1.  В классе **Task1** создайте структуру **Rectangle**, представляющую собой прямоугольник с целочисленными полями для длины и ширины прямоугольника и свойствами для чтения его сторон **A, B**. В конструктор прямоугольника передавать 2 целых числа. Напишите методы **Length()** и **Area()** для расчета периметра и площади прямоугольника. Напишите метод **Compare** для сравнения двух прямоугольников: какой из них больше по площади. На вход передается второй прямоугольник для сравнения, на выход: целое число (1 - больше текущий, -1 - переданный прямоугольник, 0 - прямоугольники равны). Переопределите метод **ToString()** для вывода информации о прямоугольнике в формате:  “a = 1, b = 2, p = 3, s = 4”, где a,b - длины сторон, p - периметр, s - площадь прямоугольника. В классе **Task1** создайте приватное поле для массива прямоугольников и свойство **Rectangles** для чтения этого массива. Переопределите метод **ToString()** для вывода всех элементов массива **Rectangles** на консоль построчно. Напишите метод **Sorting()** для сортировки массива **Rectangles** по возрастанию периметра прямоугольников (чем выше скорость сортировки, тем больше баллов за выполнение).

2. В класс **Task2** скопируйте код из класса **Task1**. Создайте в нём абстрактный класс **Embrasure**, наследующийся от **Rectangle** (тоже сделать абстрактным), который представляет собой проем с названием, длиной, шириной, толщиной, свойством для чтения толщины **Thick** и методом для расчета стоимости установки **Cost(),** как произведение сторон \* 10, который возвращает вещественное число. Переопределите метод **ToString()** для вывода проемов на консоль  в формате:  “Type = 1, p = 2, s = 3, price = 4”, где 1 - тип класса, 2 - периметр, 3 - площадь, 4 - цена. От **Embrasure** сделайте наследников: **Window** и **Door**. В классе окна добавьте поле количество слоев и свойство для их чтения **Layers**. В классе двери добавьте поле материал и для его чтения (пластик/дерево/стекло) **Material**. Переопределите метод **Cost()** с учетом дополнительных полей классов-наследников: + количество слоев / 0.3 и для материалов множители 1.25, 1.33, 1.5. Толщину передавать в конструктор класса **Embrasure,** в конструктор класса **Window** передавать дополнительно количество слоев, в конструктор класса **Door -** список названий материалов. В классе **Task2** создайте массив проемов и свойство **Embrasures** для его чтения. Напишите метод **Sorting()** для сортировки проемов по увеличению стоимости. Переопределите метод **ToString()** для вывода проемов на консоль.

3. В классе **Task3** создайте класс **Statistic** с полями для входной строки и выходного дробного числа. Сделайте публичные свойства для чтения этих полей **Input** и **Output**. В конструктор должен передаваться текст и сохраняться как входная строка.  После этого посчитать среднее арифметическое количество слогов во всех словах текста (словом считать часть строки, содержащую в себе хотя бы одну букву) и сохранить в выходное значение. Переопределите метод **ToString(),** чтобы он возвращал выходное значение построчно или пустую строку, если входной текст был неверного формата.

4. В классе **Task4** создайте интерфейс **ISerializer** с методами **Serialize** и **Deserialize** для записи объекта и чтения объекта по переданному пути. Создайте класс **FolderManager** с 4 методами для создания папок и файлов: **CreateFolder** и **CreateFile**. 2 метода получают путь, где создать папку/файл и название папки/файла. 2 других метода получают путь, где создать папки/файлы и массив названий папок/файлов. От него создать класс-наследник **StatisticSerializer** и в нем реализовать интерфейс **ISerializer** для работы с классом **Statistic** (или любым типом) для хранения экземпляра **Statistic** в формате JSON. Создайте приватное поле для хранения экземпляра Statistic и свойство для чтения этого экземпляра.