**Задание №5**

**Модификатор типа static в классах C#**

В предыдущем задании необходимо было реализовать классы с отношением «клиент - поставщик». В результате каждый класс должен был содержать член переменную типа Point.

1. Модифицировать программу, которая была создана в задании №4, следующим образом:

* Добавить в каждый «класс-клиент» член переменную типа Point, которая описывает вершины геометрической фигуры (если это еще не реализовано).
* Ввести статическую переменную count – закрытый член класса Point, которая является счетчиком числа созданных объектов типа Point.
* Создать деструктор для класса Point.
* Конструктор инкрементирует count, деструктор декрементирует count.
* Реализовать интерфейс для счетчика count – член метод, который объявлен с модификатором static, и который возвращает текущее количество существующих объектов в момент вызова данного метода.
* В теле основной программы сообщить о количестве существующих объектов типа Point в разные моменты исполнения кода основной программы.

2. Создать отдельную программу, в которой реализован класс, содержащий static-члены.

* Класс описывает объект «Вклад», который имеет фиксированную ставку, название банка, дату открытия вклада, срок вклада, ФИО держателя вклада, сумму вклада.
* Частичное пополнение и снятие средств не осуществляется.
* Ставка, название банка, срок вклада хранятся в статическом члене переменной, с модификатором доступа private.
* В классе должен быть статический конструктор, который инициализирует все члены переменные с модификатором static.
* Также, в классе должен быть конструктор, который инициализирует члены переменные, которые объявлены без модификатора static.
* Создать интерфейсные методы: для вывода полной информации о вкладе, для вывода суммы вклада по истечению заданного количества дней.
* В основной программе инициализировать объект класса «Вклад» и продемонстрировать его полезную работу.

3. Создать отдельную программу, в который реализован статический класс, содержащий только static-члены. Класс содержит коллекцию методов для выполнения математических функций. Продемонстрировать работу класса на примерах.

* Вычисление факториала аргумента *Fact()*
* Вычисление обратного числового значения аргумента *Reciprocal()*
* Возврат дробной части числового аргумента *FracPart()*
* Возврат флага четности аргумента *IsEven()*
* Возврат флага нечетности аргумента *IsOdd()*
* Вычисление кубического корня аргумента *Crt()*
* Вычисление радиан по аргументу в градусах *DegToRad()*
* Вычисление градусов по аргументу в радианах *RadToDeg()*
* Проверка числа на принадлежность ряду чисел-степеней двойки *BinaryDigit()*