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

***Прочитать самостоятельно( выделено красным)***

[1. Абстрактные типы данных](#_Toc185879268)

[2. Базовые принципы объектно-ориентированного программирования](#_Toc185879269)

[3. Основные достоинства языка С++](#_Toc185879270)

[4. Особенности языка С++](#_Toc185879271)

5[. Ключевые слова](#_Toc185879272)

[6. Константы и переменные](#_Toc185879273)

[7. Операции](#_Toc185879274)

[8. Типы данных](#_Toc185879275)

[9. Передача аргументов в функцию по умолчанию](#_Toc185879276)

[10. Простейший ввод и вывод](#_Toc185879277)

[11. Объект cin](#_Toc185879278)

[12. Объект cout](#_Toc185879279)

[13. Манипуляторы ввода/вывода](#_Toc185879280)

[14. Операторы для динамического выделения и освобождения памяти (new и delete)](#_Toc185879281)

[15. Объекты](#_Toc185879283)

[16. Понятие класса](#_Toc185879284)

17. Конструктор с параметрами, конструктор по умолчанию.

18. Деструктор.

[19. Конструктор копирования](#_Toc185879285)

[20. Конструктор explicit](#_Toc185879286)

[21. Указатель this](#_Toc185879287)

[22. Встроенные функции (спецификатор inline)](#_Toc185879288)

[23. Организация внешнего доступа к локальным компонентам класса (спецификатор friend)](#_Toc185879289)

24. Дружественные классы.

[25. Вложенные классы](#_Toc185879290)

[26. Static-члены (данные) класса](#_Toc185879291)

[27. Компоненты-функции static и const](#_Toc185879292)

[28. Proxi-классы](#_Toc185879293)

[29. Ссылки](#_Toc185879294)

[30. Параметры ссылки](#_Toc185879295)

[31. Пространства имен](#_Toc185879297)

[32. Определение пространства имен](#_Toc185879298)

[33. Ключевое слово using как директива](#_Toc185879299)

[34. Ключевое слово using как объявление](#_Toc185879300)

[35. Псевдоним пространства имен](#_Toc185879301)

[36. Наследование](#_Toc185879303)

[37. Наследование (производные классы)](#_Toc185879304)

[38. Конструкторы и деструкторы при наследовании](#_Toc185879305)

[39. Виртуальные функции](#_Toc185879306)

[40. Абстрактные классы](#_Toc185879307)

[41. Виртуальные деструкторы](#_Toc185879308)

42. Динамическое и статическое связывание

[43. Множественное наследование](#_Toc185879309)

[44. Виртуальное наследование](#_Toc185879310)

[45. Перегрузка функций](#_Toc185879312)

[46. Перегрузка операторов](#_Toc185879313)

4[7. Перегрузка бинарного оператора](#_Toc185879314)

[48. Перегрузка унарного оператора](#_Toc185879315)

[49. Дружественная функция operator](#_Toc185879316)

[50. Особенности перегрузки операции =](#_Toc185879317)

51[. Перегрузка оператора []](#_Toc185879318)

[52. Перегрузка оператора ()](#_Toc185879319)

[53. Преобразование типов](#_Toc185879322)

[54. Явные преобразования типов](#_Toc185879323)

[55. Преобразования типов, определенных в программе](#_Toc185879324)

[56. Параметризированные классы](#_Toc185879326)

[57. Передача в шаблон класса дополнительных параметров](#_Toc185879327)

[58. Шаблоны функций](#_Toc185879328)

[59. Совместное использование шаблонов и наследования](#_Toc185879329)

60. [Организация ввода-вывода](#_Toc185879336)

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](#_Toc185879337)

67. Функции чтения состояния потока

6[8. Организация работы с бинарными файлами. Запись объектов в файл и чтение объектов из файла.](#_Toc185879338)

6[9. Организация работы с файлами последовательного доступа. Запись объектов в файл и чтение объектов из файла.](#_Toc185879340)

[70. Организация работы с файлами произвольного доступа. Запись объектов в файл, чтение объектов из файла, перезапись объектов в файле.](#_Toc185879341)

71. Функции позиционирования в файле.

[72. Абсолютный обработчик](#_Toc185879342)

[73. Основы обработки исключительных ситуаций](#_Toc185879344)

[74. Перенаправление исключительных ситуаций](#_Toc185879345)

[75. Исключительная ситуация, генерируемая оператором new](#_Toc185879346)

[76. Генерация исключений в конструкторах](#_Toc185879347)

[77. Задание собственной функции завершения](#_Toc185879348)

[78. Спецификации исключительных ситуаций](#_Toc185879349)

79. Правила поиска обработчика исключительных ситуаций.

80. Механизм развертывания стека.

[81. Иерархия исключений стандартной библиотеки](#_Toc185879351)

[82. Общее понятие о контейнере](#_Toc185879353)

[83. Общее понятие об итераторе](#_Toc185879354)

[84. Категории итераторов](#_Toc185879355)

[85. Основные итераторы. Итераторы вовда/вывода](#_Toc185879356)

[86. Вспомогательные итераторы](#_Toc185879357)

[87. Операции с итераторами](#_Toc185879358)

[89. Контейнеры последовательностей](#_Toc185879359)

[90. Контейнер последовательностей vector](#_Toc185879360)

[91. Контейнер последовательностей list](#_Toc185879361)

[92. Контейнер последовательностей deque](#_Toc185879362)

[93. Ассоциативные контейнеры](#_Toc185879363)

[94. Ассоциативный контейнер multiset](#_Toc185879364)

[95. Ассоциативный контейнер set](#_Toc185879365)

[96. Ассоциативный контейнер multimap](#_Toc185879366)

[97. Ассоциативный контейнер map](#_Toc185879367)

[98. Адаптер stack](#_Toc185879369)

[99. Адаптер queue](#_Toc185879370)

[100. Адаптер priority\_queue](#_Toc185879371)

101.  STL-алгоритмы.

[102. Алгоритмы сортировки sort](#_Toc185879373)

[103. Алгоритмы поиска find](#_Toc185879374)

[104. Алгоритмы swap](#_Toc185879379)

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.