

ПРИМЕР ЭКЗАМЕНАЦИОННОЙ РАБОТЫ
ПО КУРСУ «ПРОЕКТИРОВАНИЕ БОЛЬШИХ СИСТЕМ НА C++»

1. (7 баллов) Для каждого из приведенных утверждений укажите, является ли оно верным (+) или не является(-).

- Функции C++, объявленные как `p noexcept`, дают больше возможности оптимизации, чем функции без такой спецификации.
- Паттерн проектирования Bridge ограничивает платформенные зависимости.
- Функции `constexpr` могут использоваться не так широко, как не-`constexpr`-функции.
- Потоки `std::thread`, из которых выполнено перемещение, находятся в состоянии `joinable`.
- Если класс содержит пользовательский деструктор, то конструктор перемещения не генерируется автоматически.
- Добавление пользовательских удалителей не увеличивает размер `std::unique_ptr`.
- В библиотеке `google testing framework` для определения теста используется макрос `TEST`, определяющий `void` функцию.

2. (4 балла) Что такое `rvalue`-ссылка в C++? Объясните, как работают функции `std::move` и `std::forward`.

3. (3 балла) В чем ошибка в коде:

```
#include <iostream>
#include <vector>
#include <string>
using std::vector;
using std::string;
using std::cout;
using std::endl;

int main() {
    static const vector<string> types = {"A", "B", "C"};
    for (const auto&& s: types) {
        cout << s << endl;
    }
}
```

4. (4 балла) Функция `Do` определена следующим образом:

```
template <class T>
void Do(const std::shared_ptr<T>& a, const std::shared_ptr<T>& b);
```

Опишите проблемы, связанные с таким её вызовом:

```
Do(std::shared_ptr<T>(new T(1)), std::shared_ptr<T>(new T(3)));
```

Перепишите этот вызов так, чтобы устранить эти проблемы.

5. (2 балла) Какой паттерн проектирования расширяет или изменяет функциональность класса без использования механизма наследования?

- (1) decorator,
- (2) command,
- (3) strategy,
- (4) singleton,
- (5) observer.

6. (2 балла) В чем отличие паттерна Bridge от идиомы `PImpl`?

7. (3 балла) Опишите общую идею и схему использования паттерна `Visitor`.

8. (3 балла) Опишите, какие проблемы могут возникнуть при перегрузке для универсальных ссылок.

9. (3 балла) Что такое мок-объекты в тестировании? Для чего они нужны?
 10. (1 балл) Чему равно значение `std::is_arithmetic<int const> :: value` ?
 11. (2 балла) Что напечатает программа?

```
#include <iostream>

template <typename T, int N>
struct A {
    static constexpr int res = N;
};

template <typename T, int N>
struct A<T*, N>: public A<T, N * 2> {};

int main() {
    int* p = new int(1);
    auto x = &p;
    std::cout << A<decltype(x), 2>::res << std::endl;
}
```

12. (3 балла) Для чего нужны объекты `std::optional`? Приведите пример сценария работы с таким объектом.
 13. (4 балла) Можно ли использовать `std::shared_ptr` из нескольких потоков? Объясните, какое взаимодействие потокобезопасно, а какое — нет.
 14. (3 балла) Почему следующий код не скомпилируется?

```
auto f = [](){ return 1; };
f = [](){ return 2; };
```

15. (4 балла) Задан класс

```
class ApplyEach {
private:
    std::atomic<long> counter;
    long init;

public:
    ApplyEach(long n): counter(n), init(n) {}

    template <typename TFunc>
    void operator()(TFunc func) {
        long res = --counter;
        if (res == 0) {
            counter = init;
            func(res);
        }
    }
};
```

Ожидается, что объект класса будет использоваться из нескольких потоков и при вызове оператора `()` каждый раз будет вызываться функция `func`. Объясните, почему это может быть не так, и предложите способ исправления проблем.

16. (2 балла) В программе на C++ определена функция:

```
const int& g();
```

Укажите, к какому классу (lvalue, xvalue, prvalue) относятся следующие выражения:

```
g();
std::move(g);
```