***Вопросы к экзамену по ПнаЯВУ (зимняя сессия 2022/202***

***Прочитать самостоятельно***

[~~1. Абстрактные типы данных~~](#_top)

[~~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)

~~61. Организация ввода / вывода, потоки. Перегрузка операторов << и >>~~

~~62. Функция get() с тремя параметрами.~~

~~63. Функция get() без параметров и с одним параметром~~

~~64. Функция getline()~~

~~65.~~ [~~Состояние потока~~](#_Toc185879337)

~~66. Функции чтения состояния потока~~

~~6~~[~~7. Организация работы с бинарными файлами. Запись объектов в файл и чтение объектов из файла~~](#_Toc185879338)

~~6~~[~~8. Организация работы с файлами последовательного доступа. Запись объектов в файл и чтение объектов из файла~~](#_Toc185879340)

[~~69. Организация работы с файлами произвольного доступа. Запись объектов в файл, чтение объектов из файла, перезапись объектов в файле~~](#_Toc185879341)

~~70. Функции позиционирования в файле~~

[~~71. Абсолютный обработчик~~](#_Toc185879342)

[~~72. Основы обработки исключительных ситуаций~~](#_Toc185879344)

[~~73. Перенаправление исключительных ситуаций~~](#_Toc185879345)

[~~74. Исключительная ситуация, генерируемая оператором new~~](#_Toc185879346)

[~~75. Генерация исключений в конструкторах~~](#_Toc185879347)

[~~76. Задание собственной функции завершения~~](#_Toc185879348)

[~~77. Спецификации исключительных ситуаций~~](#_Toc185879349)

~~78. Правила поиска обработчика исключительных ситуаций~~

79. Механизм развертывания стека

[~~80. Иерархия исключений стандартной библиотеки~~](#_Toc185879351)

[~~81. Общее понятие о контейнере~~](#_Toc185879353)

[~~82. Общее понятие об итераторе~~](#_Toc185879354)

[~~83. Категории итераторов~~](#_Toc185879355)

[~~84. Основные итераторы. Итераторы ввода/вывода~~](#_Toc185879356)

[~~85. Вспомогательные итераторы~~](#_Toc185879357)

[~~86. Операции с итераторами~~](#_Toc185879358)

[~~87. Контейнеры последовательностей~~](#_Toc185879359)

[~~88. Контейнер последовательностей vector~~](#_Toc185879360)

~~8~~[~~9. Контейнер последовательностей list~~](#_Toc185879361)

[~~90. Контейнер последовательностей deque~~](#_Toc185879362)

[~~91. Ассоциативные контейнеры~~](#_Toc185879363)

[~~92. Ассоциативный контейнер multiset~~](#_Toc185879364)

[~~93. Ассоциативный контейнер set~~](#_Toc185879365)

[~~94. Ассоциативный контейнер multimap~~](#_Toc185879366)

[~~95. Ассоциативный контейнер map~~](#_Toc185879367)

[~~96. Адаптер stack~~](#_Toc185879369)

[~~97. Адаптер queue~~](#_Toc185879370)

[~~98. Адаптер priority\_queue~~](#_Toc185879371)

~~99. STL-алгоритмы~~

~~100~~[~~. Алгоритмы сортировки sort~~](#_Toc185879373)

[~~101. Алгоритмы поиска find~~](#_Toc185879374)

[~~102. Алгоритмы swap~~](#_Toc185879379)

103. Алгоритм merge

~~104. Алгоритм for\_each~~

~~105. Лямбда\_функции~~

~~106. Конструктор перемещения~~

~~107. Оператор присваивания перемещения~~

~~108. Паттерны проектирования.~~

~~109. Паттерн "Абстрактная фабрика"~~

~~110. Умные указатели.~~