

Контест 13

11 янв 2022, 18:49:07

старт: 11 июн 2021, 12:30:00

финиш: 20 июн 2021, 23:59:59

длительность: 9д. 11ч.

начало: 11 июн 2021, 12:30:00

конец: 20 июн 2021, 23:59:59

Е. Новые методы

Ограничение времени	2 секунды
Ограничение памяти	64Mb
Ввод	стандартный ввод
Вывод	стандартный вывод

После лекции по программированию, на которой Женя узнал про коллекции в языке C#, он решил попрактиковаться в применении новых знаний на практике и к своему разочарованию обнаружил, что разные коллекции обладают несколько различным набором методов, а некоторые методы, которые было бы удобно использовать и вовсе отсутствуют. К сожалению, Женя пропустил некоторые занятия, поэтому не знает как добавить новые методы к уже существующему классу. Помогите ему расширить функционал коллекций следующими обобщенными методами:

- RemoveWhere, удаляющий из коллекции все элементы, удовлетворяющие переданному предикату;
- AddRange, добавляющий в конец коллекции элементы другой коллекции;
- RemoveDuplicates, удаляющий из коллекции дубликаты;
- AllWhere, возвращающий новую коллекцию, содержащую элементы данной коллекции, удовлетворяющие переданному предикату.

Формат ввода

Первая строка входных данных содержит исходную коллекцию – целые числа, разделенные пробелом. Далее идут две строки, содержащие коллекции, которые необходимо добавить к исходной, также состоящие из целых чисел, разделенных пробелом. В последней строке записано целое число – аргумент для предиката.

Формат вывода

Необходимо вывести результат применения методов к коллекции, каждый результат – с новой строки.

Пример

Ввод	Вывод
1 2 3 4 5	1 2 3 4 5 1 2 3 4 5
1 2 3 4 5	1 3 5 1 3 5
2 4	1 3 5 1 3 5 2 4
2	1 3 5 2 4
	1

Примечания

В систему необходимо сдавать только файл ICollectionExtensions.cs.

Использование Linq в задаче не допускается.

```
1 using System;
2 using System.Collections.Generic;
3
4 public static class ICollectionExtensions
5 {
6     public static void AddRange<T>(this ICollection<int> collection, HashSet<int> hash)
7     {
8         foreach (var el in hash)
9             collection.Add(el);
10    }
11
12    public static void AddRange<T>(this ICollection<int> collection, ICollection<int> collectionAdd)
13    {
14        foreach (var el in collectionAdd)
15            collection.Add(el);
16    }
17
18    public static void RemoveWhere<T>(this ICollection<int> collection, Predicate<int> lambda)
19    {
20        var list = (List<int>)collection;
21        list.RemoveAll(lambda);
22    }
23
24    public static void RemoveDuplicates<T>(this ICollection<T> collection)
25    {
26        //var list = (List<int>)collection;
27        //var listForRemove = new List<int>();
28        //int countRemoves = 0;
29
30        //for (int i = 0; i < list.Count; i++)
31        //    for (int j = i + 1; j < list.Count; j++)
32        //        if (list[i] == list[j])
33        //            listForRemove.Add(j);
34
35        //for (int i = 0; i < listForRemove.Count; i++)
36        //    list.RemoveAt(i - countRemoves++);
37
38    }
```