STEP IT ACADEMY

Task #01

C++ Simple Input-Output

LEARN. GROW. SUCCEED.

by Viktor Ivanchenko / ivanvikvik@gmail.com / Minsk

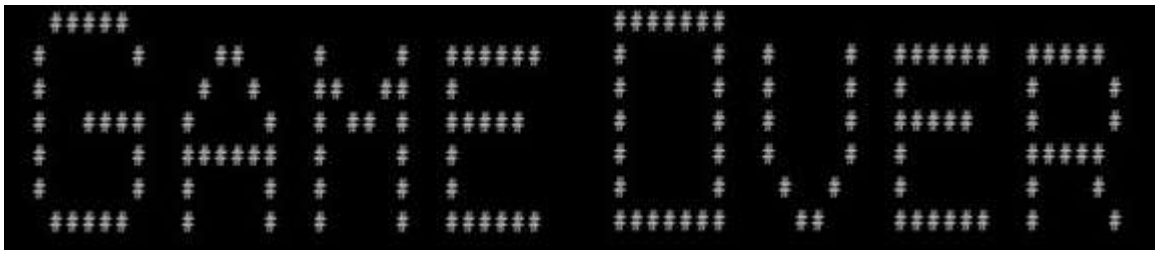
Основы ввода-вывода в языке C++

Основное задание

1) Для закрепления написания простейших программ с использованием языка программирования C++ попробуйте создать простенькие программы-конверторы для различных шкал температур (из градусов Цельсия в градусы Фаренгейта или Кельвина и наоборот) или для различных валют (к примеру, из бел. руб. в евро или наоборот). Можно использовать любую предметную область для создания однотипных приложений (к примеру, конвертор значений углов из градусы в радианы и наоборот).
2) Масса динозавра задаётся в граммах (масса задаётся целым неотрицательным числом). Разработайте программу, которая вычисляет, сколько это килограммов, центнеров и тонн.
3) Дан общий размер файла в байтах (размер задаётся в виде целого неотрицательного числа). Разработайте программу, которая вычисляет, сколько это килобайтов, мегабайтов и т.д.
4) Значение расстояния между двумя городами задаётся в сантиметрах. Разработайте программу, которая вычисляет, сколько это километров и метров.
5) Разработайте программу нахождения периметра и площади квадрата с заданной стороной a.
6) Разработайте программу нахождения периметра и площади прямоугольника с заданными сторонами a и b.
7) Разработайте программу нахождения среднего арифметического и среднего геометрического двух неотрицательных чисел a и b.
8) Разработайте программу нахождения максимального и минимального значения из двух чисел a и b.
9) Разработать программу решения линейного уравнение, заданного в виде Ax + B = 0 (коэффициент A не равен 0).
10) Попробуйте разработать программу, которая меняет местами содержимое двух переменных *a* и *b* (приведите несколько вариантов обмена значения местами).

Дополнительное задание

Необходимо разработать программу «Game Over» на языке C++, которая выводит соответствующую злостную (суровую) надпись на экран монитора с внушительным видом. К примеру:

