# Задания

## Задание 1

1. Создать класс User со следующими свойствами:
2. Имя - содержит русские или латинские символы (в одном имени не могут присутствовать и те и другие), начинается с заглавной буквы. Максимальное число символов 50. Обязательное поле.
3. Фамилия - содержит русские или латинские символы (в одной фамилии не могут присутствовать и те и другие), начинается с заглавной буквы. Максимальное число символов 200. Может содержать дефис, в этом случае первый символ после дефиса заглавная. Дефис не может быть первым или последним символом. Обязательное поле.
4. Электронная почта - имя почтового ящика может содержать буквы, цифры, точку, дефис и знак подчеркивания, причем первым и последним символами могут быть только буквы или цифры. Для имен поддоменов действуют те же самые правила, но точка допустимой не является. Имя домена первого уровня может состоять только из букв в количестве от 2 до 6 (те же правила что в Лабораторной 3 в задании 5). Необязательное поле.
5. Дата рождения – в формате DateTime. Необязательное поле.
6. Дата регистрации - в формате DateTime. Обязательное поле.
7. Логин - содержит латинские символы. Максимальное число символов 20. Обязательное поле.
8. Переопределить метод ToString() для вывода строки в формате

{Логин}, {Имя}, {Фамилия}, {Электронная почта}, {Дата рождения}

Дата рождения должна выводиться в формате dd-mm-yyyy.

1. Создать метод заполнения объекта из строки (использовать тот же формат, что и в методе ToString())
2. В случае ошибок возвращать собственные исключения.

# Задание 2

На базе обычного массива (коллекции .NET не использовать) реализовать свой собственный класс DynamicArray, представляющий собой Generic 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 все вхождения указанного элемента. При удалении элементов реальная емкость массива не должна уменьшаться. Может быть передан делегат для сравнения 2 элементов. Если делегат не передан (по умолчанию), идет прямое сравнение элементов.
7. Метод Insert, позволяющий добавить элемент в произвольную позицию массива (обратите внимание, может потребоваться расширить массив). 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.
12. Переопределить метод Equals и оператор сравнения массива. Массивы одинаковы, если содержат одни и те же элементы в том же самом порядке.
13. Переопределить операторы явного преобразования DynamicArray к массиву.
14. Переопределить операторы неявного преобразования массива к DynamicArray.
15. Реализовать интерфейс IDisposable для очистки ресурсов (установка размера массива в 0). Если элементы массива поддерживают интерфейс IDisposable, то вызвать у них метод Dispose
16. Добавить к классу DynamicArray событие, которое возникает при изменении Capacity. В аргументах события должны содержаться старое значение и новое.
17. Написать программу, демонстрирующую возможности класса.