

🏠 (<http://cppstudio.com>) / Язык программирования C++ (<http://cppstudio.com/cat/274/>)
/ Основы языка программирования C++ (<http://cppstudio.com/cat/274/275/>)
/ Форматированный ввод/вывод в C++

Форматированный ввод/вывод в C++



Оценка: **4,25** (голосов: 8)

Чтобы проголосовать, вы должны зарегистрироваться.

Возможность управлять вводом-выводом в C++, обеспечивают форматирующие функции-члены, флаги и манипуляторы. Флаги, функции и манипуляторы выполняют одну и ту же задачу — задают определённый формат ввода/вывода информации в потоках. Ввод/вывод на экран/с экрана в C++ осуществляется с помощью операторов `cin` и `cout` соответственно, а значит манипуляторы форматирования используются совместно с данными операторами ввода/вывода. Различие между функциями, флагами и манипуляторами форматирования состоит в способе их применения. Теперь рассмотрим способы применения объектов форматирования.

```
1 //Основные форматирующие функции-члены:
2 cout.fill('*symbol*'); // устанавливает символ заполнитель
3 // где symbol - символ заполнитель, символ передаётся в одинарных кавычках
4
5 cout.width(*width_field*); // задает ширину поля
6 // где width_field - количество позиций(одна позиция вмещает один символ)
7
8 cout.precision(*number*); // задает количество знаков после десятичной точки
9 // где number - количество знаков после десятичной точки
```

Доступ к функциям осуществляется через операцию **точка**, а в круглых скобках передаётся аргумент. Аргумент функции `fill()` может передаваться в виде символа, обрамленного одинарными кавычками, или в виде числа(код символа). Одних функций не достаточно для форматирования потоков ввода/вывода, поэтому в C++ предусмотрен ещё один способ форматирования — флаги.

Флаги форматирования позволяют включить или выключить один из параметров ввода/вывода. Чтобы установить флаг ввода/вывода, необходимо вызвать функцию `setf()`, если необходимо отключить флаг вывода, то используется функция `unsetf()`. Далее показаны конструкции установки и снятия флагов вывода.

```
1 // установка флага вывода
2 cout.setf( ios::*name_flag* );
3 // где name_flag - это имя флага
```

Доступ к функциям оператора вывода выполняется через операцию **точка**. Метод `setf()` принимает один аргумент — имя флага. Флаги вывода объявлены в классе `ios`, поэтому, перед тем, как обратиться к флагу, необходимо написать имя класса — `ios`, после которого, с помощью операции разрешения области действия, вызвать нужный флаг.

```
1 // снятие флага вывода
2 cout.unsetf( ios::*name_flag* );
3 // где name_flag - это имя флага
```

Если при вводе/выводе необходимо установить(снять) несколько флагов, то можно воспользоваться поразрядной логической операцией (http://cppstudio.com/obuchenie_cpp/porazradnye-logicheskie-operatsii) или `|`. В этом случае конструкция языка C++ будет такой:

```

1 // установка нескольких флагов
2 cout.setf( ios::/*name_flag1*/ | ios::/*name_flag2*/ | ios::/*name_flag_n*/ );
3
4 // снятие нескольких флагов
5 cout.unsetf( ios::/*name_flag1*/ | ios::/*name_flag2*/ | ios::/*name_flag_n*/ );

```

В таблице 1 подробно описаны основные флаги форматирования, а также показаны примеры использования флагов.

