Тема 2.1 Виды классов

Классы и объекты в C++ являются основными концепциями объектно-ориентированного программирования — ООП. Объектно-ориентированное программирование — расширение структурного программирования, в котором основными концепциями являются понятия классов и объектов. Основное отличие языка программирования C++ от C состоит в том, что в C нет классов, а следовательно язык C не поддерживает ООП, в отличие от C++.

Чтобы понять, для чего же в действительности нужны классы, проведём аналогию с каким-нибудь объектом из повседневной жизни, например, с велосипедом. Велосипед — это объект, который был построен согласно чертежам. Так вот, эти самые чертежи играют роль классов в ООП. Таким образом классы — это некоторые описания, схемы, чертежи по которым создаются объекты. Теперь ясно, что для создания объекта в ООП необходимо сначала составить чертежи, то есть классы. Классы имеют свои функции, которые называются методами класса. Передвижение велосипеда осуществляется за счёт вращения педалей, если рассматривать велосипед с точки зрения ООП, то механизм вращения педалей — это метод класса. Каждый велосипед имеет свой цвет, вес, различные составляющие — всё это свойства. Причём у каждого созданного объекта свойства могут различаться. Имея один класс, можно создать неограниченно количество объектов (велосипедов), каждый из которых будет обладать одинаковым набором методов, при этом можно не задумываться о внутренней реализации механизма вращения педалей, колёс, срабатывания системы торможения, так как всё это уже будет определено в классе. Разобравшись с назначением класса, дадим ему грамотное определение.

Классы в C++ — это абстракция описывающая методы, свойства, ещё не существующих объектов. **Объекты** — конкретное представление абстракции, имеющее свои свойства и методы. Созданные объекты на основе одного класса называются экземплярами этого класса. Эти объекты могут иметь различное поведение, свойства, но все равно будут являться объектами одного класса. В ООП существует три основных принципа построения классов:

1. **Инкапсуляция** — это свойство, позволяющее объединить в классе и данные, и методы, работающие с ними и скрыть детали реализации от пользователя.
2. **Наследование** — это свойство, позволяющее создать новый класс-потомок на основе уже существующего, при этом все характеристики класса родителя присваиваются классу-потомку.
3. **Полиморфизм** — свойство классов, позволяющее использовать объекты классов с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.

Сокрытие внутренней структуры объекта от внешнего пользователя

Инкапсуля́ция — свойство языка программирования, позволяющее пользователю не задумываться о сложности реализации используемого программного компонента (что у него внутри?), а взаимодействовать с ним посредством предоставляемого интерфейса (публичных методов и членов), а также объединить и защитить жизненно важные для компонента данные. При этом пользователю предоставляется только спецификация (интерфейс) объекта. Пользователь может взаимодействовать с объектом только через этот интерфейс. Реализуется с помощью ключевого слова: public. Пользователь не может использовать закрытые данные и методы. Реализуется с помощью ключевых слов: private, protected, internal. Инкапсуляция — один из четырёх важнейших механизмов объектно-ориентированного программирования (наряду с абстракцией, полиморфизмом и наследованием). Сокрытие реализации целесообразно применять в следующих случаях: предельная локализация изменений при необходимости таких изменений, прогнозируемость изменений (какие изменения в коде надо сделать для заданного изменения функциональности) и прогнозируемость последствий изменений.

Методы класса

Классы в программировании состоят из свойств и методов. Свойства — это любые данные, которыми можно характеризовать объект класса. В нашем случае, объектом класса является студент, а его свойствами — имя, фамилия, оценки и средний балл.

Методы — это функции, которые могут выполнять какие-либо действия над данными (свойствами) класса.

- **Методы** класса — это его функции.
- **Свойства** класса — его переменные.

Привилегии доступа

C++дает вам возможность объявить базовый класс, который инкапсулирует имена своих свойств, данных, методов и событий. Помимо способности выполнять свою непосредственную задачу объектные методы получают определенные привилегии доступа к значениям свойств и данных класса.

Каждое объявление внутри класса определяет привилегию доступа к именам класса в зависимости от того, в какой секции имя появляется. Каждая секция начинается с одного из ключевых слов: **private, protected и public.**

class className

private:

<приватные члены данных> <приватные конструкторы> <приватные методы>

protected:

<защищенные члены данных> <защищенные конструкторы> <защищенные методы>

public:

<общедоступные свойства> <общедоступные члены данных> “збщедоступные конструкторы> <общедоступный деструктор> общедоступные методы>

Объявление базового класса.

Таким образом, объявление базового класса на C++ предоставляет следующие права доступа и соответствующие области видимости:

• *Приватные* **private** имена имеют наиболее ограниченный доступ, разрешенный только методам данного класса. Доступ производных классов к приватным методам базовых классов запрещен.

• *Защищенные* **protected** имена имеют доступ, разрешенный методам данного и производных от него классов.

• *Общедоступные* **public** имена имеют неограниченный доступ, разрешенный методам всех классов и их объектов.