

## Семинар 3.

### TODO задачи.

№1. Написать класс **Complex** для работы с комплексными числами. Напишите метод (публичный) для получения модуля комплексного числа. Переопределите: инкременты “-” и “++”, **true** (если модуль комплексного числа больше 1), **false** (если модуль  $\leq 1$ ).

**ToDo:** Дополнить код класса **Complex** перегрузками бинарных операций:

1. **+** для сложения комплексных чисел
2. **\*** для умножения комплексных чисел
3. **-** для вычисления разности комплексных чисел
4. **/** для получения частного от деления комплексных чисел

Создайте два комплексных числа и продемонстрируйте работу перегруженных операций.

**сложение**  $(a + bi) + (c + di) = (a + c) + (b + d)i$

**вычитание**  $(a + bi) - (c + di) = (a - c) + (b - d)i$

**умножение**  $(a + bi) * (c + di) = (a * c - b * d) + (b * c + a * d)i$

**деление**  $\frac{(a + bi)}{(c + di)} = \frac{a * c + b * d}{c^2 + d^2} + \left( \frac{b * c - a * d}{c^2 + d^2} \right) i$

№2. (\*) Класс **RectSet** описывает множество целых точек, принадлежащих отрезку прямой. Поля класса:

1. Границы множества – максимальная и минимальная точки.
2. Элементы множества, представленные множеством **HashSet <int>**.

**ToDo:**

- В строки, помеченные метками **ToDo**, поместите программный код, реализующий следующие действия:
  1. Установить левую и правую границы множества
  2. Перегрузить операцию **\*** для формирования пересечения (**intersect**) множеств. В пересечение множеств **A** и **B** попадают, элементы входящие как в **A**, так и в **B**.
- 3. Перегрузить операцию **^** для получения множества, состоящего из различающихся элементов. В такое множество для **A** и **B** попадают все элементы множеств, исключая элементы пересечения.
- В код свойств, методов и конструкторов добавьте проверки корректности данных. В случае неподходящих значений, предусмотрите генерацию исключений.
- В код класса могут быть добавлены дополнительные свойства и методы.

### Дополнительные материалы для задачи 2

- В задаче 2 используются объекты, представляющие множества **HashSet**. Код в проекте (**Task\_02\_Exp**) содержит несколько примеров работы с элементами **HashSet**;
- Класс **HashSet** описан в пространстве имён **System.Collection.Generic**;

<https://docs.microsoft.com/ru-ru/dotnet/api/system.collections.generic.hashset-1?redirectedfrom=MSDN&view=netframework-4.8>

- Для множеств **HashSet** определена возможность объединения и пересечения, соответствующие методы реализуют методы интерфейса **System.Linq.IEnumerable**.

**Distinct** - выбрать из последовательности различающиеся элементы

**Union** – объединяет две последовательности с учетом **EqualityComparer <T>**

**Intersect** – пересечение двух последовательностей (сравнение на равенство)

**Except** – формирует разность двух последовательностей

Операции HashSet	Математический эквивалент
<a href="#">UnionWith</a>	Объединение или сложение множеств
<a href="#">IntersectWith</a>	Пересечение
<a href="#">ExceptWith</a>	Вычитание набора
<a href="#">SymmetricExceptWith</a>	Симметричное различие между