

Программная инженерия с использованием C++

Московский Физико-Технический Институт

Иван Сергеевич Макаров

- Выпускник Московского Физико-Технического Института
- Кандидат технических наук по специальности 1.2.1
- Доцент Высшей Школы Программной Инженерии МФТИ
- Автор 11 докладов на конференциях IEEE и ICPS
- Автор 20 публикаций в рецензируемых научных изданиях

В настоящее время я занимаюсь разработкой вычислительных систем с использованием технологий параллельного программирования, в частности, оптимизацией инфраструктурных компонент систем автоматизированной высокочастотной торговли.

Программа курса

- 01. Общее введение и обзор технологий
- 02. Структурное программирование
- 03. Объектно-ориентированное программирование
- 04. Обобщенное программирование
- 05. Паттерны и технологии проектирования
- 06. Организация проектов и библиотек
- 07. Обработка ошибок и исключений
- 08. Особенности математических вычислений
- 09. Низкоуровневое управление памятью
- 10. Коллекции объектов и контейнеры
- 11. Итераторы и алгоритмы на диапазонах
- 12. Кодирование символов и парсинг текстов
- 13. Потоки ввода-вывода и сериализация
- 14. Параллельное программирование
- 15. Компьютерные сети и сетевые технологии

Определение C++

C++ – это компилируемый язык программирования широкого назначения со слабой статической типизацией, который поддерживает несколько парадигм программирования и предоставляет как низкоуровневые, так и высокоуровневые средства.

- Процессор понимает низкоуровневый машинный код
- Программист пишет высокоуровневый исходный код
- Компилятор транслирует исходный код в машинный
- Типы всех объектов известны во время компиляции
- Допустимы автоматические неявные преобразования

Парадигмы программирования

- Декларативное программирование – описание результатов
- Императивное программирование – последовательности
- Процедурное программирование – подпрограммы
- Функциональное программирование – композиции функций
- Структурное программирование – ветвления и циклы
- Объектно-ориентированное программирование – классы
- Обобщенное программирование – типы и шаблоны
- Параллельное программирование – процессы и потоки
- Событийно-ориентированное программирование – события

Формирование C++

- Изначально разрабатывался как набор расширений C
- В настоящее время является самостоятельным языком
- Унаследовал компоненты Ada, Algol, Fortran и Simula
- Повлиял на Java, Go, Python и многие другие языки
- Занял место на рынке и имеет достойных конкурентов

Первый коммерческий релиз C++ состоялся 14 октября 1985 г.

Стандартизация C++

- C++98 – фундаментальный стандарт
- C++03 – патч
- Technical Report 1 2007 / Boost
- C++11 – ядро языка +30%, стандартная библиотека +100%
- C++14 – патч
- C++17 – патч
- C++20 – важные дополнения
- C++23 – патч
- C++26 – следующий разрабатываемый стандарт

Дополнительные возможности предоставляются библиотеками.

Использование C++

- Операционные системы и системы управления
- Высоконагруженные системы обработки данных
- Математическое и физическое моделирование
- Финансовые системы автоматической торговли
- Решения для отраслей высокой ответственности

C++ сочетает низкоуровневые и высокоуровневые технологии.

Инструменты разработчика

- Дистрибутивы ОС Linux – Ubuntu, Debian, Fedora
- Системы сборки – Clang, GCC, ICC, MinGW, MSVC
- Редакторы кода – Sublime, Vim, Visual Studio Code
- Отладчики и профилировщики – GDB, Valgrind
- Системы автоматизации сборки – CMake, MSBuild
- Системы контроля версий – Git, SVN, Mercurial
- Сервисы для хостинга проектов – GitHub, Bitbucket
- Графические клиенты – GitHub Desktop, SmartGit
- Среды разработки – CLion, Visual Studio, XCode
- Системы управления задачами – Asana, Jira, Trello

Полезные ресурсы

- learncpp.com – начальные учебные материалы
- cppreference.com – справочное руководство
- boost.org – документации к библиотекам Boost
- github.com – сервис для хостинга проектов
- stackoverflow.com – форум вопросов и ответов

Список изученной мной литературы доступен в Google таблице.

Этапы жизненного цикла проекта

- Формирование набора требований
- Оценка необходимых ресурсов
- Разработка архитектуры решения
- Детальное проектирование модулей
- Механическое написание кода
- Отладка и оптимизация системы
- Внедрение и техническая поддержка
- Вывод продукта из эксплуатации