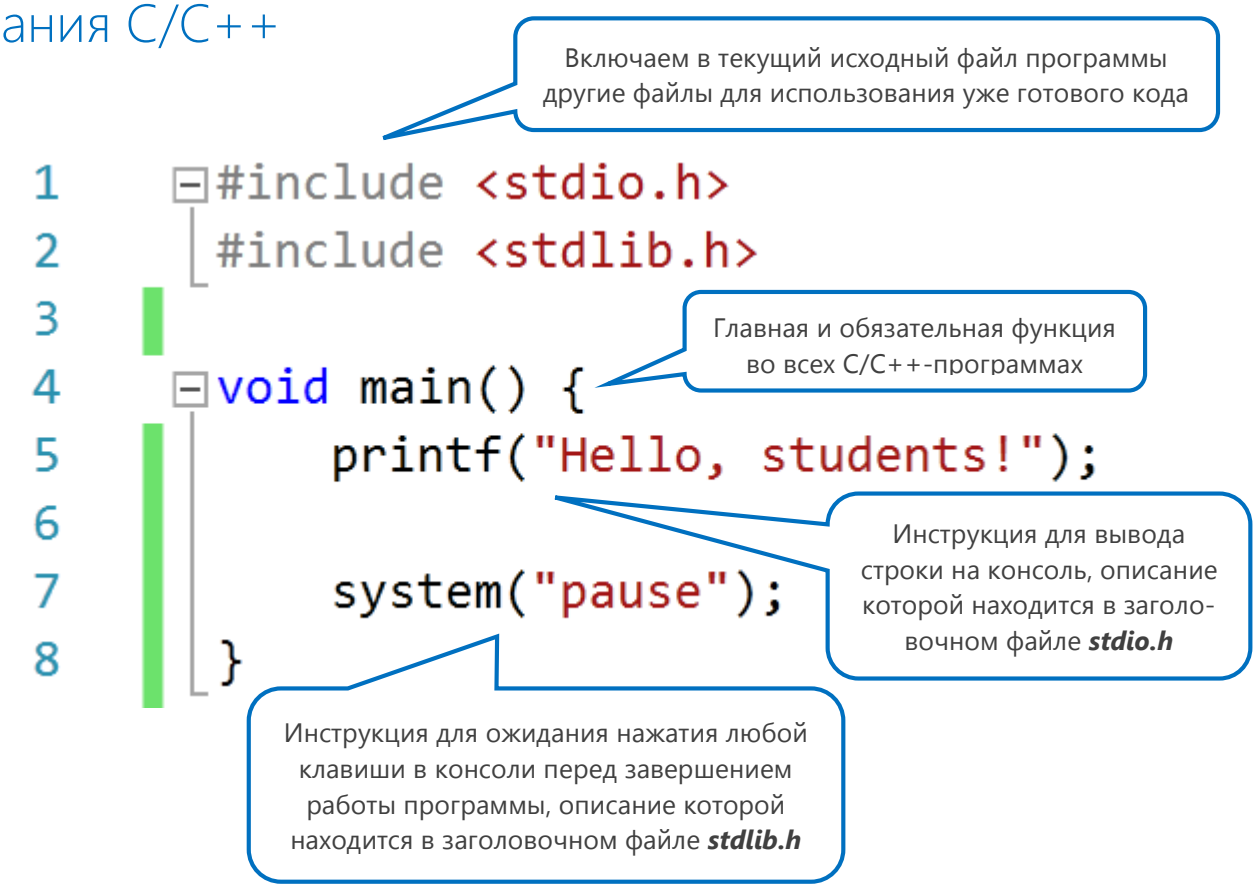


Требования

1) Каждое задание должно быть в отдельном исходном файле с расширением *.cpp.
2) Считать, что пользователь всегда вводит верные (адекватные) данные.
3) При разработке программ придерживайтесь соглашений по написанию кода на C++ (C++ Code Convention).

Best of LUCK with it, and remember to HAVE FUN while you're learning :)
Victor Ivanchenko

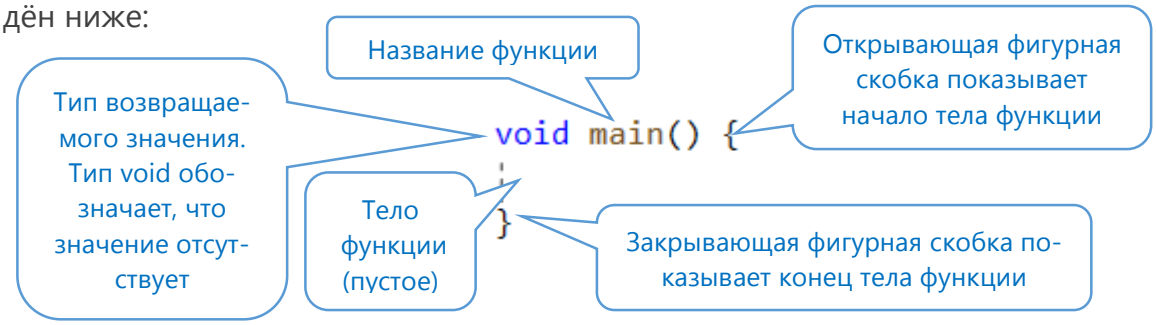
Пример первой программы на языке программирования C/C++



