

Сказка

212 Скорбилин Илья

9 сентября 2023 г.

1 Отличие структуры от класса

Для начала хочется отметить, что в языке C существуют только структуры. Там структуры применяются для удобного хранения и использования набора данных. По сути это просто список переменных или других структур инкапсулированный в одном месте. В C++ появляется такое понятие как класс. Он используется для представления набора данных и способов взаимодействия с ними (поля и методы класса). Если сравнивать структуру и класс в таких представлениях, то различие очевидно. В классе есть функции, взаимодействующие напрямую с данными из класса, а в структуре хранятся только сами данные. К тому же в классах есть возможность скрыть поля или методы класса, чего в структурах сделать было нельзя. Но есть одно большое НО. В языке C++ различия между структурой и классом практически исчезли, и со структурой можно делать практически то же самое, что и с классом. Разными остались лишь ключевые слова и тот факт, что в классе по умолчанию данные и методы скрыты (`private`), а в структуре — открыты (`public`). Но отдавая дань традиции, разработчики и по сей день используют структуру только для хранения данных, а если требуется реализовать ещё и член-функции для этих данных, то обычно используют классы.

2 Как искать классы?

Бывает при разработке программы неожиданно обнаруживается, что некоторые объекты исследования схожи между собой по некоторым свойствам и поведению. (Например люди или машины разных марок и моделей или детали разной формы) Тогда, чтобы укоротить код и упростить его читаемость, целесообразно создать класс, включающий в себя общие

для всех объектов свойства (поля класса) и поведение (методы класса). Иногда даже среди объектов одного класса можно найти группы с одинаковыми свойствами. Тогда имеет место для применения механизма наследования классов от одного базового. Такие производные классы имеют все поля и методы базового класса, но могут иметь и свои собственные. Чтобы найти классы, нужно найти нечто общее у группы объектов.

3 Что такое namespace?

Пространство имён или namespace — некоторое множество, под которым подразумевается модель, абстрактное хранилище или окружение, созданное для логической группировки уникальных идентификаторов (то есть имён). Идентификатор, определённый в пространстве имён, ассоциируется с этим пространством. Один и тот же идентификатор может быть независимо определён в нескольких пространствах. Таким образом, значение, связанное с идентификатором, определённым в одном пространстве имён, может иметь (или не иметь) такое же значение, как и такой же идентификатор, определённый в другом пространстве. Чтобы обратиться к имени, принадлежащему конкретному пространству имён необходимо написать название пространства, поставить :: (оператор разрешения видимости), а затем уже написать необходимое имя. Так же, если в тексте программы часто используется какое-то одно пространство имён, то проще написать в начале программы строчку: `using namespace "имя пространства имён"`. Тогда не придётся каждый раз писать длинную приставку к именам. В C++ есть стандартное пространство имён `std`.