Конспект лекций Мещерина

Содержание

1. Введение

Лекция 1

О лекторе

Лектора зовут Мещерин Илья Семирович.
VK: vk.com/mesyarik
TG: @mesyarik

1.1. Общие слова о языке и исторические заметки

Почему изучается именно этот язык программирования и что он даст? На сайте www.tiobe.com в котором собрана информация о популярности языков программирования C++ занимает достаточно высокое место.

Существуют языки **общего** и **специального** назначений, например на C++, C#, Python, Java можно писать почти все что угодно, а на JavaScript — нет.

По этому курсу будет довольно много лекций, потому что C++ довольно большой язык. Вместе с изучением C++ будет затронут и его прородитель — C.

Почти всегда то, что написано на C++ будет работать и на C. Также, изучая C++ будет намного проще изучать Java и C#, потому что эти языки синтаксически похожи.

Язык C++ — очень сложен, освоив его изучать остальное будет проще. Сложность освоения C++ связана со многими вещами,

как минимум он довольно низкоуровневый. Придется следить за памятью, есть возможность общаться с ОС на прямую и т.д.

В Java и C# мы защищены от низкоуровневых инструкций, что делают их проще и безопаснее. Однако в этом есть и плюсы и минусы. Из-за незащищенности и низкоуровневости код C++ может работать в разы быстрее своих аналогов. Исходя из этого C++ используется там, где нужен высокопроизводительный код.

Примеров известных сервисов, которые написаны с использованием кода на C++ довольно много.

Яндекс: Поиск, Карты, Такси, Почта, Диск.

Google: Search, Youtube.

Телеграмм.

Также на C++ написано множество игр, например WoW.

Не стоит и забывать про ОС, даже Windows написан с использованием C++.

C++ используется и в науке.

C++ создал Бьерн Страуструп (Bjarne Stroustrup). В курсе его будем называть **создателем**.

У C++ много версий. Мы будем пользоваться стандартом C++ 17.

- C++98
- C++03

  ————
- C++11 (C++0x)
- C++14 (C++1y)
- C++17 (C++1z)

Между версиями языка не зря проведена черта. Они очень сильно отличаются. У каждой версии есть **стандарт**. Тонны страниц текста с мелким шрифтом, исчерпывающе описывающим версию языка. Из-за его объема и формальности не рекомендуется читать и вникать в него полностью, а лишь обращаться к нему и читать по диагнали в неоднозначных моментах.

Рекомендуется пользоваться сайтом www.cppreference.com. Если требуется что-то узнать о языке, то первым делом лучше лезть туда.

Не стоит брезговать сайтом www.stackoverflow.com, на котором почти всегда будут ответы на возникающие вопросы.

Рекомендованная литература, автора Скота Мейерса:

- Эффективное использование C++.
- Наиболее эффективное использование C++.
- Эффективный и современный C++.

1.2. Знакомство с компилятором и первая программа

Добро пожаловать в терминал Linux! В курсе рекомендуется пользоваться именно этой операционной системой.

Напишем первую программму.

```
#include <iostream>

int main() {
    int x;
    std::cin >> x;
    std::cout << x + 5 << '\n';
}
```

В программе на C++ обязательно должна быть такая функция как **main**. Здесь мы пишем программу от которой ожидаем простые действия, а именно ввести число, увеличить его на пять и вывести.

Чтобы делать ввод-вывод необходимо подключить **заголовочный файл** — iostream.

Написав строку

```
#include <iostream>
```

мы подключаем заголовочный файл и получаем возможность работать со стандартными потоками ввода-вывода.

Для того чтобы ввести переменную, необходимо ее **объявить**.

```
int x;
```

Чтобы ввести переменную из стандартного потока ввода пишем

```
std::cin >> x;
```

а чтобы вывести число в стандартный поток вывода из этой переменной увеличенное на 5 пишем

```
std::cout << x + 5 << '\n';
```

Здесь мы также добавляем к выводу **символ перевода строки** — '\n'.

Чтобы перевести строку можно написать и

```
std::cout << x + 5 << std::endl;
```

однако есть разница. Во втором случае мы также отчищаем буффер вывода.

Файлы исходного кода обычно выглядят как *.cpp.

Перейдем к компиляции. Для того чтобы получить исполняемую программу необходимо перевести исходный код в машинный, этот

процесс называется **компиляцией**, а наличие этого процесса отличает компилируемые языки программирования от интерпретируемых.

Для того чтобы скомпилировать исходный код необходимо вызвать специальную программу — **компилятор**. У C++ есть несколько компиляторов. Наиболее известным из них является gcc — Gnu Compiler Collection. Для того чтобы его вызвать именно для языка C++ необходимо писать g++.

Также есть такие компиляторы, как clang++ и msvc. Компилятор msvc разрабатывается компанией Microsoft и не рекомендован к использованию, поскольку многие вещи от туда не всегда соответсвуют стандарту.

Для отладки программы можно пользоваться программой дебагером, например gdb.