**1. Классы и структуры, отличия.**

Основное отличие классов от структур, на мой взгляд, заключается в том, что класс представляет собой тип-ссылку, в то время как структура является типом по значению. Т.е. когда создается объект класса и сохраняется в некую фиксированную переменную, эта переменная содержит лишь ссылку на объект, который располагается в куче. В отличие от структуры, переменная которой содержит как раз непосредственно объект, в этом случае располагающийся в стеке. Основное различие при работе с ними заключается в том, что в случае с классом, если другой переменной присваивается значение некой фиксированной, то дальнейшие действия, совершаемые над новой переменной, будут отражаться на старой, т.к. они ссылаются на один объект. При работе со структурой при присваивании создается копия объекта, и при дальнейшей работе над новой переменной изменения никак не затрагивают старую.

**2. Чем отличаются константы и поля, доступные только для чтения.**

**3. Когда использовать StringBuilder предпочтительнее, чем string**

Основное отличие стинга от стрингбилдера, на мой взгляд, заключается в том, что при каком-либо изменении строки в случае работы с стрингом происходит автоматическое создание нового объекта, при этом стрингбилдер позволяет решить данную проблему. Исходя из этого, можно предположить, что при работе с большим количеством изменений в строках разумнее использовать стрингбилдер, т.к. такое количество суммирования стрингов может дать большой удар по памяти.

**4. Основные парадигмы ООП.**

Наследование, полиморфизм, инкапсуляция.

**5. Какой интерфейс должна поддерживать коллекция чтобы можно было ее использовать в foreach**

IEnumerable

**6. В чем различие между ключевыми словами "ref" и "out"?**

ref используется при необходимости передать параметры по ссылке (здесь уместно говорить о параметрах, представляющих собой типы по значению). Модификатор out используется в случае, когда нужно вернуть параметры не через оператор return, а через выходной параметр. Этот способ предоставляет возможность наличия не одного, а нескольких выходных параметров, что может быть удобно при решении определенных задач.

**7. Конструктор, виды конструкторов**

Конструктор – это специальный метод класса, не имеющий возвращаемого значения, при этом имя которого совпадает с названием класса/структуры. Вызывается в случае создания нового объекта и имеет единственную функцию – корректно инициализировать поля класса/структуры.

Выделяют как две основные группы конструкторы по умолчанию и конструкторы с параметрами. При отсутствии какого-либо конструктора, при создании нового объекта автоматически вызывается конструктор по умолчанию. Однако при наличие хотя бы одного прописанного в коде конструктора, конструктор по умолчанию при создании объекта вызываться уже не будет.