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 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.
- При конструировании объекта `std::string_view` происходит аллокация памяти под массив символов.
- 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.

5. **(1 балл)** Что из нижеследующих является примером идиомы CRTP?

(1) `std::make_unique`,
(2) `std::enable_shared_from_this`,
(3) `std::string_view`,

(4) `std::optional`.

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 mock-объекты в тестировании? Для чего они нужны?
10. **(2 балла)** Как оптимизировать код?

```
class TRequestProcessor : public IProcessor {
    // ...
    int count;
  public:
    // ...
    int GetCount () {
        return count;
    }
    bool GetFeatureName(std::string& fn, bool isOpt) {
        fn = isOpt ? "RequestFeature" : "";
        return isOpt;
    }
};
```

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. **(3 балла)** Для чего нужны объекты `std::exception_ptr`? Приведите пример сценария работы с таким объектом.
14. **(3 балла)** Почему следующий код не скомпилируется?

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

15. **(3 балла)** Метод `resize` у вектора в случае реаллокации предоставляет строгую гарантию безопасности исключений. Объясните, что это означает.

Пусть определен класс:

```
class A {
  public:
    A() { /*...*/ }
    A(const A&) { /*...*/ }
    A(A&&) { /*...*/ }
};
```

и клиентский код устроен следующим образом:

```
void Get(size_t, A&) { /*...*/ }
int main() {
    std::vector<A> v;
    A a;
    for (size_t i = 0; i < 100; ++i) {
        Get(i, a);
        v.push_back(a);
    }
}
```

При выполнении такого кода конструктор перемещения никогда не будет вызван в `main`, но будет вызван некоторое количество раз конструктор копирования. Как можно изменить код класса, чтобы заменить некоторые вызовы конструктора копирования на конструктор перемещения без изменения клиентского кода?

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(9);
```