

Hello, world!

Историческая справка

C — это язык программирования, созданный в 70-х годах XX века для разработки системы UNIX и программного обеспечения для нее. В 80-х годах XX века на основе языка C был создан язык C++, являющийся объектно-ориентированным расширением языка C++. В настоящее время языки C и C++ являются наиболее распространенными языками для профессиональной разработки программного обеспечения для всех операционных систем. Синтаксис языка C и C++ не зависит от используемой системы и компилятора, однако набор доступных библиотек (например, для разработки графических приложений) является системно-зависимым и не стандартизирован.

В данных листках речь будет идти о языке C++. Многие из того, о чем пойдет ниже речь, верно и для языка C, но мы на этом останавливаться не будем.

Программа, которая используется для перевода программы с языка программирования в машинный код, пригодный для исполнения компьютером, называется компилятором. Мы будем использовать компилятор `gcc`, вариант компилятора `gcc` для языка C++ называется `g++`, а реализация компилятора `gcc` для системы Windows называется MinGW.

При этом все рассматриваемые примеры должны правильно компилироваться любым компилятором, соответствующим стандарту языка C++. Например, таким компилятором является MS Visual C++ последних версий

Для облегчения процесса написания, запуска и отладки программы используются среды разработки. Среда разработки, которой будем пользоваться мы, называется Code::Blocks.

1.2 Hello, world

Язык C++ является компилируемым языком. Для того, чтобы написать программу, вам необходимо в любом текстовом редакторе набрать следующий текст и сохранить его в файле, например, `hello.cpp`.

```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello, world!" << endl;
    return 0;
}
```

Будьте внимательны: язык C++ является чувствительным к регистру букв, то есть заменить `main` на `Main` или `MAIN` нельзя. Весь текст (за исключением текстовой строки `"Hello, world!"`) нужно набирать в нижнем регистре, то есть строчными буквами.

В среде разработки (например, Code::Blocks) для компиляции программы существует пункт меню, вызывающий компилятор. Если ваша программа написана правильно, то компилятор не выдаст никаких сообщений об ошибках и создаст исполняемый файл

(a.out в системе Linux или exe-файл в системе Windows). Этот файл содержит исполняемый двоичный машинный код.

Рассмотрим подробнее текст этой программы. В первой строчке мы подключаем к нашей программе файл с именем `iostream`, в котором содержится описание стандартной библиотеки ввода-вывода языка C++. В этом файле находится, в частности, определение объектов `cout` и `endl`, который мы будем использовать позднее. Вторая строка указывает компилятору на то, что мы будем использовать все функции, входящие в пространство имен `std`, то есть все функции, относящиеся к стандартной библиотеке C++. Третья строка содержит объявление функции `main`, не принимающей никаких аргументов и возвращающей значение `int`. Эта функция должна быть в каждой программе, именно эта функция получает управление при запуске программы. Четвертая строка содержит открывающуюся фигурную скобку, что означает начало функции `main`. В пятой строке мы при помощи оператора `<<` помещаем в объект `cout` строку "Hello, world!", а потом специальный объект `endl`, означающий символ перевода строки. Это приводит к печати на экране этой строки и последующему переводу каретки. В шестой строке мы даем инструкцию `return`, завершающую выполнение функции `main` и возвращающую нулевое значение. Седьмая строка содержит фигурную скобку, синтаксически закрывающую функцию `main`.

Для начала можно считать, что все строки, кроме пятой, являются некоторым набором “заклинаний”, без которых программа не будет работать и которые обязательно нужно указать, а вот пятую строку можно заменить на другие строки с различными инструкциями.

Переменные

Программы оперируют с данными. Данные хранятся в переменных — специальных областях памяти, отведенных для хранения данных. Каждая переменная имеет свой **тип** и должна быть объявлена перед использованием. При использовании в программе ранее не объявленной переменной компилятор выдаст сообщение об ошибке вроде

```
test.cpp: In function 'int main()':
test.cpp:5: 'c' undeclared (first use this function)
test.cpp:5: (Each undeclared identifier is reported only once
for each function it appears in.)
```

В этом сообщении указывается имя файла и номер строки, в которой обнаружена ошибка (test.cpp:5), а также описание ошибки ('c' undeclared — объект с именем 'c' не объявлен).

Объявление переменной имеет следующий вид:

<тип переменной> <один или несколько идентификаторов переменных через запятую>;

Например, переменные `n` и `m` типа `int` можно объявить такой строкой:

```
int n, m;
```

Переменную `x` типа `double` можно объявить такой строкой:

```
double x;
```

Значения переменных сразу после объявления не определены и являются произвольными. Например, если написать следующую программу:

```
int a;  
cout << a;
```

то на экран будет выведено произвольное целое число, например, -1208725516 (так называемый "мусор").

Идентификатор переменной — это ее имя, которое должно быть последовательностью букв латинского алфавита, символа подчеркивания и цифр, начинающейся с буквы. Примеры правильных идентификаторов: `a`, `i`, `Year`, `school179`. Имена переменных чувствительны к регистру букв, то есть `Number`, `number`, `NUMBER` и `nUMbeR` — это четыре разных переменных.

