

Программа по курсу «Языки программирования»
для студентов 2-го курса направления ПМИ
2018-2019

1. Объявление и описания классов. Член-функции и член-классы. Спецификаторы доступа. Отличие класса от структуры.
2. Конструкторы и деструкторы. Виды конструкторов: по умолчанию, с параметрами по умолчанию, копии, преобразования типов, move-конструкторы. Дружественные функции и дружественные классы.
3. Перегрузка операций бинарных и унарных. Перегрузка операции вывода в поток (чтения из потока)/ Перегрузка арифметических операций. Префиксная унарная операция (`@a ++a --a`) (общий случай). Постфиксная унарная операция (`a@ a++ a--`). Перегрузка операций ввода-вывода.
4. Класс динамического массива. Константные функции-члены. Перегрузка операции `[]`. Операция копирующего присваивания. Идиома `copy-and-swap`. Метод `resize()`.
5. Return Value Optimization. Функции-члены, которые генерируются «молча».
6. Класс `frac` дроби. Реализация преобразования типов: конструктор и операция. `Explicit-конструктор`.
7. Статические члены класса.
8. Коллекции и рекурсия. Класс Линейный односвязный список. Класс Бинарное дерево поиска.
9. Наследование. Порядок вызова конструкторов и деструкторов. Преобразование в иерархии "предок-потомок". `Urcast` и `downcast`. Переопределение функций.
10. Полиморфизм. Виртуальные методы. Цена виртуальности. Раннее и позднее связывание. Деструкторы и полиморфизм. Полиморфные контейнеры и клонирование. Система RTTI (`Runtime Type Identification`). `static_cast` и `dynamic_cast`. Операция `typeid` и структура `type_info`.
11. Коллекции и итераторы. Типы итераторов. Итератор для списка.
12. Шаблоны. Шаблоны VS Перегрузка. Шаблоны функций. Инстанцирование шаблона. Явное создание экземпляров. Конкретизация. Специализация. Компиляция. Шаблон класса. Шаблон класса `Stack`. Параметры шаблонов. Нетиповой параметр. Шаблон для класса `Stack` с нетиповым параметром. Параметры-шаблоны.
13. Класс `matrix`. Операция доступа по индексу в `matrix`. Конструктор копий и `operator=` для класса `matrix`. `move-конструкторы` и `move-operator=`.
14. Общая характеристика STL. Основные контейнеры STL. Итераторы STL.
15. Векторы. Конструкторы контейнеров. Методы контейнеров. Методы, работающие с указателями и диапазонами. Примеры из лекций.
16. Общие операции для всех контейнеров. Операции, характерные для последовательных контейнеров `vector` и `list`. Обратный итератор.
17. Контейнер `array`.
18. `Map` (Ассоциативный словарь). Внутреннее представление. Основные методы `insert()`, `find()`. Итерация по `Map`. `Map` для описания графа. Требования от `Map`. Примеры из лекций.
19. `Set`. Внутреннее представление. Основные методы `insert()`, `find()`. Итерация по `set`. Требования от множества. Примеры из лекций.
20. Алгоритмы. Алгоритмы, не модифицирующие последовательность. `copy()`, `find()` и `find_if()`, `find_end()`, `find_first_of()`, `count()` и `count_if()`, `equal()`, `mismatch()`, `for_each()` – возможные реализации.
21. Алгоритмы, модифицирующие последовательность. `copy_backward()`, `for_each()`, `transform()` – возможные реализации.
22. Функторы. Лямбда-выражения. Захват переменных из внешнего контекста. Сумма элементов контейнера с помощью `for_each`.
23. Виды итераторов – категории итераторов. Итераторы потоков. Итераторы вставки.
24. Умные указатели. `unique_ptr` (сравнение с `auto_ptr` и обычными указателями), `shared_ptr`, `weak_ptr`