Программа курса «Языки программирования»
мехмат, 2 курс, направление «Фундаментальная информатика и информационные технологии» 3 семестр 2018–19 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. Инициализация с помощью {} и её отличие от обычной инициализации.
5. Основные операции и операторы.
6. Функции. Понятие ссылки. Передача параметров по ссылке.
7. inline-функции.
8. Структуры. Передача структуры в функцию. Инициализация полей структуры с помощью {}. Кортежи.
9. Генерация случайных чисел.
10. Предварительное объявление функции.
11. Многофайловая компоновка. Схема компиляции программы. Ее отличие от схемы компиляции на Паскале.
12. Понятие препроцессора. Заголовочные файлы, их содержимое. Что нельзя помещать в заголовочные файлы.
13. Заголовочные файлы и пространства имен.
14. Ошибки и особенности компоновки.
15. Стражи включения.
16. Предкомпилируемые заголовки.
17. Массивы. Хранение в памяти. Передача массивов в функции.
18. C-строки как массивы символов. Ввод C-строк. Передача C-строк в функции.
19. Строки string, ввод-вывод, присваивание, основные операции, передача в функции. Основные методы.
20. Векторы, инициализация, присваивание, цикл по вектору, передача в функции. Основные методы.
21. Определение типов (с помощью using и typedef).
22. Указатели. Передача параметров с помощью указателей. Сравнение с передачей по ссылке.
23. Бестиповые указатели void*. Правила преобразования типов для них. Операция static_cast. Ошибки, связанные с приведением типов указателей.
24. Указатели на структуры и операция −>.
25. Указатели и константность. Указатели на константы и константные указатели. Ссылки и константность.
26. Операции с указателями. Связь одномерных массивов и указателей.
27. Передача одномерных массивов в подпрограммы как указателей на своё начало.
28. Идиома *p++, примеры использования: сумма элементов массива, сложение векторов, симметричность массива.
29. Указатели и C-строки. Определение длины строки. Копирование C-строк.
30. Стандартные функции работы с C-строками.
31. Указатели и динамическая память. Массивы в динамической памяти.
32. Двумерные массивы и их хранение в памяти.
33. Двумерные динамические массивы. Передача двумерного массива как параметра функции.
34. Использование указателей для создания динамических структур данных. Примеры: создание списка, проход по списку, освобождение памяти, занимаемой списком.
35. Простой класс односвязного списка. Деструктор и его необходимость. Момент вызова деструктора. Отличие от сборки мусора.
36. Указатели на функции. Callback-вызовы.

Модуль 2. Объектно-ориентированное программирование в C++

37. Классы. Защита доступа. Конструктор. Пример: класс Date. Константные функции-члены.
38. Операции над типами. Перегрузка бинарных операций (2 способа). Перегрузка +, +=, ==, != для класса Date.
39. Дружественные функции. Операции ввода-вывода (пример реализации для класса Date).
40. Операции над типами. Перегрузка унарных операций (2 способа). Перегрузка операции ++ (префиксной и постфиксной) для Date.

41. Класс динамического массива. Операция []. Проверка выхода за границы: генерация и перехват исключения.
42. Метод push_back() и стратегия расширения вектора. Понятие емкости вектора.
43. Объекты классов в динамической памяти.
44. Конструктор копии. Операция присваивания.
45. Ситуации, в которых вызывается конструктор копии. Возврат ссылки из функции.
46. Оптимизация RVO (Return Value Optimization).
47. Ссылка на rvalue. Стандартная функция std::move.
48. Move-конструктор копии и момент, когда он вызывается (на примере класса myvector).
49. Move-операция присваивания (на примере класса myvector).
50. Идиома Copy & Swap и её использование при создании Move-конструктора копии и Move-операция присваивания.
51. Шаблоны классов. Шаблоны функций. Описание шаблонов. Два этапа компиляции шаблонов. Где следует размещать шаблоны. Отличие от обобщенных классов и методов в .NET.
52. Конструктор по умолчанию и массив объектов класса.
53. Класс matrix. Инициализация подобъекта. Порядок вызова конструкторов и деструкторов при наличии подобъектов. Список инициализации. Операция ().
54. Класс frac. Конструктор преобразования. Ключевое слово explicit. Операция приведения типа.
55. Наследование. Порядок вызова конструкторов и деструкторов. Конструктор копии и операция присваивания для потомка.
56. Преобразование типов в иерархии «Предок-Потомок» для объектов, указателей и ссылок. static_cast и его проблемы.
57. Множественное наследование. Ромбовидное наследование и виртуальные базовые классы.
58. Полиморфизм и виртуальные функции. Особенности реализации полиморфизма в C++.
59. Полиморфные контейнеры. Виртуальные деструкторы. Клонирование полиморфных контейнеров. Чисто виртуальные функции.
60. Накладные расходы на аппарат виртуальных функций. Таблица виртуальных функций VMT.
61. Система RTTI. Операция dynamic_cast. Операция typeid и структура type_info.
62. Исключения в C++. Генерация и перехват исключения. Порядок записи обработчиков исключений. Обработка всех исключений. Обработка исключений и деструкторы локальных объектов. ~~Исключения в конструкторе.~~