Что нужно запомнить (краткие тезисы)

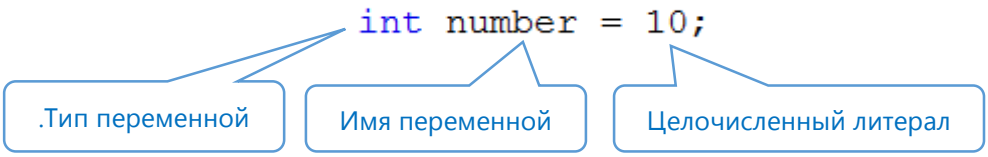
Перед написанием простейшего C++-приложения, очень важно вспомнить (запомнить) следующие вещи о C++:

1. В C++ весь код пишется в пользовательских функциях (и классах в будущем).
2. Название имени функции лучше начинать с соответствующего глагола (чтобы при чтении кода было понятно, что выполняет данная функция) и начинаться со строчной (маленькой) буквы!
3. В любой консольной запускаемой программе всегда должна быть стартовая функция ***main(...)***, которая на вход обычно ничего не принимает и ничего не возвращает. Пример общего синтаксиса стартовой функции ***main(...)*** приведён ниже:



4. В одном файле исходного кода (т.е. в модуле компиляции) на языке C++ можно описать сколько угодно функций, но только одна из них может быть функцией ***main(...)***.
5. Чтобы в процессе выполнения программы можно было хранить начальные, промежуточные и результирующие значения (данные) в языках программирования используются переменные.
6. В общем случае, ***переменная*** (*variable*) – поименованная область памяти, которую можно использовать для хранения данных и осуществления доступа к ним. Данные, которые находятся в переменной (т.е. по данному адресу памяти), называют ***значением*** (*value*) данной переменной.
7. Переменные упрощают написание кода программы, делают этот код читабельным и легко поддерживаемым.
8. ***Объявление переменной*** – это определение её типа и имени.
9. Для установления соответствующего значения переменной используется самый востребованный во всех языках программирования ***оператор присваивания***. В языке C++ он обозначается символом '=' («равно»).

10. В языке C++ можно объявить локальные и глобальные переменные.
11. ***Локальная переменная*** объявляется внутри тела функции.
12. ***Глобальная переменная*** объявляется вне тела функции и лучше их не использовать (согласно соглашению по разработке кода среди разработчиков).
13. Перед использование любой локальной переменной её необходимо объявить, затем инициализировать, а лишь потом только использовать. Нельзя использовать неинициализированную локальную переменную – будет ошибка синтаксиса языка (ошибка компиляции).
14. Объявление локальной переменной заключается в определении её типа и названии идентификатора.
15. Объявление типа локальной переменной необходимо для выделения соответствующего количества памяти, которая будет ассоциироваться с данной переменной и где будут хранится сами данные, а также для определения того, какие операции могут быть использованы с данной переменной.
16. Пример объявления целочисленной переменной в C++:



17. ***Литералом*** в языках программирования называется фиксированное значение, которое непосредственно используется при написании программного кода. Выделяют целочисленные, вещественные, булевские, символьные и строковые литералы, а также *null*-литерал (пустой литерал).
18. Все ***целочисленные литералы*** в языке C++ по умолчанию имеют тип ***int*** или тип ***long long***, которое зависит от значения литерала.
19. Целые числа-литералы могут отображать значения в четырёх системах счисления: в двоичной, в восьмеричной, в десятичной (по умолчанию) и в шестнадцатеричной. Пример представления значения литерала 36 в различных системах счисления:

```
int decimal = 36;
int octal = 044;
int hexadecimal = 0x24;
int binary = 0b100100;
```

