

Здравствуйте, Михаил Иванович!

1. Отличие структуры от класса.

На практике отличие структуры от класса заключается в следующем: структуры, как правило, используют в качестве объединения данных, а классы - в качестве объединения данных и функций. в С++ структуры также обладают возможностью включать в себя функции (что, правда, неверно в отношении языка С). Различия в синтаксисе между структурами и классами минимальны, поэтому теоретически можно использовать одно вместо другого. Однако структуры обычно используют только для хранения данных, а классы - для хранения данных и функций. Одним из достоинств классов является возможность определять сложение и другие операции над пользовательскими типами данных, в отличие от структур. Формально разница между структурами и классами заключается лишь в том, что по умолчанию все члены класса являются скрытыми, а все члены структуры - открытыми.

```
class foo{
    private:
        int data1;
    public:
        void func();
};
```

```
struct foo{
    int data1;
};
```

Если мы хотим при помощи структуры выполнять те же действия, что и с использованием класса, мы можем отменить действие принятого по умолчанию ключевого слова `public` словом `private` и расположить открытые поля структуры до слова `private`, а закрытые поля - после слова `private`:

```
struct foo{
    void func();
    private:
        int data1;
};
```

2. Как искать классы.

Классы необходимы для объектно-ориентированного программирования. Оно обеспечивает значительное сходство между объектами реального мира, моделируемыми программой, и объектами классов С++. Объекты классов могут представлять игральные карты,

геометрические фигуры, домашних животных. Если объект представляет игральную карту, то с помощью описания класса мы наделяем

объект полями для хранения достоинства карты, ее масти. Если же объект домашнее животное, то с помощью описания класса объект наделяется полями для хранения характеристик животного (как оно выглядит, какое поведение, чем питается, кличка, возраст).

3. Что такое namespace.

Namespace - это группа взаимосвязанных функций, переменных, констант, классов, объектов и других компонентов программы.

std::cout - вывод данных в терминал

std- пространство имен (namespace), в котором находится программа cout. Пространства имён нужны, чтобы логически связывать части программы. Пространство имен представляет блок кода, который содержит набор компонентов (функций, классов и т.д.) и имеет некоторое имя, которое прикрепляется к каждому компоненту из этого пространства имен.