

1 Классы и пространства имён

Класс:

```
class Monster
{
private:
    float x, y, z;
    int health, is_alive, power;
public:
    void hurt(int damage);
    void heal(int heal_power);
};
void Monster::hurt(int damage)
{
    m->health -= damage;
    if (m->health < 0)
        m->is_alive = 0;
}
void Monster::heal(int heal_power)
{
    m->health += heal_power;
}
```

Пространство имён:

```
namespace Foo
{
    int a;
    int f(int x)
    {
        std::cout << "Did something";
    }
    namespace Bar
    {
        float b;
        int g(int x)
        {
            std::cout << "Did other thing";
        }
    }
}
Foo::a = 12345;
Foo::Bar::g();
```

Чтобы не писать пространство имён перед идентификатором можно воспользоваться директивой using:

```
using namespace Foo;
using namespace std; // std - пространство имён стандартной библиотеки
```

2 Ввод вывод в C++

В C++ можно пользоваться функциями printf и scanf, но обычно используются специальные объекты cout и cin:

```
#include <iostream>
// ...
std::cout << выражение << std::endl;
std::cin >> переменная;
```

Задачи:

1. **Hello world C++** Написать программу HelloWorld используя библиотеку iostream. Скомпилировать с помощью g++.
2. **cin - cout** Написать программу, которая запрашивает радиус круга и вычисляет его площадь.
3. **Namespaces** Создать пространство имён по имени MySpace. Объявить в нём функцию, которая печатает "Hello world". Вызвать эту функцию из функции main().
4. **Complex** Написать класс комплексного числа. Перегрузить операторы сложения, вычитания, умножения и деления.
5. **Matrix** Написать класс матрицы. Перегрузить операторы сложения, вычитания, умножения.