

ПРИМЕРЫ ВОПРОСОВ И ЗАДАЧ К ЭКЗАМЕНУ
ПО КУРСУ «ПРОЕКТИРОВАНИЕ БОЛЬШИХ СИСТЕМ НА C++»

1. 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 `non-constexpr` функции.
- Если класс содержит пользовательский деструктор, то конструктор перемещения не генерируется автоматически.
- Добавление пользовательских удалителей не увеличивает размер `std::unique_ptr`.

2. Приведите пример использования указателя `std::weak_ptr`.

3. Укажите строчки, в которых допущены ошибки перекрытия:

```
class A {
public:
    void f();
    virtual void g(int x);
    virtual float h();
};

class B: public A {
public:
    void f() const override;
    virtual void g(unsigned int x) override;
    float h() override;
};
```

4. 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(new T(1), new T(3));
```

Перепишите этот вызов так, чтобы устранить эти проблемы.

5. Какой паттерн проектирования расширяет или изменяет функциональность класса без использования механизма наследования?

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

6. Пусть имеется интеллектуальный указатель для объектов, создаваемых в некоторой заданной куче, а также интеллектуальный указатель для объектов, создаваемых в некоторой управляемой памяти и т.д. Все эти указатели отличаются только по способу конструирования и удаления объектов. Какой паттерн следует применить для создания одного общего указателя, делегируя детали удаления различным политикам, число которых не ограничено?

- (1) Стратегия
- (2) Команда
- (3) Декоратор
- (4) Фабрика
- (5) Посетитель

7. В чем отличие паттерна Bridge от идиомы PImpl?
8. Опишите общую идею и схему использования паттерна Visitor.
9. Опишите, какие проблемы могут возникнуть при перегрузке для универсальных ссылок.
10. Чему равно значение `std::is_arithmetic<int const> :: value` ?
11. В программе тестируется компонент логирования данных пользователей сайта, который, в частности, умеет считать число пользователей сайта для отображения статистики. Вместо обращения к сложному хранилищу данных реальных пользователей `UserStorage.GetUserCount()` этот компонент в `unit`-тесте запоминает только те идентификаторы пользователей, которые появлялись в тесте, сохраняет их в `std::unordered_map`, и возвращает его `size`. К какому из перечисленных типов устранения зависимостей относится указанный подход?
 - dummy,
 - fake,
 - stub,
 - mock
12. Для чего нужны объекты `std::optional`? Приведите пример сценария работы с таким объектом.
13. К какому классу относится выражение, отмеченное (*)?

```
int x;  
static_cast<int&&>(x); // {(*)}
```

- lvalue
 - xvalue
 - prvalue
14. Программа на C++ работает под 64-bit платформой (указатели на все типы имеют размер 8 байтов). Какой размер (`sizeof(TObject)`) имеет структура `TObject`?

```
struct TItem {  
    char ptr;  
    int16_t index;  
};  
  
struct TObject {  
    char chr;  
    TItem item;  
};
```