

Вопросы к экзамену по ПнаЯВУ(зимняя сессия 2022/2023)

1. Абстрактные типы данных
2. Базовые принципы объектно-ориентированного программирования
3. Основные достоинства языка C++
4. Особенности языка C++
5. Ключевые слова
6. Константы и переменные
7. Операции
8. Типы данных
9. Передача аргументов в функцию по умолчанию
10. Простейший ввод и вывод
11. Объект cin
12. Объект cout
13. Манипуляторы ввода/вывода
14. Операторы для динамического выделения и освобождения памяти (new и delete)
15. Объекты
16. Понятие класса
17. Конструктор с параметрами, конструктор по умолчанию.
18. Деструктор.
19. Конструктор копирования
20. Конструктор explicit
21. Указатель this
22. Встроенные функции (спецификатор inline)
23. Организация внешнего доступа к локальным компонентам класса (спецификатор friend)
24. Дружественные классы
25. Вложенные классы
26. Static-члены (данные) класса
27. Компоненты-функции static и const
28. Проху-классы
29. Ссылки
30. Параметры ссылки
31. Пространства имен
32. Определение пространства имен
33. Ключевое слово using как директива
34. Ключевое слово using как объявление
35. Псевдоним пространства имен
36. Наследование
37. Наследование (производные классы)
38. Конструкторы и деструкторы при наследовании
39. Виртуальные функции
40. Абстрактные классы
41. Виртуальные деструкторы
42. Динамическое и статическое связывание
43. Множественное наследование
44. Виртуальное наследование
45. Перегрузка функций
46. Перегрузка операторов
47. Перегрузка бинарного оператора
48. Перегрузка унарного оператора
49. Дружественная функция operator

50. Особенности перегрузки операции =
51. Перегрузка оператора []
52. Перегрузка оператора ()
53. Преобразование типов
54. Явные преобразования типов
55. Преобразования типов, определенных в программе
56. Параметризованные классы
57. Передача в шаблон класса дополнительных параметров
58. Шаблоны функций
59. Совместное использование шаблонов и наследования
60. Организация ввода-вывода
62. Организация ввода/вывода, потоки. Перегрузка операторов << и >>
63. Функция get() с тремя параметрами
64. Функция get() без параметров и с одним параметром
65. Функция getline()
66. Состояние потока
67. Функции чтения состояния потока
68. Организация работы с бинарными файлами. Запись объектов в файл и чтение объектов из файла
69. Организация работы с файлами последовательного доступа. Запись объектов в файл и чтение объектов из файла
70. Организация работы с файлами произвольного доступа. Запись объектов в файл, чтение объектов из файла, перезапись объектов в файле
71. Функции позиционирования в файле
72. Абсолютный обработчик
73. Основы обработки исключительных ситуаций
74. Перенаправление исключительных ситуаций
75. Исключительная ситуация, генерируемая оператором new
76. Генерация исключений в конструкторах
77. Задание собственной функции завершения
78. Спецификации исключительных ситуаций
79. Правила поиска обработчика исключительных ситуаций
80. Механизм разворачивания стека
81. Иерархия исключений стандартной библиотеки
82. Общее понятие о контейнере
83. Общее понятие об итераторе
84. Категории итераторов
85. Основные итераторы. Итераторы ввода/вывода
86. Вспомогательные итераторы
87. Операции с итераторами
89. Контейнеры последовательностей
90. Контейнер последовательностей vector
91. Контейнер последовательностей list
92. Контейнер последовательностей deque
93. Ассоциативные контейнеры
94. Ассоциативный контейнер multiset
95. Ассоциативный контейнер set
96. Ассоциативный контейнер multimap
97. Ассоциативный контейнер map
98. Адаптер stack
99. Адаптер queue
100. Адаптер priority_queue

- 101. STL-алгоритмы.
- 102. Алгоритмы сортировки sort
- 103. Алгоритмы поиска find
- 104. Алгоритмы swap
- 105. Алгоритм merge
- 106. Алгоритм for_each
- 107. Лямбда функции
- 108. Конструктор перемещения
- 109. Оператор присваивания перемещения
- 110. Паттерны проектирования
- 111. Паттерн "Абстрактная фабрика"
- 112. Умные указатели