

Задание 4.1. Очередь (ссылочные типы)

Реализуйте очередь в виде списка, содержащую комплексные числа

Реализуйте методы

- void Enqueue(Complex c) – помещает комплексное число в конец очереди
- Complex Dequeue() – получает комплексное число из начала очереди и удаляет его из очереди
- Complex Peek() – возвращает комплексное число, находящееся в начале очереди (не удаляет его)
- Свойство int Count доступное только для чтения, возвращающее кол-во элементов в очереди
- void Print() - метод, распечатывающий содержимое очереди.

где Complex – класс комплексных чисел, со свойствами Re и Im и переопределённым методом ToString()

Примечание:

- Не разрешается использовать классы из пространства имен System.Collections и его производных

Задание по желанию.

Задача 4.2. Интерфейсы (явная и неявная реализации интерфейсов, особенности реализаций)

Реализуйте абстрактный класс Shape, содержащий метод Draw(), якобы рисующий фигуру (вывод текста “Shape” на экран)

- Создайте классы его потомков
 - Triangle,
 - Hexagon,
 - Circle,
 - Sphere
- Создайте интерфейсы
 - IPoint, со свойством Point, выдающим количество точек в фигуре.
 - IDrawable, с методом Draw(), якобы рисующем фигуру (вывод строки на экран)
- Реализуйте
 - IPoint для Triangle и Hexagon
 - IDrawable для Triangle (не явно) и Sphere (явно)
 -
- В основном классе:
 - Создайте метод AnalyzeShape(), принимающий Shape и распечатывающий, если возможно, кол-во точек в фигуре, и вызывающий метод Draw напрямую и через интерфейс.
 - В методе Main создайте массив из Shape с разными фигурами и проанализируйте их вызвав метод AnalyzeShape()

Примечание:

- Обратите внимание, что у класса Sphere будет 2 метода Draw()