

1. Что такое `std::future` и зачем он нужен?
2. Что такое `std::thread` и зачем он нужен?
3. Что будет если кинуть исключение из функции переданной в `std::thread`?
4. Что такое `std::mutex` и зачем он нужен?
5. Что такое SFINAE?
6. Что такое `rvalue-expression`, `lvalue-expression`, `rvalue-reference`, `lvalue-reference`?
7. `std::move` vs `std::forward`. Зачем они нужны?
8. Что такое специализация шаблона? Напишите примеры специализаций.
9. Напишите реализацию шаблона функции `min` от двух аргументов.
10. Что такое ODR? Поясните свой ответ примером кода.
11. Что нужно сделать, чтобы класс стал итератором?
12. Напишите пример `non-type function template` и пример ее вызова.
13. Зачем нужны универсальные ссылки?
14. Что такое лямбда-выражение? Напишите пример.
15. Как работают `precompiled headers`?
16. Что такое функтор? Напишите примеры функторов.
17. Приведите примеры структурных паттернов.
18. Приведите примеры поведенческих паттернов.
19. Приведите примеры порождающих паттернов.
20. Что такое `auto` и как он работает?
21. Что такое `decltype`?
22. Зачем нужны `type traits`'ы?
23. Реализуйте конструкторы и операторы присваивания (включая `move-семантику`) для класса с сырым указателем.
24. Что такое `deadlock`?
25. Какие операторы приведения типов вы знаете? Как они работают?
26. Как работают `constexpr` функции?
27. Как работают `constexpr` переменные?
28. Как работает `constexpr_if`?
29. Что делает `emplace_back` у вектора? Чем отличается от `push_back`?

30. Что такое RAII? Приведите пример.

31. Чем статический полиморфизм отличается от динамического? Приведите пример.

32. Что такое inclusion model и зачем она нужна?

33. `template <typename T> struct Sample{`

`void func(T&& rhs){}`

`}`

Аргумент `rhs` является rvalue-expression'ом ?

Аргумент `rhs` является универсальной ссылкой?