Таблица 1 — Флаги форматирования в C++

Флаг	Назначение	Пример	Результат
boolalpha	Вывод логических величин в текстовом виде (true , false)	<code>cout.setf(ios::boolalpha); bool log_false = 0, log_true = 1; cout << log_false << endl << log_true << endl;</code>	false true
oct	Ввод/вывод величин в восьмеричной системе счисления (сначала снимаем флаг dec , затем устанавливаем флаг oct)	<code>cout.unsetf(ios::dec); cout.setf(ios::oct); int value; cin >> value; cout << value << endl;</code>	ВВОД:99 ₁₀ ВЫВОД:143 ₈
dec	Ввод/вывод величин в десятичной системе счисления (флаг установлен по умолчанию)	<code>cout.setf(ios::dec); int value = 148; cout << value << endl;</code>	148
hex	Ввод/вывод величин в шестнадцатеричной системе счисления (сначала снимаем флаг dec , затем устанавливаем флаг hex)	<code>cout.unsetf(ios::dec); cout.setf(ios::hex); int value; cin >> value; cout << value << endl;</code>	ВВОД:99 ₁₀ ВЫВОД:63 ₁₆
showbase	Выводить индикатор основания системы счисления	<code>cout.unsetf(ios::dec); cout.setf(ios::oct ios::showbase); int value; cin >> value; cout << value << endl;</code>	ВВОД:99 ₁₀ ВЫВОД:0143 ₈
uppercase	В шестнадцатеричной системе счисления использовать буквы верхнего регистра(по умолчанию установлены буквы нижнего регистра)	<code>cout.unsetf(ios::dec); cout.setf(ios::hex ios::uppercase); int value; cin >> value; cout << value << endl;</code>	ВВОД:255 ₁₀ ВЫВОД:FF ₁₆
showpos	Вывод знака плюс + для положительных чисел	<code>cout.setf(ios::showpos); int value = 15; cout << value << endl;</code>	+15

Флаг	Назначение	Пример	Результат
scientific	Вывод чисел с плавающей точкой в экспоненциальной форме	<code>cout.setf(ios::scientific);</code> <code>double value = 1024.165;</code> <code>cout << value << endl;</code>	1.024165e+003
fixed	Вывод чисел с плавающей точкой в фиксированной форме(по умолчанию)	<code>double value = 1024.165;</code> <code>cout << value << endl;</code>	1024.165
right	Выравнивание по правой границе(по умолчанию). Сначала необходимо установить ширину поля(ширина поля должна быть заведомо большей чем, длина выводимой строки).	<code>cout.width(40);</code> <code>cout <<</code> <code>«cppstudio.com» <<</code> <code>endl;</code>	__cppstudio.com
left	Выравнивание по левой границе. Сначала необходимо установить ширину поля(ширина поля должна быть заведомо большей чем, длина выводимой строки).	<code>cout.setf(ios::left);</code> <code>cout.width(40);</code> <code>cout <<</code> <code>«cppstudio.com» <<</code> <code>endl;</code>	cppstudio.com__

Ещё один способ форматирования — форматирование с помощью манипуляторов. Манипулятор — объект особого типа, который управляет потоками ввода/вывода, для форматирования передаваемой в потоки информации. Отчасти манипуляторы дополняют функционал, для форматирования ввода/вывода. Но большинство манипуляторов выполняют точно, то же самое, что и функции с флагами форматирования. Есть случаи, когда проще пользоваться флагами или функциями форматирования, а иногда удобнее использовать манипуляторы форматирования. Именно по этому в C++ предусмотрено несколько средств форматирования ввода/вывода. В таблице 2 показаны основные манипуляторы форматирования C++.

Таблица 2 — Манипуляторы форматирования в C++

Манипулятор	Назначение	Пример	Результат
endl	Переход на новую строку при выводе	<code>cout <<</code> <code>«website:» <<</code> <code>endl <<</code> <code>«cppstudio.com»;</code>	website: cppstudio.com
boolalpha	Вывод логических величин в текстовом виде (true , false)	<code>bool log_true =</code> <code>1;</code> <code>cout << boolalpha</code> <code><< log_true <<</code> <code>endl;</code>	true
noboolalpha	Вывод логических величин в числовом виде (true , false)	<code>bool log_true =</code> <code>true;</code> <code>cout <<</code> <code>noboolalpha <<</code> <code>log_true << endl;</code>	1

