**Вопросы к лабораторной работе №11**

1. **Что такое LINQ?**

**Language Integrated Query** (LINQ) - набор языковых и платформенных средств для написания структурированных и безопасных в отношении типов запросов к локальным коллекциям объектов и удаленным источником данным (базы данных, документы XML и т.д.).

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

Отложенные операции: приложение не создает коллекцию в ходе выполнения метода расширения LINQ — коллекция перечисляется, только когда выполняется ее обход (в foreach). А также выполняются не во время инициализации, а только при их вызове.

Не отложенные операции сразу приводят к получению результата.

1. **Что такое лямбда-выражения?**

**Лямбда-выражение в программировании** — специальный синтаксис для определения функциональных объектов, заимствованный из λ-исчисления. Применяется как правило для объявления анонимных функций по месту их использования, и обычно допускает замыкание на лексический контекст, в котором это выражение использовано.

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

* Агрегация (Count, Min, Max)
* Преобразование (Cast, ofType, ToArray, ToList,ToDictionary)
* Конкатенация (Concat)
* Элемент (Last, First, Single , ElemetAt+ Default)
* Множество (Except, Distinct, Union)
* Генерация (Empty, Range, Repeat)
* Соединение (Join, GroupJoin)
* Упорядочивание (OrderBy, ThenBy, Reverse,….)
* Проекция (Select, SelectMany)
* Разбиение (Skip, Take , +While)
* Ограничение (Where)
* Квантификатор (Any, All, Contains)
* Эквивалентность (SequeceEqual)

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

Фильтрация данных в соответствии с указанным критерием.

1. **Как используется операция Select ?**

Для создания выходной последовательности одного типа из входной последовательности элементов другого типа.

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

* **Take** возвращает указанное количество элементов из входной последовательности, начиная с ее начала.
* **Skip** пропускает указанное количество элементов из входной последовательности, начиная с ее начала, и выводит остальные.

1. **Как используется операция Concat?**

Соединяет две входные последовательности и выдает одну выходную последовательность.

1. **Как используется операция OrderBy?**

Позволяют выстраивать входные последовательности в определенном порядке.

1. **Как используется операция Join?**

Выполняет внутреннее соединение по эквивалентности двух последовательностей на основе ключей.

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

**Distinct** - Удаляет дублированные элементы из входной последовательности.

**Union** - Возвращает объединение множеств из двух исходных последовательностей.

**Except** – Возвращает последовательность, содержащую все элементы первой последовательности, которых нет во второй последовательности.

**Intersected** - Возвращает пересечение множеств из двух исходных последовательностей.

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

* Операция **First** возвращает первый элемент последовательности или первый элемент последовательности, соответствующий предикату.
* Операция **Last** возвращает последний элемент последовательности или последний элемент, соответствующий предикату.
* Операция **Any** возвращает true, если любой из элементов входной последовательности отвечает условию.
* Операция **All** возвращает true, если каждый элемент входной последовательности отвечает условию.
* Операция **Contains** возвращает true, если любой элемент входной последовательности соответствует указанному значению.

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

* Операция **Count** возвращает количество элементов во входной последовательности.
* Операция **Sum** возвращает сумму числовых значений, содержащихся в элементах последовательности.
* Операция **Min Max** возвращает минимальное максимальное значение входной последовательности.
* Операция **Average** возвращает среднее арифметическое числовых значений элементов входной последовательности.

1. **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);

}

Ответ:

