1. **Что такое LINQ?**

**LINQ** (Language-Integrated Query) представляет простой и удобный язык запросов к источнику данных. В качестве источника данных может выступать объект, реализующий интерфейс IEnumerable (например, стандартные коллекции, массивы), набор данных DataSet, документ XML. Но вне зависимости от типа источника LINQ позволяет применить ко всем один и тот же подход для выборки данных.

1. **В чем разница между отложенными операциями и не отложенными операциями LINQ to Object?**

Есть два способа выполнения запроса LINQ: **отложенное** и **немедленное** выполнение.

При отложенном выполнении LINQ-выражение не выполняется, пока не будет произведена итерация или перебор по выборке. Рассмотрим отложенное выполнение

Немедленное выполнение запроса

С помощью ряда методов мы можем применить немедленное выполнение запроса. Это методы, которые возвращают одно атомарное значение или один элемент. Например, Count(), Average(), First() / FirstOrDefault(), Min(), Max() и т.д. Например, метод Count() возвращает числовое значение, которое представляет количество элементов в полученной последовательности. А метод First() возвращает первый элемент последовательности. Но чтобы выполнить эти методы, вначале надо получить саму последовательность, то есть результат запроса, и пройтись по ней циклом foreach, который вызывается неявно внутри структуры запроса.

1. **Что такое лямбда-выражения?**

Лямбда-выражения представляют упрощенную запись анонимных методов. Лямбда-выражения позволяют создать емкие лаконичные методы, которые могут возвращать некоторое значение и которые можно передать в качестве параметров в другие методы.

Ламбда-выражения имеют следующий синтаксис: слева от лямбда-оператора => определяется список параметров, а справа блок выражений, использующий эти параметры: (список\_параметров) => выражение. Например:

1. **Как используется операция Where в LINQ to Object?**

**Where**: определяет фильтр выборки

1. **Как используется операция Select ?**

**Select**: определяет проекцию выбранных значений

1. **Как используются операции Take, Skip?**

**Take**: выбирает определенное количество элементов

**Skip**: пропускает определенное количество элементов

1. **Как используется операция Concat ?**

**Concat**: объединяет две коллекции

1. **Как используется операция OrderBy?**

**OrderBy**: упорядочивает элементы по возрастанию

1. **Как используется операция Join?**

**Join**: соединяет две коллекции по определенному признаку

**10.Как используются операции Distinct, Union, Except и Intersect?**

* **Distinct**: удаляет дублирующиеся элементы из коллекции
* **Except**: возвращает разность двух коллекцию, то есть те элементы, которые содератся только в одной коллекции
* **Union**: объединяет две однородные коллекции
* **Intersect**: возвращает пересечение двух коллекций, то есть те элементы, которые встречаются в обоих коллекциях

1. **Как используются операции First, Last, Any, All и Contains?**

**Contains**: определяет, содержит ли коллекция определенный элемент

**Last**: выбирает последний элемент коллекции

**All**: определяет, все ли элементы коллекции удовлятворяют определенному условию

1. **Как используются операции Count, Sum, Min и Max, Average?**

**Count**: подсчитывает количество элементов коллекции, которые удовлетворяют определенному условию**Sum**: подсчитывает сумму числовых значений в коллекции

**Average**: подсчитывает cреднее значение числовых значений в коллекции

**Min**: находит минимальное значение

**Max**: находит максимальное значение

**13. Что выведет на экран данный код?**

**class Test**

**{**

**public static void Main()**

**{**

**List list = new List();**

**list.AddRange(new int[] { 3, 1, 4, 8, 10, 4 });**

**List some = list.FindAll(i => (i>=9));**

**foreach (int x in some) Console.Write(x);**

**} }**

Выведет 10