C++

compscicenter.ru

Башарин Егор

eaniconer@gmail.com

ЛЕКЦИЯ I

Intro

ОРГАНИЗАЦИОННЫЕ ВОПРОСЫ

АКТИВНОСТИ

- Лекции
- Домашние работы
- Семинары
  - Дорешки
  - Контрольные работы
- Экзамен
  - теория
  - задача

ПОПУЛЯРНОСТЬ ЯЗЫКА

- TIOBE July 2023: 3rd
- PYPL July 2023: 5th

ИЗВЕСТНОСТЬ ЯЗЫКА

- высокая эффективность и производительность
- уязвимости и UB

СФЕРЫ ПРИМЕНЕНИЯ

- Системное ПО, операционные системы
- Базы данных
- Браузеры
- Компьютерные игры
- Машинное обучение …

ИСТОРИЯ

C++ начал разрабатываться с начала 1980-х сотрудником Bell Labs Бьярне Страуструпом

"C с классами" и транслятор cfront

ИСТОРИЯ

К 1983 было реализовано большое количество возможностей, поэтому язык был переименован в C++

Имя языка связано с оператором постфиксного инкремента

СТАНДАРТИЗАЦИЯ

Classic: C++98,03

Modern: C++11,14,17,20, …

ХАРАКТЕРИСТИКИ ЯЗЫКА

- Высокая сложность изучения
- Свобода выбора стиля
- Эффективность
- Компилируемость
- Высокоуровневость и низкоуровневость
- Интероперабельность с языком C
- Кроссплатформенность

ВЫСОКАЯ СЛОЖНОСТЬ ИЗУЧЕНИЯ

- Огромный текст стандарта
- Сложный синтаксис
- Требует понимания системы, в которой запускается программа
- Знание идиом и "рецептов"

СВОБОДА ВЫБОРА СТИЛЯ

- Выбор парадигмы программирования
- Выбор уровней абстракции

ЭФФЕКТИВНОСТЬ

- Zero-overhead principle
- Возможность максимально оптимизировать участок кода

КОМПИЛИРУЕМОСТЬ

Компиляция - преобразование текста программы в машинный код

- для каждой платформы отдельно
- позволяет отловить некоторые ошибки
- нет накладных расходов при выполнении программы
- при изменении программы нужно компилировать снова

ЭТАПЫ КОМПИЛЯЦИИ

1. Preprocessor: Source Code Files → Translation Units
2. Compilation: Translation Unit → Object Files
3. Linker: (Object Files, Libraries) → Executable | Library

ВЫСОКОУРОВНЕВОСТЬ И НИЗКОУРОВНЕВОСТЬ

Низкоуровневые особенности: - Работа с памятью - Использование платформенно-специфичных функций - Прямой доступ к ресурсам

ИНТЕРОПЕРАБЕЛЬНОСТЬ С ЯЗЫКОМ С

- C++ отделился от C еще до стандартизации
- C не является подмножеством C++
- Language Linkage

КРОССПЛАТФОРМЕННОСТЬ

- Код пришется один раз, компилируется на всех платформах
- Есть возможность написать непортируемый код

ПРОГРАММА

Программа - последовательность инструкций

Точка входа:

```cpp
// main.cpp
int main() {
  return 0;
}
```

ОСНОВНЫЕ КОНСТРУКЦИИ

```cpp
// main.cpp
#include <iostream>

bool isGood(int n) { return n == 42; }

int main() {
    int n = 32;
    for (int i = 0; i < n; ++i) {
        if (isGood(i)) {
            std::cout << i << " is good!" << std::endl;
        }
    }
    return 0;
}
```

САМЫЙ ВАЖНЫЙ САЙТ

https://en.cppreference.com/

РАБОТА С ФАЙЛАМИ

```cpp
// main.cpp
#include <fstream>

int main() {
    std::ifstream in ("in.txt");
    std::ofstream out("out.txt");

    double value = 0.0;
    in >> value;
    out << value;

    return 0;
}
```

МНОГОФАЙЛОВАЯ ПРОГРАММА

```cpp
// main.cpp
#include <iostream>
#include "factorial.hpp"

int main() {
    std::cout << factorial(10);
    return 0;
}
```

ЗАГОЛОВОЧНЫЙ ФАЙЛ

```cpp
// factorial.hpp

int factorial(int n) {
   // your code here...
}
```

```cpp
// main.cpp
#include <iostream>
#include "factorial.hpp"

int main() {
   std::cout << factorial(10);
   return 0;
}
```

ПРОБЛЕМА 1

```cpp
// main.cpp
#include <iostream>
#include "factorial.hpp"
#include "factorial.hpp" // двойное включение

int main() {
    std::cout << factorial(10);
    return 0;
}
```

РЕШЕНИЕ ПРОБЛЕМЫ 1

```
// factorial.hpp
#pragma once

int factorial(int n) {
    // your code here...
}
```

ПРОБЛЕМА 2

Изменение фукнции `factorial` приводит к перекомпиляции `main.cpp`

РЕШЕНИЕ ПРОБЛЕМЫ 2

```hpp
// factorial.hpp
#pragma once

int factorial(int n);
```

```cpp
// factorial.cpp
#include "factorial.hpp"

int factorial(int n) {
   // your code here...
}
```