**Міністерство освіти і науки України**

**Харківський національний університет ім. В. Н. Каразіна**

**Факультет комп’ютерних наук**

**Кафедра безпеки iнформацiйних систем i технологiй**

**Лабораторна робота № 3**

*з навчальної дисципліни*

«**Математичні методи та технології тестування і верифікації програмного забезпечення**»

***Виконала:***

студентка групи КБ-22

Рогоза П. В.

***Перевірила:***

Мелкозьорова О. М.

Харків – 2019 р.

**Тема:** Изучения модульного тестирования (unit testing) на примере фреймверка JUnit.

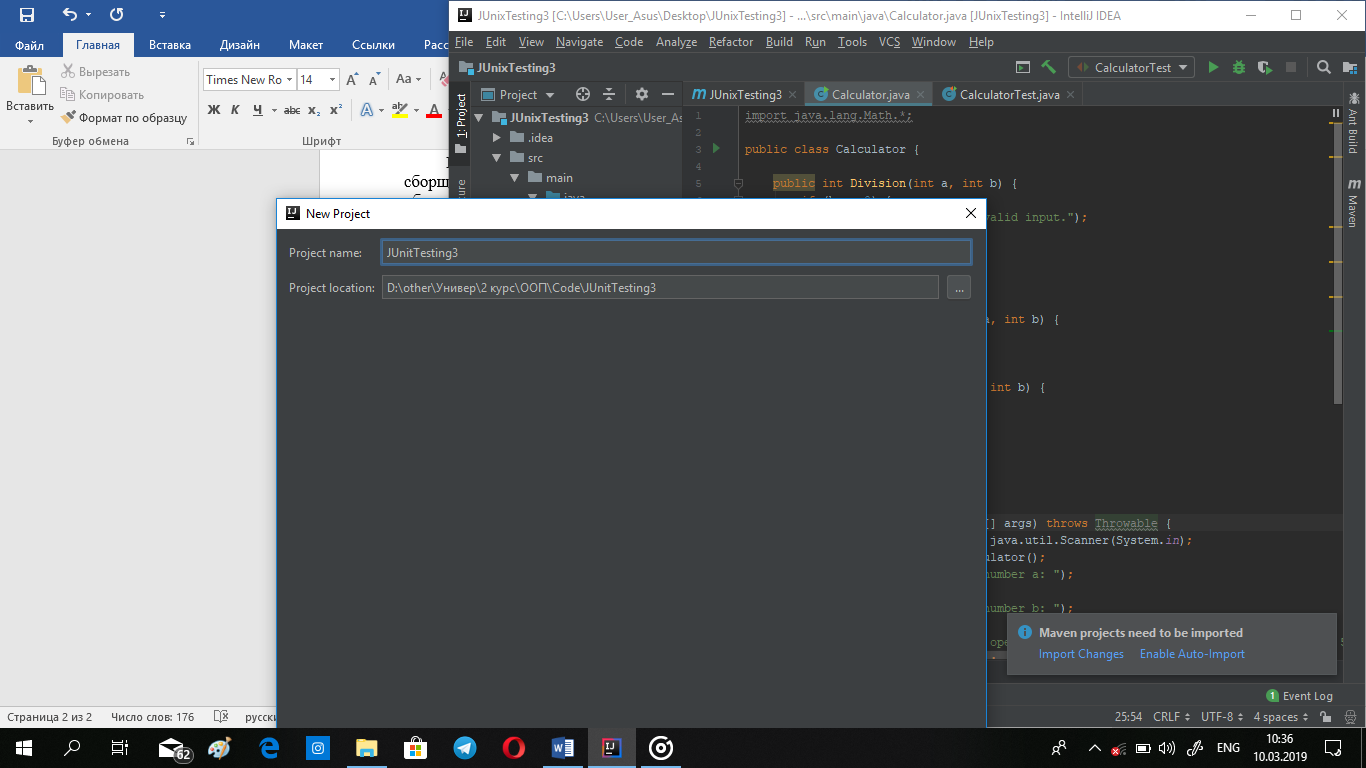
**Цель работы:** Изучение фреймверка для модульного тестировани JUnit, сборщиков проектов, написание автоматизированных тестов для программного обеспечения.

