1. этапы сборки проекта (препроцессинг, компиляция, сборка). рассказать, что на каждом этапе происходит
2. какие бывают библиотеки: динамические и статические. В чем разница, когда какие лучше использовать.
3. есть ли опыт с win API
4. как внутри устроен std::deque (как набор std::vector), чем он отличается от std::list
5. зачем нужны виртуальные методы, где они хранятся и как вообще указатель выбирает подходящую реализацию (это таблица виртуальных функций)
6. как для виртуальной функции явно указать, что в базовом классе нет реализации, а нужно определять метод в классах-наследниках (нужно написать '=null')
7. многопоточность: есть потоки и процессы, в чем разница?
8. как передавать данные между потоками? (разделяемые ресурсы, тут можно вспомнить про мьютексы)
9. как передавать данные между процессорами (ответ: через файлы на жестком диске, через сокеты или через pipe + ГЛОБАЛЬНЫЕ МЬЮТЕКСЫ на уровне операционки)
10. чем std::array отличается от std::vector'a
11. паттерны проектирование (какие знаешь, зачем примеряются, приведи пару примеров)
12. какие бывают типы cast'ов (static\_cast, const\_cast и еще три)
13. как происходит выброс исключений. <https://habr.com/ru/sandbox/28877/>
14. можно ли бросать исключения в конструкторе? а в деструкторе?
15. зачем нужен noexcept? Что будет, если метод помечен как noexcept, но в нем возникло исключение?
16. что такое дедлок, как избежать дедлока. Приведи пример, как написать код на плюсах, чтобы был дедлок
17. почему лучше использовать make\_shared, make\_unique , чем просто конструкторы умных указателей
18. как работает выравнивание структур