Арифметические операторы

Арифметическая инструкция — это некоторое выражение, состоящее из констант, идентификаторов переменных и арифметических операторов, которая завершается точкой с запятой. Самый главный арифметический оператор — это оператор присваивания `'='`, который присваивает одной переменной, идентификатор которой указывается слева от оператора `'='` значение некоторого выражения, которое стоит справа. Например:

```
a = 2;  
b = a + 1;
```

Мы начнем с рассмотрения только целочисленных величин. Для хранения в памяти целочисленных величин как правило используются переменные типа `int`. С переменными типа `int` можно делать следующие арифметические действия:

- + Сложение
- Вычитание
- * Умножение
- / Деление (целочисленное)
- % Остаток от деления

Особого внимания заслуживают операторы деления и остатка. Деление производится нацело, с отбрасыванием целой части. Например, `10/3` это 3, `179/10` это 17, `2/5` это 0. Оператор `%` возвращает остаток от деления того, что записано справа на то, что записано слева. Например, `10%3` это 1, `179%10` это 9, `2%5` это 2.

В арифметическом выражении сначала выполняются слева направо все операторы умножения и деления, затем слева направо все операторы сложения и вычитания, затем справа налево все операторы присваивания. При необходимости порядок действий можно изменить при помощи скобок.

Ввод-вывод

Для того, чтобы вывести на экран значение переменной или текстовой строки нужно использовать объект “cout” и оператор “<<”, который в данном случае следует называть *"Поместить в"*. cout — объект, связанный со стандартным выводом программы, как правило, это терминал. Для того, чтобы перейти при печати на новую строку необходимо поместить в cout стандартный объект endl.

Текстовые строки при выводе на экран необходимо заключать в двойные кавычки. Если хочется вывести на экран несколько объектов (переменных, текстовых строк и т.д.), то их нужно разделять между собой оператором ‘<<’.

Для того, чтобы считать значение переменной нужно использовать объект “cin” и оператор ‘>>’, который надо называть *"Извлечь из"*. При этом считывание данных будет производиться со стандартного ввода программы, как правило, являющегося клавиатурой. Если хочется за одну операцию считать несколько переменных, то их идентификаторы нужно разделять между собой оператором “>>”.

2.5 Более сложный пример

Обобщим все изложенное выше в более сложной программе, которая находит сумму двух введенных чисел.

```
#include <iostream>

using namespace std;

int main()
{
    int a, b, s;
    cout << "Введите два числа: ";
    cin >> a >> b;
    s = a + b;
    cout << a << "+" << b << "=" << s << endl;
    return 0;
}
```

Упражнения

В: Следующее и предыдущее

Напишите программу, которая считывает целое число и выводит текст, аналогичный следующему:

```
The next number for the number 179 is 180.
The previous number for the number 179 is 178.
```

В данном случае было введено число 179.

С: a^4

Дано целое число a . Вычислите и выведите значение a^4 . Сколько операций умножения вы использовали? Какое наименьшее число операций умножения необходимо для решения этой задачи?

D: a^6

Дано целое число a . Вычислите a^6 при помощи трех операций умножения.

E: a^7

Дано целое число a . Вычислите a^7 при помощи четырех операций умножения.

F: a^{21}

Дано целое число a . Вычислите a^{21} при помощи шести операций умножения.

Запустите эту программу и введите число 10. Почему получился такой странный ответ?.

G: Последняя цифра

Дано натуральное число. Найдите его последнюю цифру.

H: Число десятков двузначного числа

Дано двузначное число. Найдите число десятков в нем.

I: Число десятков

Дано натуральное число. Найдите число десятков в его десятичной записи (то есть вторую справа цифру его десятичной записи).

J: Сумма цифр

Дано трехзначное число. Найдите сумму его цифр.

K: Следующее четное

Дано целое число n . Выведите следующее за ним четное число.

Пример

Ввод	Вывод
7	8
8	10

L: Обмен значений - 1

Напишите программу, которая считывает значения двух переменных a и b , затем меняет их значения местами (то есть в переменной a должно быть записано то, что раньше

хранилось в b , а в переменной b записано то, что раньше хранилось в a). Затем выведите значения переменных.

Пример

Ввод	Вывод
3 7	7 3

Примечание. Решением задачи **не** является такой код:

```
cin >> a >> b; cout << b << " " << a;
```

М: Обмен значений - 2

Решите предыдущую задачу без использования дополнительной переменной.

Н: Стоимость покупки

Пирожок в столовой стоит a рублей и b копеек. Определите, сколько рублей и копеек нужно заплатить за n пирожков. Программа получает на вход три числа: a , b , n , и должна вывести два числа: стоимость покупки в рублях и копейках.

Пример

Ввод	Вывод
10 15 2	20 30
2 50 4	10 0

О*: Симметричное число

Дано четырехзначное число. Определите, является ли его десятичная запись симметричной. Если число симметричное, то выведите 1, иначе выведите любое другое целое число. Число может иметь меньше четырех знаков, тогда нужно считать, что его десятичная запись дополняется слева незначащими нулями.

Пример

Ввод	Вывод
2002	1
2008	37

Р*: Проверьте делимость

Даны два натуральных числа n и m . Если одно из них делится на другое нацело, выведите 1, иначе выведите любое другое целое число.

Пример

Ввод	Вывод
2 8	1
8 2	1
3 5	7

Q **: Максимум

Напишите программу, которая считывает значения двух переменных *a* и *u*, затем выводит наибольшее значение из них. *a* и *u* — натуральные числа от 1 до 1000.

При решении задачи можно пользоваться только переменными типа `int`, операциями `+`, `-`, `*`, `/`, `%`, `=`. Нельзя пользоваться нелинейными конструкциями: ветвлениями, циклами, функциями.

Пример

Ввод	Вывод
8 5	8
5 8	8
5 5	5