

ПРАКТИЧЕСКИЙ КУРС ДЛЯ НАЧИНАЮЩИХ

Боевой курс C++

Концентрированный курс по языку программирования C++: основные конструкции языка, работа с указателями и машинной памятью, объектно-ориентированное программирование, стандартная библиотека...

⌚ Время изучения курса: 5 тематических блоков

Начать курс

Шамин Роман Вячеславович
доктор физико-математических наук,
директор Института перспективных технологий и индустриального программирования
МИРЭА – Российского технологического университета

Почему язык программирования C++?



Язык программирования C++ самый мощный и быстрый!
Программы на C++ работают на всех платформах, а программисты всегда востребованы.

C++ является базовым языком программирования, который обязательно нужно знать.

Разделы курса

1

Основные конструкции языка C++. Технология профессионального программирования

2

Арифметика указателей. Динамическая память. Структуры и массивы.

3

Объектно-ориентированное программирование: инкапсуляция, наследование, полиморфизм.

4

Шаблоны и стандартная библиотека STL. Алгоритмы и итераторы. Работа с файлами.

5

Раздельная компиляция. Профессиональное программирование на C++

Практическое овладение материала позволяет сразу использовать полученные знания



Инструменты, которые нам понадобятся:



Visual Studio Code

Бесплатный, но мощный редактор кода. Загрузить можно по адресу: code.visualstudio.com



Terminal

Если Вы используете Microsoft Windows, то рекомендую использовать Windows PowerShell. Также можно использовать Far Manager. На Mac и Linux хорошие встроенные терминалы.



Компиляторы C++

Компилятор – это центральный инструмент при программировании. Язык C++ является компилируемым в машинный код, поэтому выбирать компилятора нужно очень ответственно. Хороших и бесплатных компиляторов не так много: GCC (GNU Compiler Collection) – стандартный набор компиляторов Linux, который включает в себя и C++. Для Windows есть его вариант Mingw-64. Также можно использовать Clang, но при его использовании могут быть проблемы с русскими буквами. Прекрасной средой разработки, включающей компилятор, является Microsoft Visual Studio, но только для Windows.



Другие инструменты также могут быть Вам полезны, о них мы будем упоминать по ходу нашего курса.

1

Устанавливаем Visual Studio Code с официального сайта:

<https://code.visualstudio.com/>

2

Если Вы работаете в Microsoft Windows, то установите компилятор Mingw-64 с сайта: <https://www.mingw-w64.org/>

Во вкладке “Downloads” выбираем MingW-W64-builds, скачиваем, устанавливаем.

3

Находим папку, в которую был установлен компилятор – у меня это такая:
C:\Program Files\mingw-w64\x86_64-8.1.0-posix-seh-rt_v6-rev0\mingw64\bin
и прописываем этот путь в переменные среды PATH.

О том как это сделать можно прочитать тут:

<https://www.java.com/ru/download/help/path.html>



Напишем нашу первую программу, которая будет печатать строку “Hello, World!”.

Любая программа на C++ должна иметь функцию `main`, которая является точкой входа (началом выполнения программы). Эта функция в простейшем варианте имеет следующий вид:

```
int main()  
{  
}
```

Слово “`int`” означает, что функция `main` возвращает целое значение (об этом мы будем говорить потом). В круглых скобках указываются параметры, которые в нашем случае отсутствуют. Наконец, в фигурных скобках будет тело функции `main`.

В языке C++ операции ввода/вывода реализуются с помощью стандартной библиотеки, поэтому для того, чтобы вывести на печать строку, необходимо подключить нужную библиотеку. Это делается следующей строкой, которую мы поместим в самое начало программы.

```
#include <iostream>  
int main()  
{  
}
```

В языке C++ печать на экране – это вывод строки в терминал или как его еще называют консоль. Вывод организуется в виде потока, который называется `cout`. Мы можем «отправлять» в этот поток последовательно символы, которые будут печататься на экране. Как мы уже говорили, консоль `cout` – это элемент стандартной библиотеки, а не языка C++, поэтому обращаться к нему нужно указывая имя библиотеки: `std::cout`. Вот как вывести строку:

```
std::cout << “Hello, World!”;
```

Вызов функции или оператора в C++ заканчивается точкой запятой, но не каждая строка программы.



Вот наша первая программа:

```
#include <iostream>
int main()
{
    std::cout << "Hello, World!";
}
```

Сохраните текст этой программы в файле `hello.cpp` и откомпилируйте следующим образом:

```
g++ hello.cpp -o hello.exe
```

В результате будет создан файл `hello.exe`, который можно запустить из проводника или из терминала. Если запустить из проводника, то мелькнет черный экран, потому что консоль, в которой будет наша строка, сразу же закроется, как завершится наша программа. Запустите ее в терминале командой:

```
./hello.exe
```

Вы увидите строку «Hello, World!».

Теперь добавим что-нибудь полезное, например, пусть программа вычислит $\ln(2)$. Для этого нужно, прежде всего, подключить математическую библиотеку `cmath` и модифицируем программу:

```
#include <iostream>
#include <cmath>
int main()
{
    std::cout << "Hello, World!";
    std::cout << log(2);
}
```



После того, как вы запустите эту программу, то увидите, что приветствие и логарифм печатаются встык:

```
Hello, World!0.693147
```

Добавим переносы, модифицировав строки следующим образом:

```
std::cout << "Hello, World!" << std::endl;  
std::cout << log(2) << std::endl;
```

Слово endl означает специальный символ окончания строки и переход на новую строку. Заметим, что это слово тоже из стандартной библиотеки и нуждается в префиксе std.

Поскольку много раз писать std:: может быть утомительно, то в C++ можно подключить пространство имен и более не писать std::

```
#include <iostream> // подключаем библиотеку ввода/вывода  
#include <cmath> // математическая библиотека  
using namespace std; // подключение пространства имен  
  
int main()  
{  
    cout << "Hello, World!" << endl; // выводим приветствие  
    cout << log(2) << endl; // выводим логарифм  
}
```

Обратите внимание на текст, написанный зеленым цветом, начинающийся с // это комментарий. Комментарии могут быть в любом месте. Текст комментариев полностью игнорируется, поэтому там можно писать по-русски. Комментарии нужны, чтобы пояснять текст программы, и являются признаком профессионального программирования.

