Чем удобен LINQ?

LINQ позволяет запрашивать и преобразовывать строки и коллекции строк. Запросы LINQ можно комбинировать с традиционными строковыми функциями и регулярными выражениями C#. гибче и способен управлять широким диапазоном логических структур данных, может обрабатывать данные с иерархической организацией

Какие источники данных могут быть?

Коллекции, базы данных, файлы

Какие есть формы записи + названия

Изображение выглядит как текст

Автоматически созданное описание

Join

Собирает вместе 2 множества по какому-то ключу

Объединение

Для объединения двух последовательностей используется метод Union. Его результатом является новый набор, в котором имеются элементы, как из первой, так и из второй последовательности. Повторяющиеся элементы добавляются в результат только один раз

1. Что такое LINQ?

**LINQ (Language Integrated Query)** – язык запросов к источнику данных  
Источник данных: объект, реализующий интерфейс IEnumerable (коллекции, массивы), набор д-х DataSet, документ XML

string[] names = {“Анна», «Станислав», «Ольга», «Сева» };

IEnumerable<string> result2 = from n in names //Синтаксис выражения запроса  
 where n.Length < 6  
 select n;

IEnumerable<string> rezult1 = names //Точечная нотация  
 .Where (n => n.Length < 6)  
 .Select (n => n);

Синтаксис выражений запросов поддерживается:  
Where, Select, SelectMany, Join, GroupJoin, GroupBy, OrderBy  
ThenBy, OrderByDescending и ThenByDescending

**System.LINQ**

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

**Отложенные** – не вып., пока не произведен перебор по выборке (вып запроов не в .where… а в foreach)  
**Неотложенные** – возвр. 1 эл-т (Min(), Max(), Count())

3. Что такое лямбда-выражения?

Лямбда-выражения – упрощенная запись анонимных методов (нет return, типа)

4. 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, ElementAt+ Default)  
Мн-во (Except, Distinct, Union)**  
Генерация (Empty, Range, Reapeat)  
Соединение (Join, GroupJoin)  
Упорядочивание (OrderBy, ThenBy, Reverse…)  
Проекция (Select, SelectMany)  
Разбиение (Skip, Take, +While)  
Ограничение (Where)  
Квантификатор (Any, All, Contains)  
Эквивалентность (SequenceEqual)

5. Как используется операция Where в LINQ to Object?

Определяет фильтр выборки

6. Как используется операция Select?

**Select** - передает выбранные значения в рез. выборку, кот. возвращается LINQ-выражением

7. Как используются операции Take, Skip?

Разбиение (+While)  
Выбирает/пропускает опр. кол-во элементов

**Concat** - Объединяет две коллекции  
**OrderBy** - Упорядочивает по возрастанию  
**Join** - Соединяет две коллекции по опр. признаку

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

**Distinct** – удаляет дублирующиеся эл-ты из коллекции  
**Union** – объединяет две однородные коллекции  
**Except** – возвр. разность двух коллекция, то есть те эл-ты, кот. есть только в одной коллекции  
**Intersect** – возвр. пересечение двух коллекций, т.е. эл-ты, кот. встречаются в обоих коллекциях

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

First - для возврата первого элемента массива.  
Last-для возврата последнего  
Any-Метод Any действует подобным образом, только позволяет узнать, соответствует ли хотя бы один элемент коллекции определенному условию:  
All-метод All проверяет, соответствуют ли все элементы условию  
Contains- Определяет, содержится ли указанный элемент в последовательности, используя компаратор проверки на равенство по умолчанию.

13. Как используются операции Count, Sum, Min Max, Average?

**Count** – подсчет эл-тов коллекции, кот. удовл. условию  
**Sum** – подсчет суммы числ. значений в коллекции  
**Min** **Max** – находит мин/макс значение  
**Average** – подсчет ср. значение числ. значений в коллекции