

|  |
| --- |
| Task 3.2  DYNAMIC ARRAY |
| EPAM XT COMMUNITY |

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**



## Dynamic ARRAY

***\* включает дополнительное задание***

***\*\* включает углубленное дополнительное задание***

На базе массива (именно массива, специфичные коллекции .NET не использовать) реализовать свой собственный класс DynamicArray<T>, представляющий собой массив с запасом, хранящий объекты 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.

В этот раз можете обойтись без консольного интерфейса. Но коллекция должна быть работоспособна, что и будут проверять ментора.

***Вариант сo \**** - дополните динамический массив 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.

***Вариант с \*\**** - снова попробуйте создать из своей сборки переносимую библиотеку (DLL). Попробуйте использовать написанные вами классы в другом проекте.