1. Что такое LINQ?

LINQ (Language-Integrated Query) представляет простой и удобный язык запросов к источнику данных.

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

Разница между отложенными и неотложенными операциями в LINQ to Objects связана с моментом выполнения запроса и получения результатов.

Неотложенные операции (Immediate Operations): Выполняются немедленно при вызове операции LINQ. Результаты операции сразу же возвращаются как коллекция или значение. Примеры неотложенных операций включают ToList(), ToArray(), Count(), First(), Last(), Single() и т. д.

Отложенные операции (Deferred Operations): Отложенные операции не выполняются немедленно при вызове операции LINQ, а создаются как запрос, который позднее будет выполнен при доступе к результатам. Они сохраняют логику запроса и откладывают фактическое выполнение до тех пор, пока не будет запрошен результат. Примеры отложенных операций включают Where(), OrderBy(), Select() и т. д.

1. Что такое лямбда-выражения?

Лямбда-выражения представляют упрощенную запись анонимных методов.

1. Какие есть группы операции в LINQ to Object? Перечислите

Операции фильтрации и проекции:

* Where: Фильтрация элементов по заданному условию.
* Select: Проекция элементов на новую форму или выбор определенных свойств.
* SelectMany: Проекция и объединение вложенных коллекций.

Операции сортировки и упорядочивания:

* OrderBy: Сортировка элементов по возрастанию.
* OrderByDescending: Сортировка элементов по убыванию.
* ThenBy: Дополнительная сортировка элементов по возрастанию.
* ThenByDescending: Дополнительная сортировка элементов по убыванию.
* Reverse: Изменение порядка элементов на противоположный.

Операции группировки и агрегации:

* GroupBy: Группировка элементов по заданному ключу.
* Aggregate: Выполнение агрегатной операции над элементами коллекции.
* Count, Sum, Average, Min, Max: Вычисление агрегатных функций над элементами коллекции.

Операции соединения:

* Join: Объединение двух коллекций на основе совпадающих ключей.
* GroupJoin: Объединение двух коллекций с созданием групп по ключу.

Операции множества:

* Distinct: Возврат уникальных элементов коллекции.
* Union: Объединение двух коллекций без дубликатов.
* Intersect: Возврат пересечения двух коллекций.
* Except: Возврат разности двух коллекций.

Операции кванторов:

* All: Проверка, что все элементы коллекции удовлетворяют заданному условию.
* Any: Проверка, что хотя бы один элемент коллекции удовлетворяет заданному условию.

Операции преобразования:

* OfType: Фильтрация элементов по заданному типу.
* Cast: Приведение элементов к заданному типу.

1. Как используется операция Where в LINQ to Object?

Where: определяет фильтр выборки

var result = collection.Where(element => condition);

* collection - коллекция или массив, из которой нужно выбрать элементы.
* element - временная переменная, представляющая каждый элемент коллекции по очереди.
* condition - логическое выражение, определяющее условие фильтрации. Если условие истинно для элемента, то он будет включен в результат.

1. Как используется операция Select ?

Select: определяет проекцию выбранных значений

var result = collection.Select(element => expression);

* collection - коллекция или массив, из которой нужно выбрать элементы.
* element - временная переменная, представляющая каждый элемент коллекции по очереди.
* expression - выражение, определяющее, какие данные нужно выбрать или как произвести преобразование над элементами.

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

var result = collection1.Concat(collection2);

1. Как используется операция OrderBy?

OrderBy: упорядочивает элементы по возрастанию

var result = collection.OrderBy(element => keySelector);

10.Как используется операция Join?

Join: соединяет две коллекции по определенному признаку

var result = collection1.Join(collection2, outerKeySelector, innerKeySelector, (outer, inner) => resultSelector);

11.Как используются операции Distinct, Union, Except и Intersect?

**Distinct**: удаляет дублирующиеся элементы из коллекции

**Except**: возвращает разность двух коллекцию, то есть те элементы, которые создаются только в одной коллекции

**Union**: объединяет две однородные коллекции

**Intersect**: возвращает пересечение двух коллекций, то есть те элементы, которые встречаются в обоих коллекциях

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

**First**: выбирает первый элемент коллекции

**Last**: выбирает последний элемент коллекции

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

**Any**: определяет, удовлетворяет хотя бы один элемент коллекции определенному условию

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

13. 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 среднее значение числовых значений в коллекции

**Min**: находит минимальное значение

**Max**: находит максимальное значение

14. Что выведет на экран данный код?

class Test

{

public static void Main()

{

List<int> list = new List<int>();

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

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

foreach (int x in some)

Console.Write(x);

}

}

10