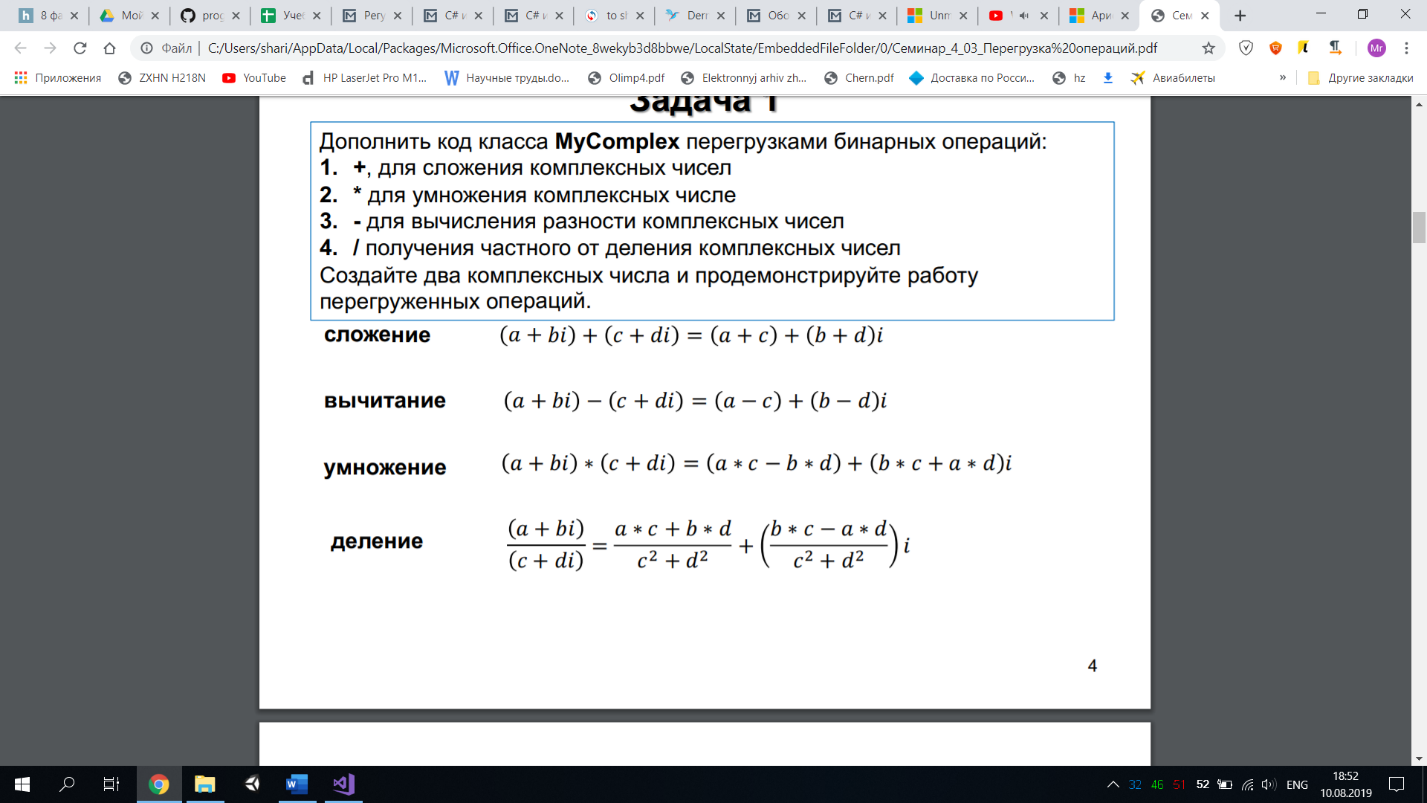
Семинар 3.

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 ≤ 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.

№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 *А* и *В* попадают, элементы входящие как в *А*, так и в *B*.

