

## Лабораторная работа №9

### Пользовательские LINQ

**Цель работы:** реализовать пользовательские LINQ запросы на языке C#.

#### Теоретическая часть

Помимо LINQ запросов в классе `Enumerable`, можно также описать собственные методы расширения. Для этого нужно следовать правилам:

- В качестве обобщенных типов использовать `TSource`, `TResult`, `TKey`, `TValue`;
- В имени метода нужно перечислить используемые дженерики, например, `GroupBy<TSource, TKey, TElement, TResult>`;
- Метод должен возвращать `IEnumerable<TSource>` или `IEnumerable<TResult>`, если он возвращает коллекцию;
- Первый аргумент `this IEnumerable<TSource> source`;
- Можно передавать функции, например, `Func<TSource, IEnumerable<TResult>> selector`.

Разработаем собственный метод, который будет соединять две коллекции (по аналогии с методом `Concat`).

```
public static class LinqExtensions
{
    0 references
    ... public static IEnumerable<TSource> MyConcatMethod<TSource>(
    ...     ... this IEnumerable<TSource> first,
    ...     ... IEnumerable<TSource> second)
    ... {
    ...     var result = new List<TSource>();
    ...     ... result.AddRange(first);
    ...     ... result.AddRange(second);
    ...     ... return result;
    ... }
}
```

Рисунок 1 – Метод расширения

Этот метод появится в списке доступных методов для коллекции.

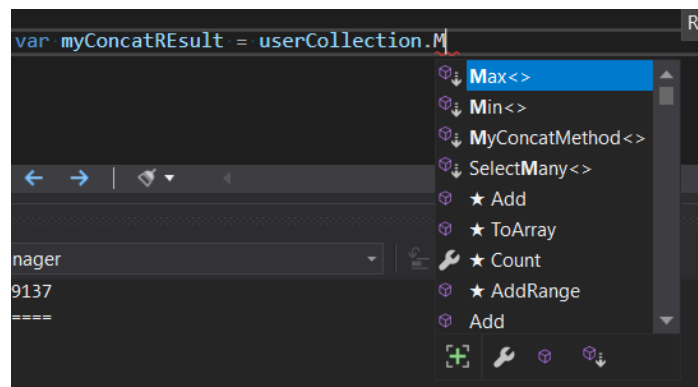


Рисунок 2 – Список доступных методов

Можно использовать его вместе с другими методами LINQ.

```
var atIndexUser = userCollection.ElementAt(19);  
var whereUser = userCollection.Where(x => x.Name.Length > 5).ToList();  
var skipUser = userCollection.Skip(2);  
  
var myConcatResult = userCollection.MyConcatMethod(whereUser).Except(skipUser).ToList();
```

Рисунок 3 – Пример использования

## Практическая часть

Необходимо реализовать 5 собственных методов расширения для коллекций. Использовать разные дженерики. Как минимум в одном методе передать в качестве аргумента Func.

Продемонстрировать работу на примерах коллекций различных типов.

## Содержание отчета

1. Титульный лист
2. Цель работы
3. Задание
4. Код программы
5. Результат выполнения