

Вопросы к зачету

1. Ручное управление памятью. Выделение памяти на стеке. Опасности при возвращении адресов локальных переменных. Объект и его свойства. Классы объектов. Пространства имен. Жизненный цикл классов. Конструкторы и деструкторы.
2. Понятие объектно-ориентированного языка. Объектно-ориентированная декомпозиция. Базовые понятия ООП: абстрагирование. Абстракция сущности, поведения, виртуальной-машины, произвольная. Работа с вводом/выводом в C++. Потоки вывода и ввода.
3. Ссылки. Отличие от указателей. Lvalue и RValue переменные. LValue и RValue ссылки. Указатель nullptr. Базовые понятия ООП: инкапсуляция в C++.
4. Базовые понятия ООП: модульность в C++. Базовые понятия ООП: иерархия в C++. Агрегация и наследование. Виртуальные функции и полиморфизм. Модификаторы override и final. Абстрактные классы. Работа конструкторов и деструкторов при наследовании.
5. Базовые понятия ООП: Типизация и эквивалентность типов в C++. Базовые понятия ООП: Сохраняемость. Временные объекты в C++. Модификатор const в c++. Модификатор mutable в C++. Модификатор constexpr в C++. Явные и не явные конструкторы. Ключевое слово explicit.
6. Перегрузка операторов. Синтаксис. Возможные к перегрузке операции. Перегрузка ++ и --. Перегрузка бинарных операторов.
7. Перегрузка оператора =. Запрет операторов копирования. Перегрузка (). Функтор. Пользовательские литералы.
8. Исключения в C++. std::exception и std::exception_ptr. std::current_exception и std::rethrow_exception. Модификатор noexcept. Метод terminate
9. Идиома RAII. Умные указатели. std::shared_ptr. Наследование и std::weak_ptr. Проблемы при использовании std::shared_ptr.
10. Идиома RAII. Умные указатели. Шаблон std::weak_ptr. Шаблон std::unique_ptr. Семантика перемещения. std::move
11. Шаблоны классов и функций. Сравнение наследования и шаблонов. Параметры шаблонов. Специализация шаблонов.
12. Шаблоны классов и функций. Вычисление факториала на шаблонах.
13. Шаблоны классов и функций. inclusion model и explicit instantiation model. Шаблоны с переменным числом параметров.
14. Шаблоны классов и функций. Устройство шаблона tuple
15. Шаблоны классов и функций. Curiously Recurring Template Pattern. Множественное наследование классов с помощью шаблонов.
16. Шаблоны классов и функций. Шаблон из шаблонов. Ключевое слово auto. template alias. Протечка абстракции.
17. Итераторы. range-based циклы. std::initializer_list
18. Контейнеры STL. vector. Итераторы.
19. Контейнеры STL. map. Итераторы.
20. Контейнеры STL. deque. Итераторы.
21. Контейнеры STL. set. Итераторы.
22. Контейнеры STL. bitset. Итераторы.
23. Аллокаторы памяти. Перегрузка оператора new. Простой аллокатор памяти на массиве.
24. Проектирование структуры классов. Характеристики (жесткость, хрупкость, повторное использование. Способы улучшения структуры классов. SOLID: Принцип единой ответственности. Пример.
25. Метрика cohesion. Метрика coupling. Виды связанности. SOLID: Принцип открытости/закрытости. Пример.
26. Шаблон проектирования template method (обычный и CRTP). Шаблон проектирования strategy.
27. SOLID: Принцип подстановки Барбары Лисков.
28. Принцип TDA. Принцип Command Query Separation. Закон Постеля.
29. SOLID: Принцип разделения интерфейсов.
30. SOLID: Dependency Inversion Principle.
31. Мультипроцессирование и мультипрограммирование. SMP, MPP и NUMA. Мультипрограммирование. Вытесняющая и не вытесняющая многозадачность. Планировщик задач. Жизненный цикл потока.
32. std::thread. Функции и функторы как параметры. Использование семантики перемещения в std::thread. Функции из пространства имен std::this_thread.
33. Потокобезопасность. Реентерабельность. Race condition. Взаимное исключение. std::mutex и std::recursive_mutex.
34. std::lock_guard и std::unique_lock. Реализация потокобезопасного стека. dead_lock
35. Просачивание данных за пределы lock_guard. std::future и std::async std::promise.
36. Условные переменные. Закон Амдала.
37. Неблокирующие алгоритмы. Атомарные типы. CAS операции. Потокобезопасный стек на CAS-операциях.
38. Лямбда выражения. Списки захвата в лямбда выражениях. Лямбда выражения, порождающие лямбда выражения.

