



Task 03

COLLECTIONS

EPM-FARM

Legal Notice: This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM®.

CONFIDENTIAL | Effective Date: 2015-09-28

3.1. LOST

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.

3.2. WORD FREQUENCY

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

3.3. DYNAMIC ARRAY

На базе обычного массива (коллекции .NET не использовать) реализовать свой собственный класс `DynamicArray<T>`, представляющий собой динамический массив (массив с запасом) для хранения объектов произвольных типов. Класс должен содержать:

1. Конструктор без параметров (создаётся массив ёмкостью 8 элементов).
2. Конструктор с одним целочисленным параметром (создаётся массив указанной ёмкости).
3. Конструктор, который в качестве параметра принимает коллекцию, реализующую интерфейс `IEnumerable<T>`, создаёт 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, ёмкость массива должна удваиваться.
5. Метод `AddRange`, добавляющий в конец массива содержимое коллекции, реализующей интерфейс `IEnumerable<T>`. 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 ёмкость 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 количества элементов. Не путать с ёмкостью (`Capacity`).
9. Свойство `Capacity` — получение ёмкости: длины внутреннего массива.
10. Методы, реализующие интерфейсы `IEnumerable` и `IEnumerable<T>`.
11. Индексатор, позволяющий работать с элементом с указанным номером. При выходе за границу массива должно генерироваться исключение `ArgumentOutOfRangeException`.

3.4. * DYNAMIC ARRAY (HARDCORE MODE)

Дополнить динамический массив из задания 3.3 следующим функционалом:

1. Доступ к элементам с конца при использовании отрицательного индекса (-1: последний, -2: предпоследний и т.д.).
2. Возможность ручного изменения значения `Capacity` с сохранением уцелевших данных (данные за пределами новой `Capacity` сохранять не нужно).
3. Реализовать интерфейс `ICloneable` для создания копии массива.
4. Добавить метод `ToArray`, возвращающий новый массив (обычный), содержащий все содержащиеся в текущем динамическом массиве объекты.

5. Создать новый класс: циклический динамический массив (CycledDynamicArray) на основе DynamicArray, отличающийся тем, что при использовании foreach после последнего элемента должен снова идти первый и так по кругу.