

EPAM Systems, RD Dep.

Homework Task

07 – Extended C#

REVISION HISTORY					
Ver.	Description of Change	Author	Date	Approved	
				Name	Effective Date
1.0	Initial version	Alexander_Kuznetsov1	20.01.2012		
2.0		Nikolay Piskarev	29.05.2017		

Задание 1

В кругу стоят N человек, пронумерованных от 1 до N. При ведении счета по кругу вычёркивается каждый второй человек, пока не останется один. Составить программу, моделирующую процесс.

Задание 2

Задан английский текст. Выделить отдельные слова и для каждого посчитать частоту встречаемости. Слова, отличающиеся регистром, считать одинаковыми. В качестве разделителей считать пробел и точку.

Задание 3

На базе обычного массива (коллекции .NET не использовать) реализовать свой собственный класс `DynamicArray`, представляющий собой массив с запасом. Класс должен содержать:

1. Конструктор без параметров (создается массив емкостью 8 элементов).
2. Конструктор с 1 целочисленным параметром (создается массив заданной емкости).
3. Конструктор, который в качестве параметра принимает коллекцию, реализующую интерфейс `IEnumerable`, создает массив нужного размера и копирует в него все элементы из коллекции.
4. Метод `Add`, добавляющий в конец массива один элемент. При нехватке места для добавления элемента емкость массива должна расширяться в 2 раза.
5. Метод `AddRange`, добавляющий в конец массива содержимое коллекции, реализующей интерфейс `IEnumerable`. Обратите внимание, метод должен корректно учитывать число элементов в коллекции с тем, чтобы при необходимости расширения массива делать это только один раз вне зависимости от числа элементов в добавляемой коллекции.
6. Метод `Remove`, удаляющий из коллекции указанный элемент. Метод должен возвращать `true`, если удаление прошло успешно и `false` в противном случае. При удалении элементов реальная емкость массива не должна уменьшаться.

7. Метод Insert, позволяющий добавить элемент в произвольную позицию массива (обратите внимание, может потребоваться расширить массив). Метод должен возвращать true, если добавление прошло успешно и false в противном случае. При выходе за границу массива должно генерироваться исключение ArgumentOutOfRangeException.
8. Свойство Length – получение длины массива.
9. Свойство Capacity – получение реальной длины массива.
10. Методы, реализующие интерфейсы IEnumerable и IEnumerator.
11. Индексатор, позволяющий работать с элементом с указанным номером. При выходе за границу массива должно генерироваться исключение ArgumentOutOfRangeException.

Требования к оформлению

1. Для каждой домашней работы необходимо создать собственную папку в корне репозитория.
2. Каждое задание должно представлять собой отдельный проект (project) в рамках общего решения (solution).
3. Правила именования:
Все названия и имена должны быть переведены на английский язык. Именованье транслитерацией не допускается.
 - a. Маска именования репозитория: EPAM_Ext_Lab_<Квартал старта группы>_<Год>_<Фамилия_Имя>.
Например, EPAM_Ext_Lab_Q2_2017_Ivanov_Ivan
 - b. Маска именования решения: HMT_<Порядковый номер>.
Например, HMT_01, HWT_11.
 - c. Маска именования проектов: <Задача><Номер задачи>.
Например, Task01, Task12.
 - d. Каждое задание должно быть оформлено следующим образом:
 1. Формулировка задания в виде комментария (просто копировать-вставить из этого файла).
 2. Код, решающий задачу.
4. Написать письмо преподавателю, информирующее о том, что домашняя работа выполнена. Тема письма должна выглядеть так: [STYYYY_N]Фамилия_Имя_TaskNN. Здесь ST – сокращённое наименование программы (Students Training), YYYY – номер года, N – номер группы, NN – порядковый номер задания в виде двух цифр, напр. 01, 05, 12 и т.д.
Например, ST2017_1 Иванов Иван Task01.

Рекомендации к оформлению кода

1. При добавлении нового файла желательно вверху файла добавить блок с комментарием, для чего предназначен этот файл.
2. При добавлении метода в блоке `<summary></summary>` описать предназначение метода, его входных/выходных параметров и возвращаемых значений.
3. Использовать утилиту StyleCop для проверки правил оформления кода. Перед отправкой задания на проверку запускать StyleCop и исправлять все предупреждения, возникающие в секции Warning Visual Studio. Предупреждения об отсутствии документации можно игнорировать (либо настроить StyleCop таким образом, чтобы он не проверял документирование кода).