«Проектирование больших систем на C++» Студент: _____

Экзаменационная работа (вариант X-01-24) Группа: _____

(1) *(4 балла)* Для каждого из приведенных утверждений укажите, является ли оно верным (+) или не является (–). Не отмеченные утверждения и утверждения, помеченные знаком "?", не будут учитываться при оценивании

(a) Нельзя написать `std::unique_ptr<int> p = new int;` (будет ошибка компиляции).
(b) Объекты `constinit` являются константными.
(c) Объекты `constexpr` инициализируются объектами, значения которых известны во время компиляции.
(d) Объявления псевдонимов `using` поддерживают шаблонизацию.

(2) *(4 балла)* Что такое `constexpr` и `consteval` функции? Как они работают и в чём их отличия?

(3) *(4 балла)* Что такое `std::string_view`? Приведите пример, когда его использование может быть полезным.

(4) *(4 балла)* Для чего предназначен `std::forward` и как он работает?

(5) *(4 балла)* Какую семантику владения реализует умный указатель `std::unique_ptr`? Что происходит при его копировании и перемещении? Чему равен его размер?

(6) *(4 балла)* Для чего предназначена библиотека Protobuf? В чём заключаются её основные достоинства и недостатки?

(7) *(7 баллов)* Опишите общую схему и идею паттерна проектирования Декоратор (Decorator). В каких случаях этот паттерн может использоваться?

(8) *(4 балла)* Что такое идиома CRTP? Для чего она может использоваться? Приведите пример любого класса (в виде кода на C++), в котором используется данная идиома.

(9) *(2 балла)* Как называется паттерн проектирования, который позволяет объектам автоматически получать уведомления об изменениях, происходящих в других объектах?

Вопрос сгенерирован при помощи ChatGPT

(10) *(2 балла)* В программе есть класс, который отправляет SMS-уведомления пользователям. Вместо реальной отправки SMS-сообщений через провайдера услуг этот класс в unit-тесте использует объект, который сохраняет отправленные уведомления в специальный список для последующей проверки. К какому из перечисленных типов устранения зависимостей относится указанный подход?

(a) dummy
(b) stub
(c) fake
(d) mock

Вопрос сгенерирован при помощи ChatGPT

(11) *(3 балла)* Что напечатает программа?

```
#include <iostream>
#include <functional>
```

```
int f(int x, int y, int z) {
    return -2 * x - y - 2 * z;
}

int g(int x, int y) {
    return -x * x + 3 * y;
}

int main() {
    auto h = std::bind(
        g,
        std::placeholders::_3,
        std::bind(f, -1, std::placeholders::_2, std::placeholders::_3)
    );

    std::cout << h(-2, 3, 3) << std::endl;

    return 0;
}
```

(12) *(3 балла)* Какая проблема в работе с памятью есть в следующем коде? Как её можно исправить?

```
void f(std::shared_ptr<C> ptr, int param);

void g() {
    C* ptr = new C;
    for (int i = 0; i < 5; ++i) {
        f(std::shared_ptr<C>(ptr), i);
    }
}
```

(13) *(3 балла)* Пусть определена шаблонная функция `f`, имеющая следующий прототип:

```
template <typename T>
void f(T& param);
```

Чему равны тип параметра `T` и тип аргумента `param` в следующих вызовах функции `f`:

```
const float num = 61;
char str[] = "foo";
f(num);
f(str);
```

(14) *(3 балла)* Что напечатает следующая программа, работающая под 64-bit платформой (указатели на все типы имеют размер 8 байтов)?

```
#include <cstdint>
#include <iostream>

struct S {
    char a : 8;
    int64_t b : 26;
    uint32_t c : 9;
    uint32_t d : 24;
};

int main() {
```

```
    std::cout << sizeof(S) << " " << alignof(S) << std::endl;
    return 0;
}
```

(15) *(3 балла)* Что напечатает программа?

```
#include <iostream>

template <typename T>
int f(T x) {
    return x;
}

template <typename T1, typename... T2>
int f(T1 x, T2... y) {
    return x - f(y...);
}

int main() {
    std::cout << f(4, 1, 1, 2, 2) << std::endl;
    return 0;
}
```