**Использование принципов ООП при разработке приложений.**

Описание обобщённого интерфейса и выделение наиболее важных характеристик и информации об объекте осуществляется с помощью принципа абстракции. Изначально создаётся некий абстрактный класс, в котором описывается некоторая обобщённая информация о свойствах предметов или объектов. Однако, абстракция также может быть многоуровневой. На основе абстрактного класса могут быть созданные новые типы объектов с различающимся функционалом.

С помощью инкапсуляции открывается возможность для определения атрибутов и методов, которые будут доступны для открытого доступа и, возможно, изменения, или которые будут являться описанием внутренней реализации объекта – такие атрибуты и методы должны быть скрыты и недоступны для изменения. Подобные ограничения можно накладывать, используя модификаторы доступа к полям и методам: private, protected, public и default.

Для описания новых классов используются уже существующие классы, что реализуется благодаря принципу наследования. Производный класс может использовать функционал, описанный в классах родителях, и при этом расширять и добавлять свои возможности. Использование этого принципа позволяет значительно уменьшить объём кода.

Принцип полиморфизма позволяет переопределять методы, определённые в базовом классе, без изменения особенностей этих методов. По сути, происходит замена метода, и выбор реализации в процессе выполнения программы падает именно на новый метод, переопределённый в производном классе.