3. Перегрузить операцию ^ для получения множества, состоящего из различающихся элементов. В такое множество для *А* и *В* 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**;

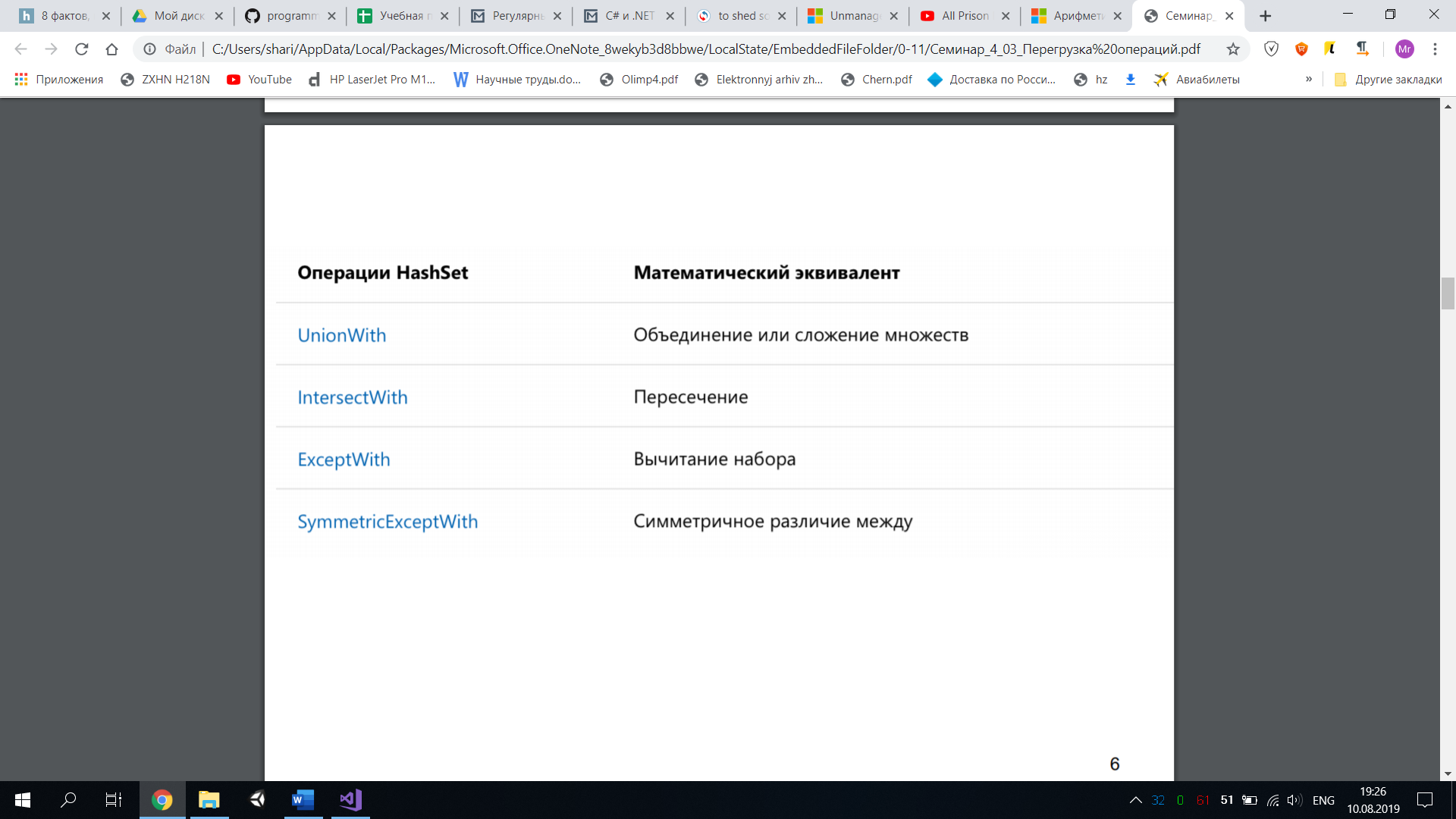
<https://docs.microsoft.com/ru-ru/dotnet/api/system.collections.generic.hashset-1?redirectedfrom=MSDN&view=netframework-4.8>

* 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