Манипулятор	Назначение	Пример	Результат
oct	Вывод величин в восьмеричной системе счисления	<code>int value = 64; cout << oct << value << endl;</code>	100₈
dec	Вывод величин в десятичной системе счисления (по умолчанию)	<code>int value = 64; cout << dec << value << endl;</code>	64₁₀
hex	Вывод величин в шестнадцатеричной системе счисления	<code>int value = 64; cout << hex << value << endl;</code>	40₈
showbase	Выводить индикатор основания системы счисления	<code>int value = 64; cout << showbase << hex << value << endl;</code>	0x40
noshowbase	Не выводить индикатор основания системы счисления (по умолчанию).	<code>int value = 64; cout << noshowbase << hex << value << endl;</code>	40
uppercase	В шестнадцатеричной системе счисления использовать буквы верхнего регистра (по умолчанию установлены буквы нижнего регистра).	<code>int value = 255; cout << uppercase << hex << value << endl;</code>	FF₁₆
nouppercase	В шестнадцатеричной системе счисления использовать буквы нижнего регистра (по умолчанию).	<code>int value = 255; cout << nouppercase << hex << value << endl;</code>	ff₁₆
showpos	Вывод знака плюс + для положительных чисел	<code>int value = 255; cout << showpos << value << endl;</code>	+255
noshowpos	Не выводить знак плюс + для положительных чисел (по умолчанию).	<code>int value = 255; cout << noshowpos << value << endl;</code>	255
scientific	Вывод чисел с плавающей точкой в экспоненциальной форме	<code>double value = 1024.165; cout << scientific << value << endl;</code>	1.024165e+003

Манипулятор	Назначение	Пример	Результат
fixed	Вывод чисел с плавающей точкой в фиксированной форме (по умолчанию).	<pre>double value = 1024.165; cout << fixed << value << endl;</pre>	1024.165
setw(int number)	Установить ширину поля, где number — количество позиций, символов (выравнивание по умолчанию по правой границе). Манипулятор с параметром.	<pre>cout << setw(40) << «cppstudio.com» << endl;</pre>	__cppstudio.com
right	Выравнивание по правой границе(по умолчанию). Сначала необходимо установить ширину поля(ширина поля должна быть заведомо большей чем, длина выводимой строки).	<pre>cout << setw(40) << right << «cppstudio.com» << endl;</pre>	__cppstudio.com
left	Выравнивание по левой границе. Сначала необходимо установить ширину поля(ширина поля должна быть заведомо большей чем, длина выводимой строки).	<pre>cout << setw(40) << left << «cppstudio.com» << endl;</pre>	cppstudio.com__
setprecision(int count)	Задаёт количество знаков после запятой, где count — количество знаков после десятичной точки	<pre>cout << fixed << setprecision(3) << (13.5 / 2) << endl;</pre>	6.750
setfill(int symbol)	Установить символ заполнитель. Если ширина поля больше, чем выводимая величина, то свободные места поля будут наполняться символом symbol — символ заполнитель	<pre>cout << setfill('0') << setw(4) << 15 << ends << endl;</pre>	0015

Форматированный ввод/вывод в C++ это одна из самых простых тем в программировании. Как использовать те или иные средства форматирования показано в таблицах, поэтому затруднений по данной теме возникнуть не должно.

Практика



- 🔗 Манипуляции с битами (<http://cppstudio.com/post/1539/>) 🔥🔥🔥🔥🔥
- 🔗 Перевод в верхний регистр (<http://cppstudio.com/post/4615/>) 🔥🔥🔥🔥🔥
- 🔗 Рамка на титул (<http://cppstudio.com/post/1395/>) 🔥🔥🔥🔥🔥
- 🔗 Сумма чисел нацело делящихся на 5 (<http://cppstudio.com/post/1360/>) 🔥🔥🔥🔥🔥
- 🔗 Шесть равнобедренных треугольников (<http://cppstudio.com/post/6611/>) 🔥🔥🔥🔥🔥

💬 Обсудить на форуме (/topics/)

👤 Автор: admin (/forums/users/admin/)

📅 Дата: 25.08.2012

🔗 Поделиться:

[← Явное и неявное преобразование типов данных C++ \(http://cppstudio.com/post/310/\)](#)[Генератор случайных чисел rand\(\) в C++ → \(http://cppstudio.com/post/339/\)](#)**Похожие статьи:**

1. **Операции инкремента и декремента в C++** (<http://cppstudio.com/post/282/>)
2. **Ссылки в C++** (<http://cppstudio.com/post/429/>)
3. **Операции присваивания в C++** (<http://cppstudio.com/post/279/>)
4. **Наследование классов** (<http://cppstudio.com/post/10103/>)
5. **Функция fprintf** (<http://cppstudio.com/post/1270/>)

14 комментариев

Ваш комментарий...

