

Вопросы к зачету

Продвинутое программирование на C++, 2023 г.

Влада, Ваня, Стёпа, Андрей, Олег, Богдан, Сева

1. Cmake. Написать Cmake для проекта, состоящего из main.cpp, class1.h, class1.cpp. Стандарт C++ - 17.
2. Cmake. Алгоритм сборки библиотеки без зависимостей из исходных кодов, которые содержат CmakeLists.txt.
3. Cmake. Добавление сторонней прекомпилированной библиотеки к своему проекту.

3

4. Git. Отличие checkout от switch.
5. Git. Отличия rebase/merge/cherry-pick.
6. Git. Что нужно сделать, чтобы зафиксировать изменения в локальном репозитории и добавить их в удаленный репозиторий.

3

7. Отличия waterfall от agile.
8. Алгоритм Флойда.

2

9. Инварианты класса. Определение.
10. Инкапсуляция. Определение
11. Модификатор доступа в классе по умолчанию. Модификатор доступа в классе по умолчанию.
12. Наследование в классе по умолчанию. Наследование в структуре по умолчанию.
13. Отличия private/protected/public (внутри определения класса и при наследовании).
14. Отличие malloc/free от new/delete.

6

15. Когда не применима инициализация «в круглых скобках», «после =», «фигурная».
16. Сужающие преобразованию и разные виды инициализации («в круглых скобках», «после =», «фигурная»).
17. «Наиболее неприятный анализ» в C++ и разные виды инициализации («в круглых скобках», «фигурная»).
18. В каких случаях возникает двойная инициализация? Как избежать?

19. В каком порядке выполняется инициализация полей класса с использованием списков инициализации (в порядке расположения полей класса или в порядке, определенном в списке инициализации)?

5

20. Что такое RVO? Пример?

21. Полиморфизм. Определение. Виды полиморфизма.

22. Примеры статического полиморфизма в C++.

23. Примеры динамического полиморфизма в C++.

4

24. Отличие overloading от overriding.

25. Ключевое слово virtual.

26. Ключевое слово override.

27. Абстрактный класс. Определение.

28. Использование аргументов со значениями по умолчанию в виртуальных функциях.

6

29. NVI. Пример.

30. Может ли существовать шаблон виртуального метода?

31. Можно ли перегружать виртуальные функции?

32. Как добавить в overloading set класса-наследника методы базового класса, являющиеся перегрузками виртуального метода?

4

33. Чем плох GoTo?

34. Примеры goto -маскирующих конструкций

2

35. RAII. Концепция.

36. Инвариант умного указателя с единственным(уникальным) владением.

37. Drill down behaviour. Где применяется?

38. Lvalue и rvalue. Отличия.

39. Rvalue ссылки.

40. Что делает std::move?

41. Правило нуля, трех, пяти.

7

- 42. Виды умных указателей.
- 43. Отличие `unique_ptr` от `shared_ptr`.
- 44. Отличие `weak_ptr` от `shared_ptr`.
- 45. Способы инициализации `shared_ptr`.

4

- 46. ODR.
- 47. Include guards.
- 48. CV-квалификаторы при выводе типов.
- 49. Манглирование имен. (Как перегрузки функций представляются компилятором).
- 50. Правила разрешения перегрузки для нешаблонных функций.
- 51. Правила разрешения перегрузки для шаблонных функций.

6

- 52. Что такое специализация шаблонного класса?
- 53. Что такое частичная специализация шаблонного класса?
- 54. Правила вывода типов для `auto`.
- 55. Что делает `decltype`?
- 56. Когда используем `auto`, а когда `decltype` для вывода типов?
- 57. Когда C++ осуществляет «ленивые», а не «энергичные» вычисления?

6

- 58. SFINAE. Определение.
- 59. Зачем в старом C++ (до C++11) для реализации SFINAE создавались структуры разных размеров?
- 60. Замена структурам разных размеров в современном C++.

3

- 61. CRTP. Определение. Зачем нужно.
- 62. Ограничения CRTP.
- 63. Реализация чисто виртуальных функций в CRTP.
- 64. Реализация виртуальных функций в CRTP.
- 65. Реализация методов для удаления объекта класса в CRTP.
- 66. Способы применения CRTP.
- 67. Как применяется CRTP для делегирования?

7