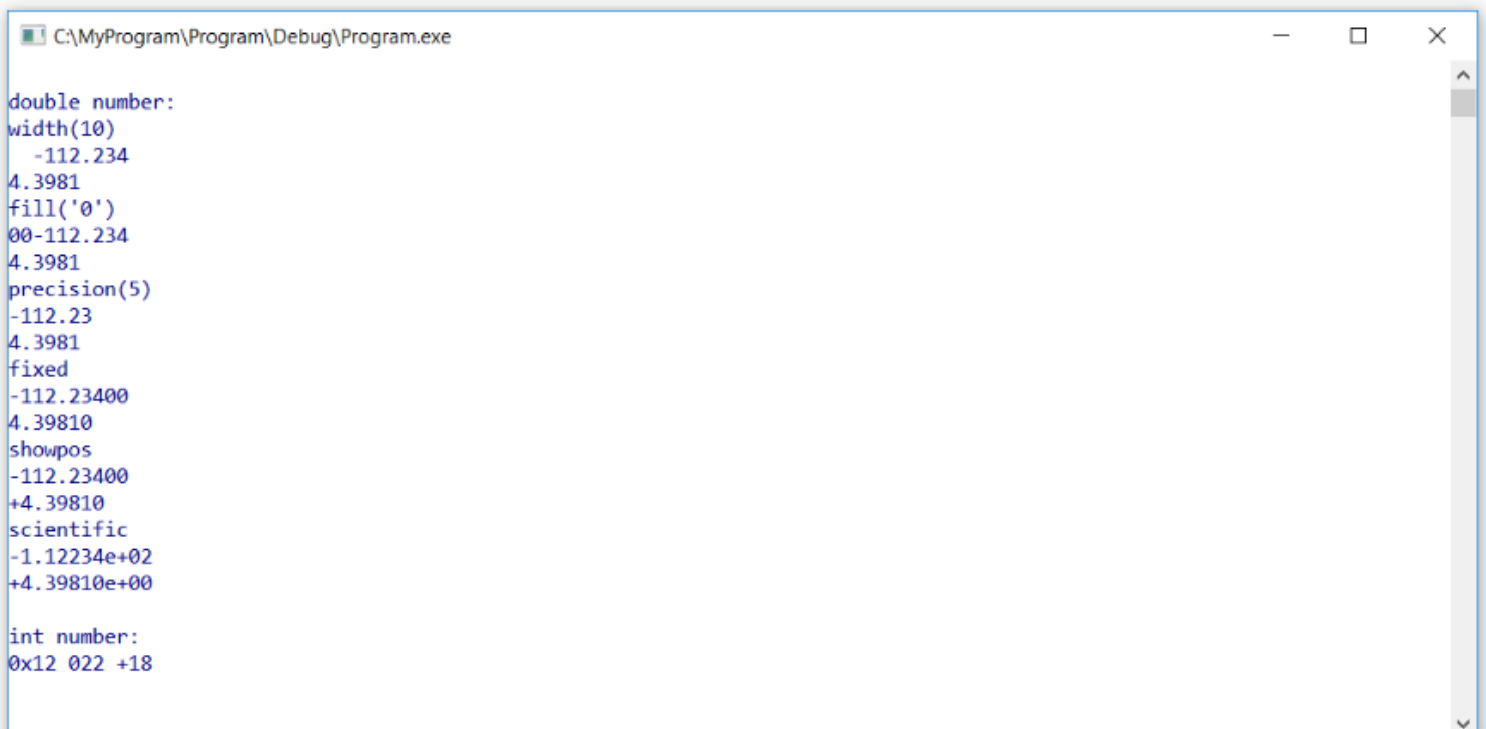
Та же программа, написанная на языке Си

```
1  #include <stdio.h>
2  int main()
3  {
4      int n;
5      printf("Введите n:");
6      scanf("%d", &n);
7      printf("Значение n равно: %d\n", n);
8      getchar(); getchar();
9      return 0;
10 }
```

### Пример Использование форматированного вывода

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      double a = -112.234;
6      double b = 4.3981;
7      int c = 18;
8      cout << endl << "double number:" << endl;
9      cout << "width(10)" << endl;
10     cout.width(10);
11     cout << a << endl << b << endl;
12     cout << "fill('0')" << endl;
13     cout.fill('0');
14     cout.width(10);
15     cout << a << endl << b << endl;
16     cout.precision(5);
17     cout << "precision(5)" << endl << a << endl << b << endl;
18     cout << "fixed" << endl << fixed << a << endl << b << endl;
19     cout << "showpos" << endl << showpos << a << endl << b << endl;
20     cout << "scientific" << endl << scientific << a << endl << b << endl;
21     cout << endl << "int number:" << endl;
22     cout << showbase << hex << c << " " << showbase << oct << c << " ";
23     cout << showbase << dec << c << endl;
24     cin.get();
25     return 0;
26 }
```

### Результат выполнения



```
C:\MyProgram\Program\Debug\Program.exe

double number:
width(10)
-112.234
4.3981
fill('0')
00-112.234
4.3981
precision(5)
-112.23
4.3981
fixed
-112.23400
4.39810
showpos
-112.23400
+4.39810
scientific
-1.12234e+02
+4.39810e+00

int number:
0x12 022 +18
```