Отправить

**Миша Карпов**

до подключения библиотеки `#include "iomanip"` очень ругалась программа на строчку `cout << setfill("*") << setw(4) << 15 << ends << endl;`
а именно на `"setfill(*)"` и `:"setw"`
если библиотека необходима для работы с манипуляторами мб надо это написать в статье или я что то путаю?

7 июн в 14:00 [Комментировать](#)

5

**Григорий Васильев**

Скажите, как однажды применив настройку, использовать ее во всех последующих выводах? Скажем, есть у меня цикл, который в теле содержит операцию вывода. Если описать настройку форматирования вывода вне цикла - она используется при первом выводе, затем все по умолчанию

18 авг 2016

1

**Михаил Кривостовский**

к сожалению, насколько мне известно в языке C++ такое не предусмотрено, т.к такие функции являются не залипающими, поэтому их постоянно нужно прописывать. Есть только один выход постараться написать свой класс, включив в него метод алгоритма "вывода" при заданных параметрах. Но это надо хорошо постараться.

22 янв в 19:32 [Ответить](#)

Написать комментарий...

**Ден Ацыдов**

как то про ввод забыли уточнить(

15 июл 2016

1

**Егор Сергеев**

Вряд ли чем-то отличается от вывода. Вместо `cout` `cin` и все

18 июл 2016 [Ответить](#)

**Ден Ацыдов**

Егор, попробуйте строку с пробелами считать руководствуясь этой статьёй)

19 июл 2016 [Ответить](#)**Александр Домашенко** ответил Дену

Ден,
string str;
getline(cin, str);
либо
char str[50];
cin.getline(str, 50);

8 авг 2016 [Ответить](#)**Вітьок Михейкін**

Як вивести числа в таблиці посередині і так щоб знак були поруч, та десяті аїд десятими соті під сотими цілі під цілими відповідно по своїх розрадах(Не лише по середині)

25 окт 2015

[Показать все 4 комментария](#)**Evil Spirit**

Для начала говорит на человеческом языке.

8 мар 2016 [Ответить](#)**Дмитрий Браницкий**манипулятор setw - ровняет по правому боку (почитай о нем).
удобно, для вывода табличных данных.

*иногда может некорректно работать с кириллицей

30 мар 2016 [Ответить](#)**Владислав Точаненко** ответил EvilEvil, хватит тут политики. Этот сайт создан для изучения C++, а не оскорбления других людей.
Я би і сам сказав, що твоя мова потворна, проте не можу, бо сам нею розмовляю22 мая 2016 [Ответить](#)

1

**Михаил Рягузов**

```
cout.precision(*number*);
```

 // задает количество знаков после десятичной точки
у меня когда я это ввожу то введенный номер это общее количество знаков, а не только после запятой.15 окт 2015 [Комментировать](#)**Виталий Малий**

это альтернатива printf ? или как ?

24 сен 2015

1

[Показать все 5 комментариев](#)**Андрей Иванилов** ответил Дмитрию

Дмитрий, у меня ехешники весят при конфигурации релиза 8кб с cin или же 7кб с printf

20 ноя 2015 [Ответить](#)



Валерий Бас ответил Дмитрию

Дмитрий, 15кб, не ну это конечно чревато серьезными проблемами со свободным местом на жестком диске.

25 янв 2016 [Ответить](#)

4



Александр Штерн

Для правнуков: В `iostream` можно очень сильно ускорить ввод-вывод убрав синхронизацию `cstdio` (`"ios_base::sync_with_stdio(false);"`). В этом случае скорость становится почти одинаковой.