Модуль 3. Стандартная библиотека C++

63. Иерархия потоковых классов ввода-вывода.
64. Файловые потоки. ~~Двоичные файловые потоки.~~ Примеры.
65. Строковые потоки. Примеры.
66. Ошибки ввода-вывода и их обработка.
67. Класс std::initializer_list. Пример использования.
68. Цикл foreach и условия его применимости.
69. Общая характеристика и состав стандартной библиотеки шаблонов C++ (STL): контейнеры, итераторы и алгоритмы.
70. Алгоритм copy, его универсальность.Требования, предъявляемые к шаблонному параметру алгоритма copy. Ошибки copy во время выполнения.
71. Неассоциативные контейнеры vector, list. Их внутренняя реализация, операции, асимптотические оценки операций.
72. Реализация итератора для класса vector. Пример использования в алгоритме copy. Цикл по вектору с помощью итератора. Константные итераторы и пример их использования.
73. Реализация итератора для класса list. Пример использования в алгоритме copy. Цикл по списку с помощью итератора.
74. Общие операции с контейнерами. Общие типы для контейнеров. Операции с последовательными контейнерами. Внешние функции begin, end. Понятие недействительного итератора.
75. Специфические операции для list и vector. Примеры использования.
76. Адаптеры контейнеров stack и queue.
77. 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. Использование auto с лямбда-выражениями.
78. Объекты-функции (функторы). Пример использования.
79. Итераторы, их классификация. Операции над итераторами. Недействительные итераторы.

80. Алгоритмы STL – обзор. Универсальность алгоритмов. Классификация алгоритмов.
81. 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. Примеры использования.
82. Поиск элемента с помощью лямбды и алгоритма find_if.
83. Накопление суммы с помощью лямбды и алгоритма for_each.
84. Алгоритмы, модифицирующие последовательность: copy, copy_if, swap_ranges, transform, replace, replace_if, fill, generate, remove, remove_if, unique, reverse, rotate, random_shuffle. Примеры использования.
85. Итераторы вставки. Реализация итератора вставки в конец.
86. Итераторы ввода-вывода. Реализация итератора вывода.
87. Примеры использования transform. Стандартные арифметические объекты-функции и их преимущества перед лямбда-выражениями. Примеры использования.
88. Стандартные логические объекты-функции и их преимущества перед лямбда-выражениями. Примеры использования.
89. Примеры использования generate. Объекты-функции для генерации арифметической прогрессии и чисел Фибоначчи.
90. Удаление элементов в контейнере. Удаление неуникальных элементов в отсортированном контейнере. Необходимость использовать методы контейнера для физического удаления элементов.
91. Алгоритмы сортировки последовательности: partition, stable_partition, sort, is_sorted, stable_sort, partial_sort, lower_bound, upper_bound, equal_range, binary_search, merge, inplace_merge. Примеры использования.
92. Алгоритмы для работы с множествами: includes, set_union, set_intersection, set_difference, set_symmetric_difference. Примеры использования.
93. Алгоритмы для работы с перестановками: next_permutation, prev_permutation, is_permutation. Примеры использования.
94. Ассоциативные контейнеры map<K,V>, set<K>: их методы, внутренняя реализация, асимптотические оценки операций. Условия, налагаемые на K и V. Эквивалентность и равенство объектов. Цикл по ассоциативным контейнерам. Примеры использования. Реализация set и map для пользовательского типа (три способа).
95. Ассоциативные контейнеры unordered_map<K,V>, unordered_set<K>: их внутренняя реализация, асимптотические оценки операций. Хеш-функция. Реализация unordered_set<K> для пользовательского типа (три способа).
96. Умные указатели. unique_ptr. Основные методы и приёмы работы.
97. Умные указатели. shared_ptr. Основные методы и приёмы работы. Сравнение со сборкой мусора.
98. Умные указатели. weak_ptr. Основные методы и приёмы работы.