

Контест 10

11 янв 2022, 18:23:22

старт: 16 мар 2021, 12:30:00

финиш: 22 мар 2021, 23:59:59

длительность: 6д. 11ч.

начало: 16 мар 2021, 12:30:00

конец: 22 мар 2021, 23:59:59

H. MyList

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

Сегодня Егор поспорил с Андреем, что сможет написать класс `List`, как в языке `C#`, только лучше. Ну или хотя бы также. Пока Егор пишет свою реализацию предлагаем Вам последовать его примеру и также разработать такой класс.

Вам необходимо разработать обобщённый класс `MyList<T>` со следующими членами:

- Закрытое поле – ссылка на массив объектов типа, соответствующего типизирующему параметру (для хранения данных).
- Закрытое поле – количество элементов в списке, изначально равно нулю.
- Открытый конструктор без параметров, инициализирующий поле для хранения данных ссылкой на массив нулевой длины.
- Открытый конструктор с одним целочисленным параметром, вместимостью, инициализирующий поле ссылкой на массив заданной длины.
- Открытое свойство `Count` возвращающее количество элементов в списке.
- Открытое свойство `Capacity` возвращающее текущую вместимость списка.
- Метод `Add` добавляющий элемент в конец массива, в случае, если массив заполнен, размер массива удваивается (за исключением случая, когда длина была равна нулю. В этом случае принять длину нового массива равной 4).
- Открытый индексатор, получающий на вход одно целочисленное значение, возвращающий элемент с указанным индексом. Если элемента с таким индексом нет, необходимо выбрасывать исключение `IndexOutOfRangeException` (без указания сообщения).
- Метод `Clear` очищающий список, при этом вместимость списка не изменяется.
- Метод `RemoveLast` удаляющий последний элемент списка, вместимость списка не изменяется. Если такого элемента нет, необходимо выбрасывать исключение `IndexOutOfRangeException` (без указания сообщения).
- Метод `RemoveAt`, принимающий один целочисленный параметр – индекс элемента, который надо удалить. При этом все элементы, следующие за удаляемым, должны быть сдвинуты ближе к началу. В случае, если элемента с таким индексом не существует необходимо выбросить исключение `IndexOutOfRangeException` (без указания сообщения).
- Метод `Max`, возвращающий максимальный элемент коллекции, если это возможно, выбрасывающий исключение `IndexOutOfRangeException` (без указания сообщения), в случае, если коллекция пуста, и выбрасывающий исключение `NotSupportedException` с текстом "This operation is not supported for this type" в случае невозможности сравнения элементов.
- Метод `ToString`, возвращающий записанный через пробел результат вызова `ToString()` для всех элементов коллекции. Пробел после последнего элемента также добавляется.

Работа с входными и выходными данными реализована в коде шаблона и направлена на проверку корректности реализации обобщённого класса `MyList`.

Пример 1

Ввод

Вывод

| Ввод <input type="text"/> | Вывод <input type="text"/> |
|---------------------------|----------------------------|
| string                    | a b                        |
| empty                     | a b d                      |
| add a                     |                            |
| add b                     |                            |
| print                     |                            |
| add c                     |                            |
| remove_last               |                            |
| add d                     |                            |
| print                     |                            |

Пример 2

| Ввод <input type="text"/> | Вывод <input type="text"/> |
|---------------------------|----------------------------|
| string                    | 3                          |
| size 3                    | 0                          |
| capacity                  | 3                          |
| count                     | 3                          |
| add a                     | 6                          |
| add b                     | 4                          |
| add c                     | a b c d                    |
| capacity                  |                            |
| count                     |                            |
| add d                     |                            |
| capacity                  |                            |
| count                     |                            |
| print                     |                            |

Примечания

В систему необходимо сдавать только файл `MyList.cs`.  
В данной задаче запрещено использовать типы из сборки `System.Collections`, то есть, за базовый тип для хранения элементов следует взять массив.

Набрать здесь

Отправить файл

```
1 using System;
2 using System.Diagnostics.CodeAnalysis;
3 using System.Text;
4
5 public class MyList<T>
6 {
7     private T[] array;
8
9     private int countElements;
10
11     public MyList()
12     {
13         array = new T[0];
14     }
15
16     public MyList(int capacity)
17     {
18         array = new T[capacity];
19     }
20
21     public int Count => countElements;
22
23     public int Capacity => array.Length;
24
25
26     public void Add(T element)
27     {
28         if (array.Length == 0)
29             Array.Resize(ref array, 4);
30         else
31             if (countElements == array.Length)
32                 Array.Resize(ref array, array.Length * 2);
33
34         array[countElements++] = element;
35     }
36
37     public T this[int v]
```

[Отправить](#)[Предыдущая](#)[Следующая](#)