**1. Расписать в текстовом документе зачем нужны интерфейсы и их отличие от классов.**

Класс – структурный элемент приложения, описывающий какую-либо сущность в пространстве имен. В классе содержатся поля, методы (в т.ч. конструкторы и свойства), а так же события.

Интерфейс – аспект языка, служащий для того, чтобы формировать интерфейс взаимодействия класса (где интерфейс взаимодействия класса – совокупность всех публичных членов класса).

Интерфейс содержит в себе чистую абстракцию: сигнатуры методов, свойств, событий и индексаторов. Не может содержать реализацию. Не поддерживает создание полей.

Интерфейс нужен для того, чтобы обойти запрет на множественное наследование. Интерфейсы при этом не наследуются, а реализуются. Допустима множественная реализация.

**2. Расписать в текстовом документе зачем нужны обобщение и привести несколько примеров их использования.**

Обобщения (generics) – переменный тип данных, которыми манипулирует класс или метод. Обобщения позволяют описывать данные и алгоритмы так, что их можно применять к любым типам данных, при этом не меняя сам алгоритм. Обобщениями могут быть поля, методы, классы, структуры, интерфейсы и делегаты. Generics помогают упростить программный код, избавляют от необходимости упаковки и распаковки типов данных.

Пример1: есть метод, которому на вход подаются 2 переменные типа int (static void Swap (ref int a, ref int b)). В ходе разработки появляется необходимость использовать этот метод для других типов данных. Его можно перегрузить, но это усложнит код. Поэтому проще привести его к обобщенному виду (static void Swap <T> (ref T a, ref T b)).

Пример2: похожая ситуация может быть рассмотрена на примере классов, допустим мы создаем класс BancAccount, у которого есть поля Id и Sum, но нам точно не известно, какой тип данных должен быть у Id: строковый или числовой, тут на помощь снова могут прийти обобщения, с помощью которых можно спокойно дальше прописывать логику класса.