

Введение

В начале 2000-х гг. возникла необходимость составления курса программирования на C++. С литературой в то время было не очень хорошо. Единственный учебник, который можно было использовать в качестве начального курса, — это издание из серии «Классика Computer Science» Р. Лафоре, 2003 г. [1], которое можно было бы рекомендовать и сейчас всем изучающим язык C++. Поэтому потребовалось создание учебника, ориентированного на уровень студентов того времени. Что-то получилось хорошо, что-то не очень, но с тех пор прошло уже достаточно времени, и сам язык претерпел определенные изменения.

Так получилось, что пришлось вновь вернуться к начальному курсу C++ и существенно его переработать, учитывая изменения, внесенные в стандарт языка C++11 от 2011 г., C++14 и C++17, соответственно 2014 и 2017 гг. Помимо множества мелких доработок в новом стандарте появилось несколько принципиальных новшеств — это класс списка инициализации, технология перемещения, и, жемчужина коллекции, — лямбда-выражение, которое существенно изменяет стиль программирования.

Учебное пособие построено в виде конспекта 20 лекций.

- **Структура C++ программы.** Директивы препроцессора. Описание функций, глобальных переменных и констант. Пространство имен. Создание программы в среде Visual Studio.
- **Типы данных. Описание переменных.** Переменные целого типа. Переменные с плавающей точкой. Константы числовые и символьные. Классы памяти.
- **Операторы и операции.** Арифметические операции. Битовые операции. Логические операторы и операции. Операторы: `goto`, `if`. Операторы цикла: `while`, `do while`, `for`. Переключатель `switch`.
- **Массивы и указатели.** Одномерные и многомерные массивы. Описание массивов, навигация, запись и извлечение данных.
- **Динамическое выделение памяти.** Операторы `new`, `delete`. Выделение памяти «с размещением».

- **Функции.** Описание функций. Сигнатура функции. `inline`-функции. Рекурсивные функции.
- **Структуры.** Описание и применение структур, объединений, битовых полей. Передача данных через головную функцию.
- **Классы.** Создание класса. Соккрытие данных — инкапсуляция. Конструкторы и деструкторы. Переменные и методы класса. Перегрузка операторов.
- **Разработка пользовательского класса `String`.** Статические функции класса. Семантика перемещения.
- **Наследование.** Модели наследования: является (`is-a`), содержит (`has-a`). Виртуальные методы и классы. Абстрактные классы.
- **Шаблоны.** Шаблонные функции. Построение шаблонного класса. Функции с переменным числом параметров. Обработка исключений.
- **Потоки.** Консольный ввод/вывод. Флаги. Манипуляторы потоков. Управление потоками ввода/вывода. Память как поток. Файловый ввод/вывод.
- **Работа с файлами.** Произвольный доступ к файлам. Доступ к файловому буферу. Итераторы потоковых буферов.
- **Стандартная библиотека STL.** Контейнер `vector`. Конструкторы, операторы и методы вектора. Контейнеры: `array`, `deque`, `list`.
- **Алгоритмы.** Алгоритмы поиска. Функциональные объекты. λ -выражения. Немодифицирующие алгоритмы.
- **Модифицирующие алгоритмы.** Поточковые итераторы. Алгоритмы: `copy()`, `transform()`, `fill()`, `generate()`, `replace()`, `remove()`, `unique()`.
- **Модифицирующие алгоритмы, продолжение.** Алгоритмы: `random()`, `partition()`, `sort()`. Алгоритмы упорядоченных интервалов. Численные алгоритмы.
- **Ассоциативные контейнеры.** Множества и мультимножества: `set`, `multiset`. Отображения и мультиотображения: `map`, `multimap`. Неупорядоченные ассоциативные контейнеры.
- **Итераторы.** Итераторы ввода/вывода. Прямые и двунаправленные итераторы. Итераторы прямого доступа. Обратные итераторы. Итераторы вставки. Вспомогательные функции итераторов. Специальные контейнеры: `stack`, `queue`, `priority_queue`.
- **Класс `string` библиотеки STL.** Конструкторы. Методы и операторы. Функции ввода/вывода.