

Программа курса «Языки программирования»
мехмат, 2 курс, направление «Фундаментальная информатика и
информационные технологии»
3 семестр 2014–15 уч.г.

Модуль 1. Основные конструкции языка программирования C++

1. Простейшая программа. Ввод-вывод.
2. Пространства имен. Коллизия имен из разных пространств и ее разрешение.
3. Препроцессор, его роль. Основные директивы препроцессора.
4. Основные типы. Неявные преобразования типов.
5. Основные операции и операторы.
6. Понятие ссылки.
7. Функции. Передача параметров по ссылке.
8. inline-функции.
9. Предварительное объявление.
10. Многофайловая компоновка. Схема компиляции программы. Ее отличие от схемы компиляции на Паскале.
11. Содержимое заголовочных файлов. Что нельзя помещать в заголовочные файлы.
12. Стражи включения.
13. Предкомпилируемые заголовки.
14. Ошибки и особенности компоновки.
15. Массивы. Хранение в памяти. Передача массивов в функции.
16. Строки как массивы символов. Ввод строк.
17. Двумерные массивы и их хранение в памяти.
18. Определение типов.
19. Указатели. Передача параметров с помощью указателей.
20. Бестиповые указатели. Правила преобразования типов для них. Ошибки, связанные с приведением типов указателей.
21. Указатели и константность. Указатели на константы и константные указатели. Ссылки и константность.
22. Операции с указателями. Связь одномерных массивов и указателей.
23. Передача одномерных массивов в подпрограммы как указателей на своё начало.
24. Идиома *r++, пример использования.
25. Указатели и C-строки. Стандартные функции работы с C-строками.
26. Указатели и динамическая память.
27. Двумерные динамические массивы. Передача двумерного массива как параметра функции.
28. Указатели на функции. Callback-вызовы.
29. Структуры. Указатели на структуры.
30. Использование указателей для создания динамических структур данных. Пример: создание списка.

Модуль 2. Объектно-ориентированное программирование в C++

31. Классы. Защита доступа. Пример: класс Date. Константные функции-члены.
32. Операции над типами. Перегрузка бинарных операций.
33. Операции над типами. Перегрузка унарных операций.
34. Дружественные функции. Операции ввода-вывода.
35. Класс динамического массива. Операция []. Деструкторы, момент вызова деструктора. Отличие механизма деструкторов от сборки мусора.
36. Объекты классов в динамической памяти.
37. Конструктор копии.
38. Операция присваивания.
39. Move-конструктор и момент, когда он вызывается (на примере класса myvector).
40. Move-операция присваивания (на примере класса myvector).
41. Члены и роль класса std::initializer_list. Пример использования (на примере класса myvector).
42. Ситуации, в которых вызывается конструктор копии. Return Value Optimization.
43. Строки string, их операции и методы.
44. Шаблоны классов. Шаблоны функций. Описание шаблонов. Два этапа компиляции шаблонов. Где следует размещать шаблоны. Отличие от обобщенных классов и методов в.NET.
45. Тип vector, его операции. Понятие емкости вектора.
46. Массив объектов класса.
47. Класс matrix. Инициализация подобъекта. Порядок вызова конструкторов и деструкторов. Список инициализации. Операция ().
48. Класс frac. Конструктор преобразования. Ключевое слово explicit. Операция приведения типа.

49. Наследование. Порядок вызова конструкторов и деструкторов. Конструктор копии и операция присваивания для потомка.
50. Принцип «Выделение ресурса есть инициализация»
51. Преобразование типов в иерархии «Предок-Потомок» для объектов, указателей и ссылок. `static_cast` и его проблемы.
52. Множественное наследование. Ромбовидное наследование и виртуальные базовые классы. //Проблемы множественного наследования.
53. Полиморфизм и виртуальные функции. Особенности реализации полиморфизма в C++.
54. Накладные расходы на аппарат виртуальных функций.
55. Полиморфные контейнеры. Виртуальные деструкторы.
56. Операция `dynamic_cast`.
57. Операция `typeid` и структура `type_info`.
58. Исключения в C++.

Модуль 3. Стандартная библиотека C++

Иерархия наследования потоков.

Файловые потоки. Примеры работы с файлами.

Строковые потоки.

Общая характеристика и состав стандартной библиотеки шаблонов C++ (STL).

Неассоциативные контейнеры `vector`, `list`. Их внутренняя реализация, операции, асимптотические оценки операций.

Итераторы, их классификация. Операции над итераторами. Недействительные итераторы.

Реализация итератора для класса `vector`. Пример использования в алгоритме `copy`.

Реализация итератора для класса `list`. Пример использования в алгоритме `copy`.

Общие операции с контейнерами. Общие типы для контейнеров. Операции с последовательными контейнерами.

Внешние функции `begin`, `end`.

Специфические операции для `list` и `vector`. Примеры использования.

Адаптеры контейнеров `stack` и `queue`.

Алгоритмы – обзор. Универсальность алгоритмов. Универсальность алгоритма `copy` для различных типов контейнеров. Классификация алгоритмов.

Алгоритмы, не модифицирующие последовательность: `for_each`, `find`, `find_if`, `find_first_of`, `adjacent_find`, `all_of`, `any_of`, `none_of`, `count`, `count_if`, `mismatch`, `equal`, `search`.

Поиск элемента с помощью лямбды и алгоритма `find_if`.

Накопление суммы с помощью лямбды и алгоритма `for_each`.

Алгоритмы, модифицирующие последовательность: `copy`, `copy_if`, `swap_ranges`, `transform`, `replace`, `replace_if`, `fill`, `generate`, `remove`, `remove_if`, `unique`, `reverse`, `rotate`, `random_shuffle`.

Примеры использования `transform`.

Примеры использования `generate`.

Удаление элементов в контейнере. Удаление неуникальных элементов в отсортированном контейнере.

Необходимость использовать методы контейнера для физического удаления элементов.

Устройство алгоритма `random_shuffle`.

~~Алгоритмы сортировки последовательности: `partition`, `stable_partition`, `sort`, `is_sorted`, `stable_sort`, `partial_sort`, `lower_bound`, `upper_bound`, `equal_range`, `binary_search`, `merge`, `inplace_merge`.~~

~~Сортировка последовательности по внешнему критерию.~~

~~Алгоритмы для работы с множествами: `includes`, `set_union`, `set_intersection`, `set_difference`, `set_symmetric_difference`. Примеры использования.~~

~~Алгоритмы для работы с перестановками: `next_permutation`, `prev_permutation`, `is_permutation`. Примеры использования.~~

Ассоциативные контейнеры `map`, `set`: их внутренняя реализация, асимптотические оценки операций.

Эквивалентность и равенство объектов. Цикл по ассоциативным контейнерам. Примеры использования.

Создание графа на базе `map`.

Ассоциативные контейнеры `unordered_map`, `unordered_set`: их внутренняя реализация, асимптотические оценки операций. Хеш-функция. Задание хеш-функции для нестандартных типов.

Лямбда-выражения. Примеры использования. Захват переменных по ссылке и по значению. Примеры использования лямбд с захваченными по ссылке и по значению переменными. Переменная, хранящая лямбда-выражение, и ее тип.

Объекты-функции (функторы). Примеры использования.

Итераторы вставки. Примеры использования. Реализация итератора вставки в конец.

Итераторы потоков. Примеры использования. Реализация итератора потока вывода.