**IEnumberable<int>** GetEnumerator(); **IDisposible** Dispose();

**IOC –** принцип ООП, служит для уменьшения связности слоев приложения, Принцип Заключается в том что каждый слой приложения работает с нижестоящим не напрямую а опосредованно(через слой абстракции). В результате позволяет легко заменить реализацию слоев на новую, не затронув работу вышестоящих слоев.

Action Predicate Func(select, where, Join) – проверить

**Интерфейс** – все методы имеют только объявление но без реализации

**Asp net.** ***Request*** *–* ***Middleware****-****Routing****-****Controller Initialization***-Action Method Execution (Authoriz Filters, Controller Creation, Model Binding,Action Filter, Action Execution)

*-****Result Execution*** *–* ***View Rendering*** *-****Response***

**Сборщик мусора –** справляется с управляемыми объектаме но не знает что делать с неуправляемыми.Разработчик должен сам реализовать отчистку. Сборщик вызывает Finalize, он определен в базовом через дескриптом. Finalize не гарантирует сборку. Сборщик мусора когда видит что на участок памяти нет ссылок, отчисттит. Сборщика запускается не сразу.Он запускается например когда CLR обнаружит потребность в дополнительной памяти. Три поколения 0, 1, 2. 1- Когда пережили уже сборку мусора. 0 – удаляется с наименьшими издержками

**Generic** – **упаковка** object obj = 1; Значимый в ссылочный. **Распаковка** int Id = (int) obj; Ссылочный в значимый распаковка. Снижается производительность.ArrayList, SortedList, Stack, Queue, Hashtable, Gen- List, Dictionary, Queue, Stack

**http** - протокол передачи гипертекста.  Основой HTTP является технология «клиент-сервер».  HTTP не сохраняет своего состояния.  HTTP не сохраняет своего состояния. Компоненты, использующие HTTP, могут самостоятельно осуществлять сохранение информации о состоянии, связанной с последними запросами и ответами (например, «[куки](https://ru.wikipedia.org/wiki/HTTP_cookie)» на стороне клиента, «сессии» на стороне сервера

**Паттернами проектирования** (Design Patterns) называют решения часто встречающихся проблем в области разработки программного обеспечения

**Блокировка** гарантирует, что один поток не будет перехватывать другой поток, который выполняет часть кода. Таким образом, оператор блокировки заставит поток ждать, блокировать до освобождения объекта.

**Делегат** – это объект, инкапсулирующий ссылку на метод.С помощью делегатов можем вызвать методы; delevate void Message();

**Анонимные методы** – Лямбда упрощенная запись аноноимных методов.

**Структуры** – в стеке и не могут наследоваться ???

**Перегрузка** Add(), Add(string Name), **Переопределение** Override, **Сокрытие** new  
Иными словами это тип, который ссылается на метод. Как только делегату назначен метод, он начинает работать точно также как и этот метод. Метод Делегат, может быть использован точно также как и любой другой метод с параметрами и возвращать значение.

 What are the uses of delegates in C#? Callback Mechanism; Asynchronous Processing; Abstract and Encapsulate method; Multicasting

Если конструктор объявлен как статический, то он будет вызван только один раз для всего числа экземпляров класса. Статический конструктор инициализирует статические поля класса.

Абстрактрый класс- экземпляр не можем создать,