**Задание 1.**

Напишите библиотеку классов **Computer**

Реализуйте класс **PC,** содержащий:

* Список Драйверов необходимых для работы
* Свойство выдающее количество необходимый драйверов
* Вещественное поле стоимость компьютера
* Вещественное поле, лежащее в пределах от 0 до 10 и показывающее насколько мощный данный компьютер
* Свойство, выдающее общую стоимость всех драйверов.
* Конструктор от 3-ех параметров.
* Переопределите **ToString(),** чтобы он выдавал всю информацию о компьютере и список необходимых драйверов.

Реализуйте класс **Driver**, содержащий:

* Вещественное поле стоимость драйвера
* Целочисленное поле Id драйвера (у каждого уникальный и неотрицательный)
* Конструктор от 2-ух параметров.
* Переопределите **ToString()**

В первой программе создайте экземпляр класса **PC**, создав список случайной длины драйверов (все поля заполнять случайно одним рандомом). С помощью xml сериализуйте объект класса **PC.**

Во второй программе десериализуйте объект из файла и выведете на консоль таким образом **Console.WriteLine(obj)**.

**Задание 2.**

Измените библиотеку классов так, чтобы для прошлого задания она также работала, но при этом реализуйте бинарную сериализацию.

**Задание 3**

Создайте классы **Human**, **Professor**, **Department, University**.

Класс **Human** содержит строковое поле Name.

Класс **Professor**, наследник **Human**, содержит целочисленное поле Experience – стаж работы преподавателем.

Класс **Department** содержит строковое поле DepartmentName. Также включает в себя список преподавателей департамента – объект *List<Human>*

Классы **Human** и **Department** должны состоять в отношении агрегации.

Класс **University** содержит строковое поле UniversityName. Также включает в себя список департаментов – объект *List<Department>*

Классы **University** и **Department** должны состоять в отношении композиции.

Для всех классов реализовать конструкторы, в конструкторе класса **Professor** обращаться к конструктору базового класса.

Создать объект класса **University**, заполнить случайными значениями необходимые поля и выполнить XML сериализацию.