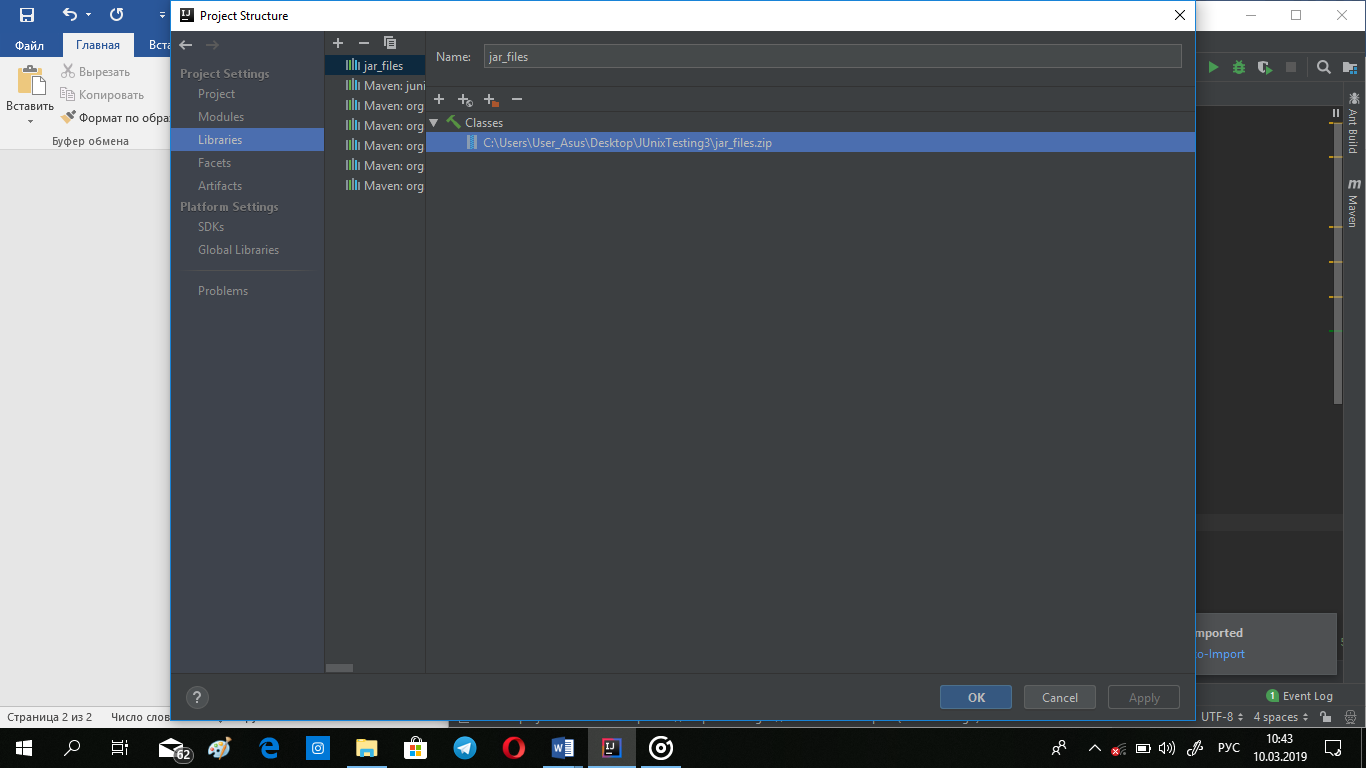
***ЗАДАНИЕ***

1. ***Установить IntelliJ IDEA (если установлено, то пропустить).***

На ПК уже установлена программа IntelliJ IDEA Community Edition 2018.3.4 x64а. Запускаем программу.

***2. Создать проект Java, подключь JUnit к проекту.***

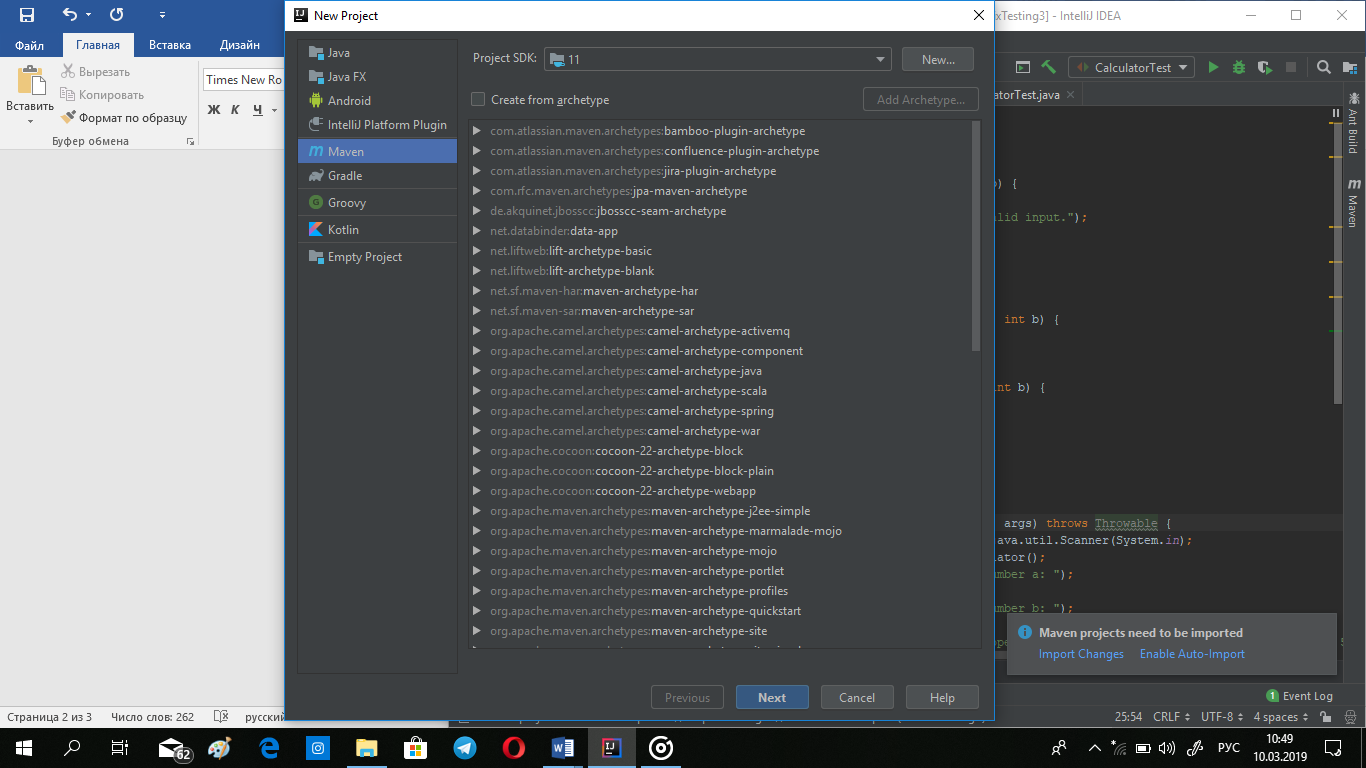
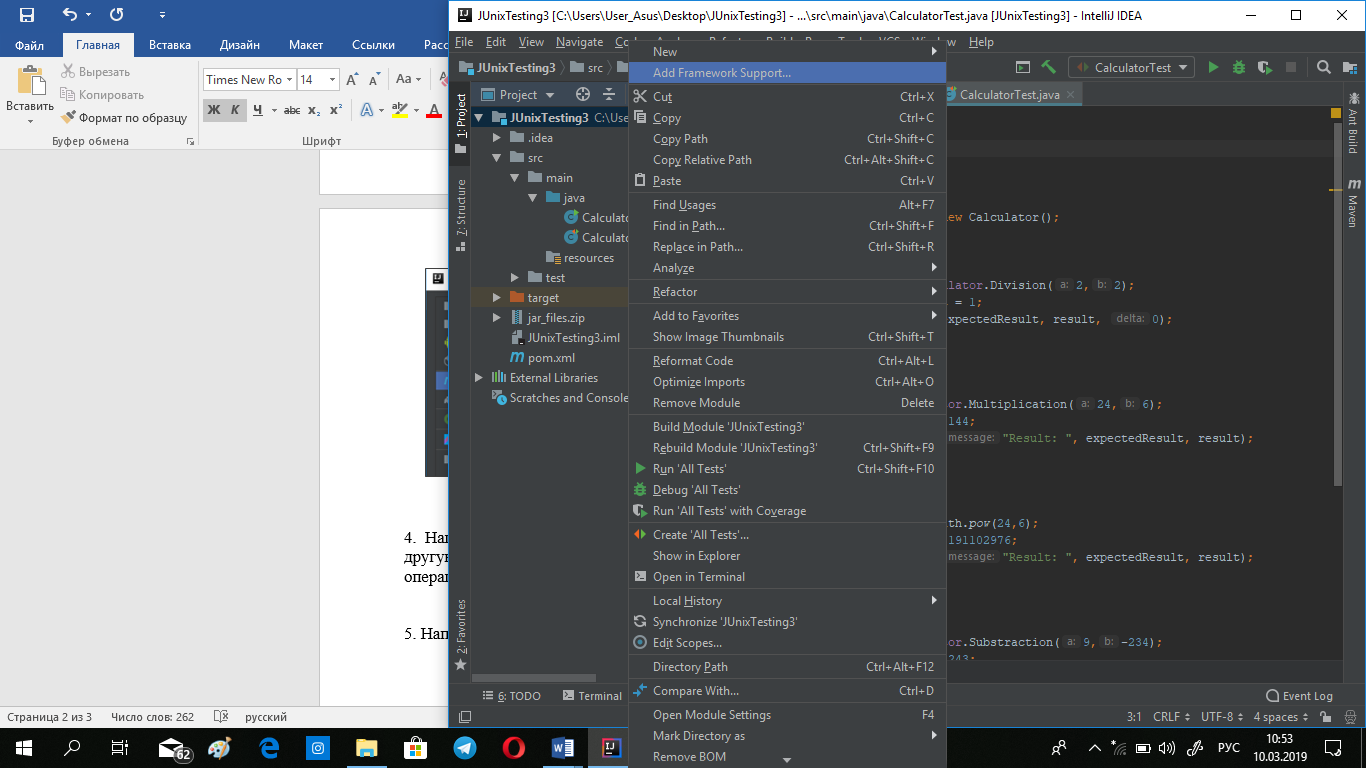
Чтобы создать новый проект, в левом верхнем углу экрана переходим по команде **File 🡪 New 🡪 Progect…**. В открывшемся окне два раза нажать **Next** и вписать имя проекта, нажать **Finish.**

