

## 1. Некоторые особенности C++11.

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 `std::bind`. Использование `std::function` и `std::mem_fn`.

## 2. Семантика перемещения.

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

## 3. Умные указатели.

1. Динамическое выделение памяти. Размещающие операторы `new` и `delete`. Аллокаторы.
2. Указатель `std::unique_ptr` и исключительное владение.
3. Указатель `std::shared_ptr` и совместное владение. Указатель `std::weak_ptr`.
4. Особенности использования `std::make_unique` и `std::make_shared`. Пользовательские удалители.

## 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.
10. Создание объектов с помощью фабрик.
11. Идиома Type Erasure.

## 5. Метапрограммирование.

1. Правило SFINAE. Применение `std::enable_if`, `std::is_base_of`, `std::integral_constant`.
2. Диспетчеризация дескрипторов. Способы решения проблем перегрузки с универсальными ссылками.
3. Вычисления на этапе компиляции при помощи шаблонов.
4. Подход X-macro.

## 6. Обработка ошибок.

1. Возврат ошибки, опциональное значение.
2. Обработка исключений, захват исключений в `std::exception_ptr`, подход `ValueOrError`.
3. Asserts времени компиляции и времени выполнения.

## 7. Некоторые вопросы многопоточного программирования.

1. Многопоточность, класс `std::thread`, свойство `joinable`.
2. Способы синхронизации: `mutex`, условные переменные, `atomic`.
3. Частичная потокобезопасность `std::shared_ptr`.