20. Целочисленный литерал в двоичной системе счисления начинается с символов ***0b***, а затем идёт само число, состоящее только из цифр *0* и *1*.
21. Целочисленный литерал в восьмеричной системе счисления начинается с символа ***0*** (ноль), а затем идёт само число, состоящее только из цифр *0, 1, 2, 3, 4, 5, 6 и 7*.
22. Целочисленный литерал в десятичной системе счисления записывается обычным числом, где используются все цифры от *0* до *9*.
23. Целочисленный литерал в шестнадцатеричной системе счисления начинается с символа ***0x***, а затем идёт само число, состоящее из цифр *0* до *9* и первых шести букв латинского алфавита: *A*, *B*, *C*, *D*, *E* и *F*.
24. Все ***вещественные литералы*** в языке Java по умолчанию имеют тип ***double*** и две нотации написания: простую и научную (*scientific notation*).
25. Чтобы вещественный литерал имел тип ***float***, необходимо в конце данного литерала дописать символ '***f***' ('***F***').
26. Пример представления значений вещественных литералов 0.0005 и 12345678.9 в различных нотациях:

```
double d1 = 0.0005;
double d2 = 12345678.9;
```

Стандартная форма вещественного литерала

```
double d1 = 5.0e-4;
double d2 = 1.23456789e+7;
```

Научная форма вещественного литерала

27. Все ***символьные литералы*** берутся в одинарные кавычки и имеют тип ***char***.
28. Символьные литералы имеют несколько нотаций для отображения символов. Пример, в котором в различных нотациях представляется символ доллара:

```
char c1 = '$';
char c2 = '\u0024';
char c3 = '\044';
char c4 = 36;
```

Стандартная форма символьного литерала

Unicode-форма символьного литерала, где символ задаётся четырёхзначным числом в 16-ой системе счисления

Форма символьного литерала, где символ задаётся трёхзначным числом в 8-ой системе счисления

Задание символа с помощью его целочисленного эквивалента

29. Все ***строковые литералы*** берутся в двойные кавычки.

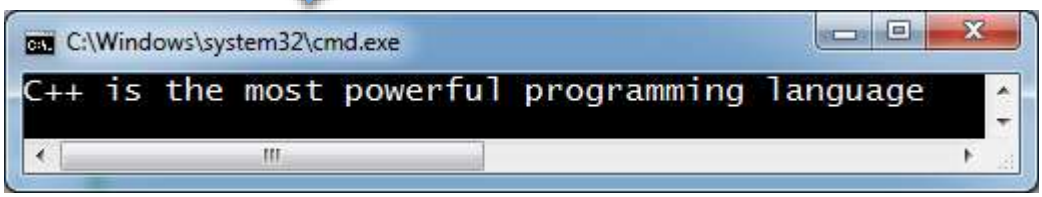
30. В языке C++ есть также два ***булевских литерала*** – ***true*** и ***false***.
31. Для использования простейших операций ввода-вывода в языке C++ используется заголовочный файл стандартной библиотеки потоков ввода-вывода ***iostream***. Чтобы его подключить к исходному файлу, необходимо использовать директиву препроцессора ***#include***.
32. Для стандартного вывода данных в языке C++ будет использоваться стандартный поток вывода в виде объекта ***cout*** из пространства имён ***std***.
33. Для стандартного ввода данных в языке C++ будет использоваться стандартный поток ввода в виде объекта ***cin*** из пространства имён ***std***.
34. Для перехода на новую строку вывода будет использоваться объект ***endl*** из пространства имён ***std***.
35. Для того, чтобы отвязать пространство имён от вышеописанных объектов и использовать данные объекты по простому имени, используйте оператор ***using namespace*** с указанием соответствующего пространства имён.

