1. Классы и структуры, отличия.

Структуры это размерный тип, а класс-ссылочный. Все структурные типы неявно наслед. От System.ValueType, не бывают абстрактными и всегда неявно запечатаны(sealed). Структуры не могут содержать явных конструкторов без параметров. Модификатор доступа поля структуры не может быть protected.

1. Чем отличаются константы и поля, доступные только для чтения.

Значение константы можно установить только один раз. Используется ключевое слово const. При использовании констант надо помнить , что объявить ы их можем только один раз и к моменту компиляции они должны быть опреределны.

Константа это статическое поле – необязателно создавать объект класса с помощью конструктора. Обратиться к ней можно с помощью имени класса.  
Поля для чтения инициализируются при объявлении либо на уровне класса, либо инициализировать или изменять их можно в конструкторе. В других местах иниц. или изменять их нельзя. Объявляются с ключевым словом readonly.

1. Когда использовать StringBuilder предпочтительнее, чем string

String невозможно изменить после того, как строка создана. Если мы захотим ее изменить, то в итоге мы получим не старую строку с изменениями, а новую строку, которая будет содержать в себе измененную старую.

StringBuilder наоборот выделяет память по мере необходимости. И если мы захотим изменить строку, то она измениться по-настоящему.

1. Основные парадигмы ООП.

Абстракция — в объектно-ориентированном программировании это придание объекту характеристик, которые отличают его от всех других объектов, четко определяя его концептуальные границы.

Инкапсуляция

Полиморфи́зм — возможность объектов с одинаковой спецификацией иметь различную реализацию

Наследование- позволяет описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом

1. Какой интерфейс должна поддерживать коллекция чтобы можно было ее использховать в foreach

IEnumerable

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

Модификатор ref - Для передачи по ссылке в С# предусмотрен модификатор параметра ref.

Модификатор out

Аргумент out похож на аргумент ref за исключением следующих аспектов:

он не нуждается в присваивании значения перед входом в функцию;

ему должно быть присвоено значение перед выходом из функции.

Модификатор out чаще всего используется для получения из метода нескольких

возвращаемых значений

7.Конструктор, виды конструкторов

Конструкторы позволяют программисту задавать значения по умолчанию, ограничивать число установок и писать код, который является гибким и удобным для чтения.

Виды:

1. конструктор с параметрами;
2. [конструктор по умолчанию](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80_%D0%BF%D0%BE_%D1%83%D0%BC%D0%BE%D0%BB%D1%87%D0%B0%D0%BD%D0%B8%D1%8E), не принимающий аргументов;
3. именованный конструктор — функция, предполагающая явный вызов по имени, работающая как конструктор
4. [конструктор копирования](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D0%BE%D1%80_%D0%BA%D0%BE%D0%BF%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F) — конструктор, принимающий в качестве аргумента объект того же класса (или ссылку из него);
5. конструктор преобразования — конструктор, принимающий один аргумент (эти конструкторы могут вызываться автоматически для преобразования значений других типов в объекты данного класса).
6. конструктор перемещения