

## Вопросы к экзамену АЯ 2018

1. Двусвязный список. Основные операции: поиск элемента по значению, вставка элемента, удаление элемента. Их алгоритмическая сложность. Сортировка. Реализация функции удаления элемента из двусвязного списка. Основные методы `std::list`. Пример работы с `std::list`
2. Класс `std::map`. Внутренняя реализация `map`, его основные методы. Сложность поиска по ключу, сортировки, удаления элемента, добавления элемента. Пример работы с `std::map`
3. Класс `std::set`. Внутренняя реализация `set`, его основные методы. Сложность поиска, удаления элемента, добавления элемента. Пример работы с `std::set`
4. Класс `std::unordered_map`. Внутренняя реализация `unordered_map`, его основные методы. Сложность поиска, сортировки, удаления элемента, добавления элемента. Пример работы с `std::unordered_map`
5. Класс `std::vector`. Внутренняя реализация `vector`, его основные методы. Сложность поиска, сортировки, удаления элемента, добавления элемента. Амортизированная сложность добавления элемента в конец. Пример работы с `std::vector`. Особенность `std::vector<bool>`.
6. Класс `std::vector` и `std::array`. Отличия. Внутренняя реализация `vector`, его основные методы. Сложность поиска, сортировки, удаления элемента, добавления элемента. Амортизированная сложность добавления элемента в конец. Пример работы с `std::array`.
7. Парадигмы ООП. Полиморфизм (статический, динамический). Инкапсуляция. Наследование. Примеры динамического и статического полиморфизмов.
8. Парадигмы ООП. Полиморфизм (статический, динамический). Инкапсуляция. Наследование. Ключевые слова `virtual`, `override`, `final`. Примеры.
9. Разработка обобщенных типов: шаблоны C++. Инстанцирование. Спецификация шаблонов. Примеры.
10. Итераторы: определение, назначение, преимущества. Итераторы прямого доступа, итераторы ввода, итераторы вывода, двунаправленные итераторы, прямой итератор. Примеры.
11. Современный C++: `auto`, `decltype`, `range base loop`, `nullptr`, `constexpr`, `enum class`, `if constexpr`.
12. Современный C++: `static_assert`, `default`, `final`, `override`. Преимущества `using` по сравнению с `typedef`. Конструкторы принимающие `initializer_list`. Примеры.
13. Современный C++: `std::optional`, `std::variant`, `std::any`, `std::string_view`. Примеры использования.
14. Лямбда-функции, функторы, указатели на функции, `std::functional`. Примеры использования `std::functional`. Примеры использования лямбда-функций.
15. Rvalue ссылки. Семантика перемещения. `std::move`, `std::forward`. Make-функции. Примеры.

16. Алгоритмы STL: `count_if`, `find_if`, `find`, `count`, `transform`, `sort`, `any_of`, `all_of`, `remove_if`, `replace_if`, `merge`, `max`, `max_element`, `is_sorted`, `lower_bound`, `upper_bound`, `accumulate` и др.
17. Обработка ошибок с использованием механизма обработки исключений. RAII. Примеры классов, использующих RAII. Ключевое слово `noexcept`.
18. RAII. «Умные» указатели. Класс `std::shared_ptr`. Устройство `shared_ptr`. Основные методы. Примеры использования.
19. RAII. «Умные» указатели. `std::weak_ptr`. Основные методы. Примеры использования.
20. RAII. «Умные» указатели. Класс `std::weak_ptr`. Основные методы. Связь с классом `std::shared_ptr`. Примеры использования.
21. Шаблоны проектирования. Фабрика. Пример реализации фабрики.
22. Шаблоны проектирования: синглтон, `PImpl`. Пример реализации `PImpl`.
23. Сетевое взаимодействие. Berkeley sockets. Основные функции для работы с сокетами
24. Сетевое взаимодействие. Сокеты. Библиотека `boost::asio`. `io_service`, `endpoint`, `ip::address`. Функции асинхронного и синхронного чтения и записи.
25. Управление потоками. Состояния гонок в интерфейсе структур данных. Класс `std::future`, функция `std::async`.
26. Переключение контекста потоков. Класс `std::thread`. Ключевое слово `thread_local`. Примеры использования `thread_local`.
27. Переключение контекста потоков. Класс `std::thread`. Ключевое слово `thread_local`. Примеры использования `std::thread`.
28. Синхронизация потоков. Состояние гонок. Классы `std::mutex`, `std::lock_guard`, `std::unique_lock`. Функция `std::lock`. Примеры использования мьютексов.
29. Синхронизация потоков. Состояние гонок. Классы `std::recursive_mutex`, `boost::shared_mutex`, `std::unique_lock`. Функция `std::lock`. Пример использования `std::unique_lock`.
30. Синхронизация потоков. Состояние гонок. Потокбезопасные структуры данных с блокировками.
31. Класс `std::condition_variable`. Потокбезопасные структуры данных с блокировками.
32. Синхронизация потоков. Пул потоков. Класс `std::condition_variable`. Примеры работы с `std::condition_variable`.
33. Управление потоками. Состояния гонок. Класс `std::future`, функция `std::async`. Пример работы с `std::future` и `std::async`.
34. Атомарные операции. Классы `std::atomic`, `std::atomic_flag`. Примеры работы с `std::atomic`
35. Перегрузка `new` и `delete`. Аллокаторы, системы управления памятью. `TSMalloc`
36. Пространства имен (`namespace`). Анонимные `namespace`. Что такое, для чего необходимы.

37. Синхронизация потоков. Состояние гонок. Шаблон producer-consumer. Примеры.
38. Функции для работы с файловой системой. `std::filesystem::path`, `directory_iterator`. Функции `exists`, `is_regular_file`, `is_directory`, `is_symlink`. Примеры работы с `std::filesystem`.