Способы вывода данных на консоль в языке C++

1) Простейший вывод строк в C++ осуществляется с использованием инструкции ***std::cout*** и оператора **<<** (из библиотеки ***iostream***):

```
std::cout << "C++ is the most powerful ";
std::cout << "programming language";
```

Результат:



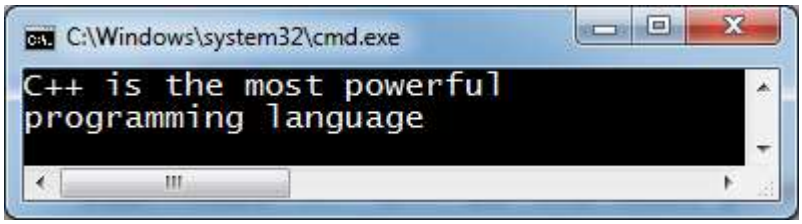
2) Как видно из предыдущего примера, инструкция ***std::cout << "..."*** не осуществляет перевод каретки на новую строку. Если необходимо всё-таки осуществить переход на следующую строку, то можно дополнительно добавить в шаблонную строку специальный символ новой строки ***'\n'*** или воспользоваться объектом ***std::endl***:

```
std::cout << "C++ is the most powerful\n";
std::cout << "programming language\n";
```

или

```
std::cout << "C++ is the most powerful" << std::endl;
std::cout << "programming language" << std::endl;
```

Результат:



Пример простой C++-программы:

Данная программа демонстрирует выполнение операции деления над целочисленными данными в C++.

```
2   #include <iostream>
3
4   void main() {
5
6       int a = 7;
7       int b = 2;
8       int c = a / b;
9
10      std::cout << "Division operation:\n";
11      std::cout << a << " / " << b << " = " << c << "\n";
12  }
```

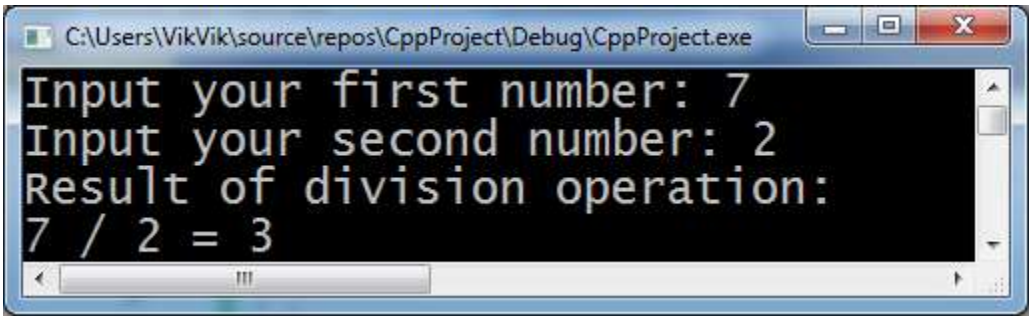
Результат работы программы:

Пример интерактивной C++-программы:

Данная программа демонстрирует использования стандартных объектов ввода-вывода языка C++ для организации диалога между программой и пользователем (создание интерактивного режима работы программы), а также оператора using для упрощения работы с объектами стандартной библиотеки потоков ввода-вывода.

```
1     #include <iostream>
2     using namespace std;
3
4     void main() {
5         int a, b;
6
7         cout << "Input your first number: ";
8         cin >> a;
9
10        cout << "Input your second number: ";
11        cin >> b;
12
13        int c = a / b;
14
15        cout << "Result of division operation: " << endl;
16        cout << a << " / " << b << " = " << c << endl;
17    }
```

Результат работы программы:



Контрольные вопросы

1. Какие пользовательские типы данных можно использовать в языке C++?
2. Что необходимо добавить в исходный код программы на языке C++, чтобы он был запускаемым?
3. Каков синтаксис стартовой функции *main(...)*?
4. Что такое литерал (*literal*) и зачем он нужен?
5. Что такое переменная (*variable*) и зачем она нужна?
6. Что такое локальная переменная?
7. Что такое глобальная переменная?
8. Как в языке Java объявить локальную (глобальную) переменную?
9. Что гласит правило объявления переменных в языке C++?
10. Что нужно помнить при выборе имени переменной?
11. Какие системы счисления поддерживаются в языке C++ при работе с целыми числами?
12. Как язык C++ работает с вещественными данными?
13. Опишите типы литералов (*целочисленные, вещественные, булевские, символьные* и *строковые*), доступных в языке C++. Какими способами каждый из них можно отобразить?
14. Какой тип по умолчанию имеют целочисленные и вещественные литералы?
15. Зачем нужны спецификаторы типов и сколько их?