Также вместо `endl` следует просто писать `"\n"`. Суть в том что `endl` перегружен всякими синхронизациями и прочим для безопасного вывода, а это нам не надо, так как мы не используем параллельно `cstdio`.

Некоторые просто проставляют `define endl "\n"`, но это уже дурной тон.

В олимпиадном программировании кстати делают именно так. Там оч важна скорость

16 мая в 17:54 [Ответить](#)

Написать комментарий...



Богдан Бусь

нихера не понял

3 сен 2015

3



Слава Моисеев

+1

24 сен 2015 [Ответить](#)



Леонид Смирнов

-1

24 сен 2015 [Ответить](#)

2



Алексей Иванов

всё очень просто, чтобы понять лучше советую изучить ООП

25 сен 2015 [Ответить](#)

6

Написать комментарий...



Георгий Петров

"в каких случаях использовать манипуляторы? флаги? проще ведь написать `cout<<showpos...` чем прописывать `cout.setf(ios::showpos)`, кроме синтаксиса они чем нибудь отличаются в применении?

18 июн 2015



Алексей Иванов

я в начале тоже так думал, проще манипулятор, но потом подумал и сделал вывод - лучше флаги, они полностью формируют ввод-вывод, один раз установишь флаг и все последующие будут форматированными выводиться, исключение только ширина поля, а манипуляторы постоянно надо прописывать.

29 авг 2015 [Ответить](#)



Снежана Синега ответила Алексею

Алексей, спасибо. Просто идеально

1 фев 2016 [Ответить](#)

Написать комментарий...

**Иван Жук**

setw(int number), setprecision(int count), setfill(int symbol) не работают почему-то. "cout « setw(40) « «cppstudio.com» « endl;" при добавлении строки пишет: main.cpp(14): error C3861: setw: идентификатор не найден

9 апр 2015

**Паша Лесников**

int там нафиг не нужен.

setw(5) - инт тут имеется в виду, что в скобках должно быть целочисленное значение.

23 апр 2015 [Ответить](#)**Артём Остапов**

не забываем подключить <iomanip>

25 апр 2015 [Ответить](#)**Иван Жуков**

А так же в noshowpos в примере написано showpos

9 мар 2015 [Комментировать](#)[Показать ещё](#)

Комментарии

**zybr zybr** (<https://plus.google.com/104817200868387653174>)

06.03.2016 (/post/319/comment-page-1/#comment-3087)

хорошая статья!

Войдите, чтобы ответить (<http://cppstudio.com/wp-login.php?>

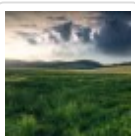
redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F)

**Kto To**(https://www.facebook.com/app_scoped_user_id/67333123947351)

21.06.2016 (/post/319/comment-page-1/#comment-3180)

Войдите, чтобы ответить (<http://cppstudio.com/wp-login.php?>

redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F)

**rajul1431**

24.07.2015 (/post/319/comment-page-1/#comment-2833)

Здравствуйте!)

Извините за такой вопрос, когда я написал этот код для решения одной из задач:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int a;
8      cout << "Vvedite chislo" << endl;
9      cin >> a;
10     cout << hex << "chislo v shesnadceterichnom vide: " << a << endl;
11
12     return 0;
13 }

```

И когда вместо числа написал 55, в ответе было 37,

А когда исправил на этот:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int a;
8      cout << "Vvedite chislo" << endl;
9      cout.setf(ios::showbase);
10     cin >> a;
11     cout << hex << "chislo v shesnadceterichnom vide: " << a << endl;
12
13     return 0;
14 }

```

то ответ был 0x37, какая между ними разница?

Компилятор Code::Blocks.

Войдите, чтобы ответить (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F)



n196184611

22.01.2015 (/post/319/comment-page-1/#comment-2332)

Почему установку/снятие нескольких флагов нужно разделять поразрядной ИЛИ (|) — просто такой принятый синтаксис, или как-то связано с битовыми вычислениями?

