

Классы и структуры являются конструкциями, в которых пользователь определяет собственные типы. Классы и структуры могут включать данные-члены и функции-члены, позволяющие описывать состояние и поведение данного типа.

В C++ эти конструкции идентичны, за исключением того факта, что структуры по умолчанию открыты для доступа, а классы — закрыты.

1.) Отличие структуры от класса

Класс состоит обычно из двух объектов – набор данных и набор функций. Одно из отличий класса от структуры – не ко всем переменным из класса можно обращаться извне, а только переменным из *public*.

Структура может содержать в себе некоторые переменные обычного типа (*int, double, bool, char, string* и т.д.), так же массивы и векторы, функции. С помощью структур удобно создавать новые переменные, являющиеся объединением некоторых переменных стандартного типа. Ко всем переменным из структуры можно обращаться извне.

СТРУКТУРЫ	КЛАССЫ
Ключевое слово для класса: struct	Ключевое слово для класса: class
Доступ по умолчанию: public (открытый).	Доступ по умолчанию: private (закрытый).
Нет ограничений на использование	Нет ограничений на использование

2.) Как искать классы?

Один из способов поиска классов — документация. Она содержит описание классов и функций, доступных в библиотеках языка.

Ещё один способ — интегрированная среда разработки (*IDE*), например, *Visual Studio* или *Code::Blocks*.

3.) Что такое *namespace*

Пространство имен — это декларативная область, в рамках которой определяются различные идентификаторы (имена типов, функций, переменных, и т. д.). Пространства имен используются для организации кода в виде логических групп и с целью избежания конфликтов имен, которые могут возникнуть, особенно в таких случаях, когда база кода включает несколько библиотек. Все идентификаторы в пределах пространства имен доступны друг другу без уточнения. Идентификаторы за пределами пространства имен могут обращаться к членам, используя полное имя для каждого идентификатора, например `std::vector<std::string> vec;`, или с помощью объявления **using** для одного идентификатора (`using std::string`) или директивы **using** для всех идентификаторов в пространстве имен (`using namespace std;`). Код в файлах заголовков всегда должен содержать полное имя в пространстве имен.

Также с пространством имен напрямую связаны следующие понятия:

1.) Директивы **using**

2.) Объявление пространств имен и их членов

3.) *Глобальное пространство имен*

4.) *Пространство имен **std***

5.) *Вложенные пространства имен*

6.) *Псевдонимы пространств имен*

7.) *Анонимные или безымянные пространства имен*