

1. Некоторые особенности C++11/14/17/20.

1. Вывод типов шаблонов, вывод типа `auto` и их отличия. Использование `decltype`. Инициализация с фигурными скобками. Range-based циклы.
2. Ключевые слова `override`, `final`, удалённые функции.
3. Перечисления с областью видимости, псевдонимы, `constexpr`-значения и `constexpr`-функции.
4. Лямбда-выражения. Связыватели. Функции `std::bind`, `std::invoke` и `std::apply`.
5. Шаблоны с переменным количеством параметров.
6. Неупорядоченные контейнеры `hashmap` и `hashset`.

2. Семантика перемещения.

1. Типы ссылок. `std::move`. Оператор перемещения и `move`-конструктор. Универсальные ссылки.
2. Прямая передача. Свертывание ссылок. Проблемы перегрузки универсальных ссылок.
3. Автоматическая генерация компилятором перемещающих и копирующих операций.

3. Умные указатели и владение памятью.

1. Динамическое выделение памяти. Размещающие операторы `new` и `delete`.
2. Типы владения памятью. Передача владения.
3. Указатель `std::unique_ptr` и исключительное владение.
4. Указатель `std::shared_ptr` и совместное владение. Указатель `std::weak_ptr`. Указатель `intrusive_ptr`.
5. Особенности использования `std::make_unique` и `std::make_shared`. Пользовательские удалители.
6. Использование `std::enable_shared_from_this`.

4. Паттерны и идиомы в проектировании.

1. Идиома CRTP.
2. Идиома PImpl. Особенности использования `std::unique_ptr` в основном интерфейсе.
3. Паттерн проектирования Bridge.
4. Паттерн проектирования Command.
5. Паттерн проектирования Singleton.
6. Паттерн проектирования Strategy.
7. Паттерн проектирования Decorator.
8. Паттерн проектирования Observer.
9. Паттерн проектирования Visitor и двойная диспетчеризация.
10. Создание объектов с помощью фабрик.
11. Идиома Type Erasure.

5. SFINAE и его применение.

1. Правило SFINAE. Применение `std::enable_if`, `std::is_base_of`, `std::integral_constant`.
2. Диспетчеризация дескрипторов. Способы решения проблем перегрузки с универсальными ссылками.

6. Обработка ошибок, тестирование и отладка кода.

1. Возврат ошибки, опциональное значение.
2. Обработка исключений, захват исключений в `std::exception_ptr`, подход `ValueOrError`.
3. Asserts времени компиляции и времени выполнения.
4. Основные подходы к тестированию систем. Unit-тесты через `google testing framework`.
5. Устранение нетестируемых зависимостей в unit-тестах. Mock-, fake-, stub-, dummy-объекты.
6. Отладка и дебаг кода.
7. Неопределённое поведение.

7. Оптимизация программ на C++.

1. Оптимизация строк, идиома COW. Оптимизация алгоритмов, функций, инструкций, использования динамической памяти.
2. Примеры оптимизации кода.
3. Плотная упаковка полей структур.
4. Бинарная сериализация. Protocol buffers.