(чтобы лучше понять и запомнить)

Войдите, чтобы ответить (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F)



Chi

30.10.2014 (/post/319/comment-page-1/#comment-2197)

Учусь. Узнал что-то новое. Спасибо большое.

В таблице «Манипуляторы» `setfill(int symbol)` неправильно указан тип параметра <http://www.cplusplus.com/reference/iostream/setfill/> (<http://www.cplusplus.com/reference/iostream/setfill/>)

Войдите, чтобы ответить (http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F)

**1lena**

10.04.2014 (/post/319/comment-page-1/#comment-1478)

День добрый. Почему в примере с флагами, флаг fixed записан так:

double value = 1024.165;**cout << value << endl;**

а не так

cout.setf(ios::fixed);

double value = 1024.165;**cout << value << endl;**

и соответственно флаг right?

Войдите, чтобы ответить ([http://cppstudio.com/wp-login.php?](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F)[redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F))**NaikoN**

05.09.2013 (/post/319/comment-page-1/#comment-484)

В примере с манипулятором **hex**, в строке: **cout << dec << value << endl;**, нужно **dec** заменить на **hex** и тогда все будет правильно, т. е. **cout << hex << value << endl;**Войдите, чтобы ответить ([http://cppstudio.com/wp-login.php?](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F)[redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F))**admin** (<http://twitter.com/ShadowSport>)

18.11.2013 (/post/319/comment-page-1/#comment-717)

Сделал

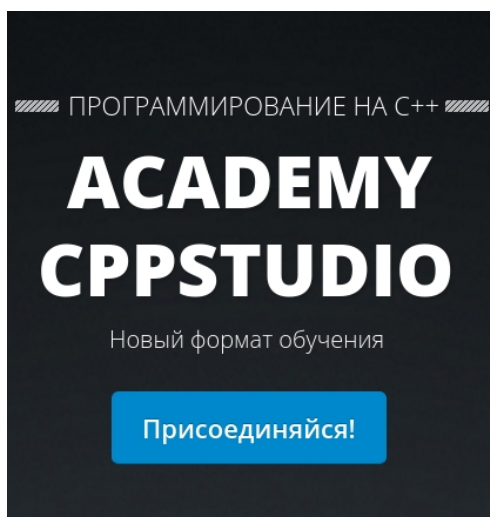
Войдите, чтобы ответить ([http://cppstudio.com/wp-login.php?](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F)[redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F))

Оставить комментарий

Вы должны войти ([http://cppstudio.com/wp-login.php?](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F)[redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F](http://cppstudio.com/wp-login.php?redirect_to=http%3A%2F%2Fcppstudio.com%2Fpost%2F319%2F)), чтобы оставить комментарий.

Поиск...

Поиск



(<http://academy.cppstudio.com/>)

Translation

 (/post/319/)Русский (/post/319/)
 (/uk/post/319/)Українська (/uk/post/319/)
 (/en/post/319/)English (/en/post/319/)
 (/de/post/319/)Deutsch (/de/post/319/)
 (/be/post/319/)Беларуская (/be/post/319/)
 (/kk/post/319/)Қазақ тілі (/kk/post/319/)
 (/uz/post/319/)O'zbek tili (/uz/post/319/)
 (/tr/post/319/)Türkçe (/tr/post/319/)

Новое

➤ Особенности Qt: слоты и сигналы, описание QObject и QApplication, виды окон и т.д.
(<http://cppstudio.com/post/11167/>)

➤ Первая программа на Qt:
(<http://cppstudio.com/post/11127/>)

➤ Введение - графическая библиотека Qt
(<http://cppstudio.com/post/11097/>)

➤ Наследование классов
(<http://cppstudio.com/post/10103/>)

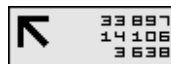
➤ Перегрузка операторов в C++ (часть 2)
(<http://cppstudio.com/post/10058/>)

Популярное

Оптимизировать программу
(<http://cppstudio.com/post/8327/>)



(<https://plus.google.com/u/0/106109650739084338784>)



(<http://www.liveinternet.ru/click>)

(<http://orphus.ru>)