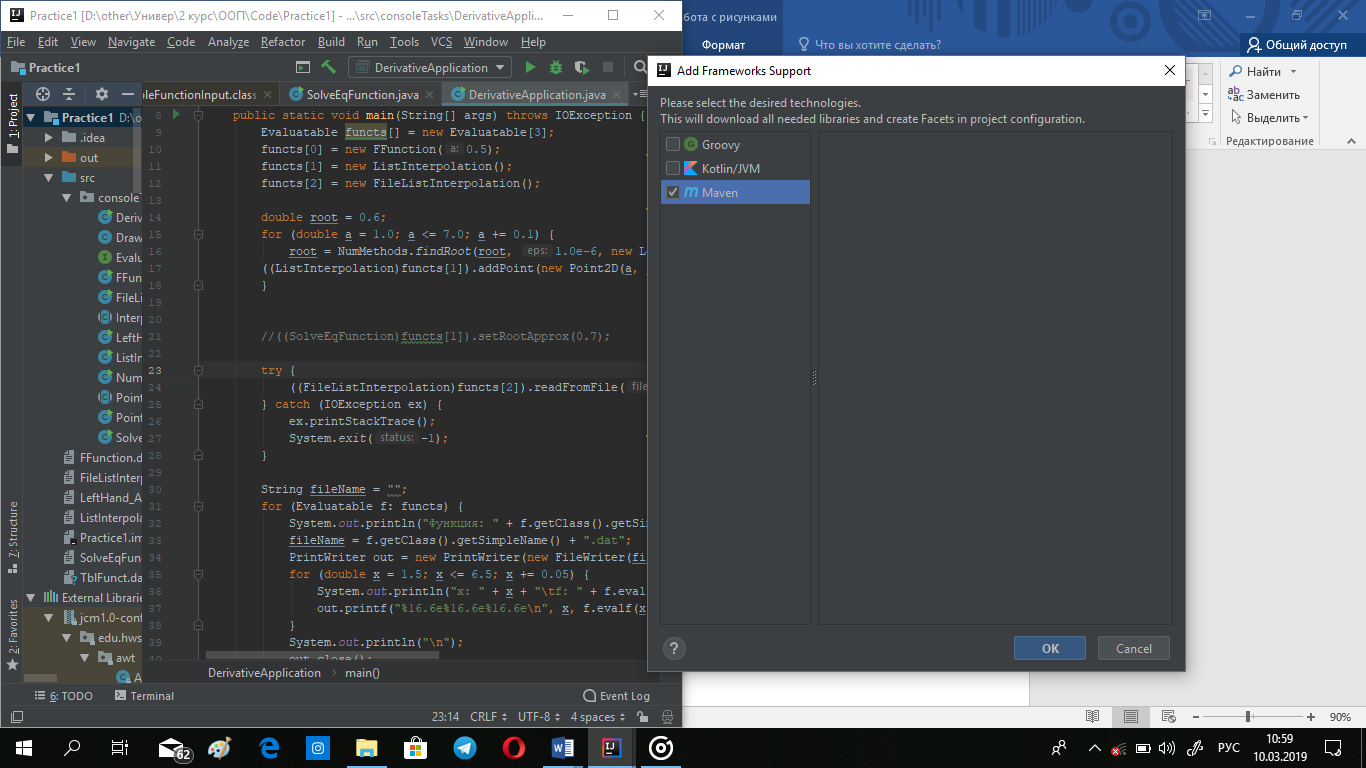
Для подключения JUnit, необходимо скачать zip-файл (библиотеку) и добавить ее в проект с помощью команды: **File 🡪 Progect Structure…**, затем во всплывающем окре перейти во вкладку **Libraries**, нажать на символ «плюс» (или воспользоваться комбинацией **Alt + Insert**), выбрать архив, подтвердить. Таким образом, мы подключили JUnit к Java-проекту.

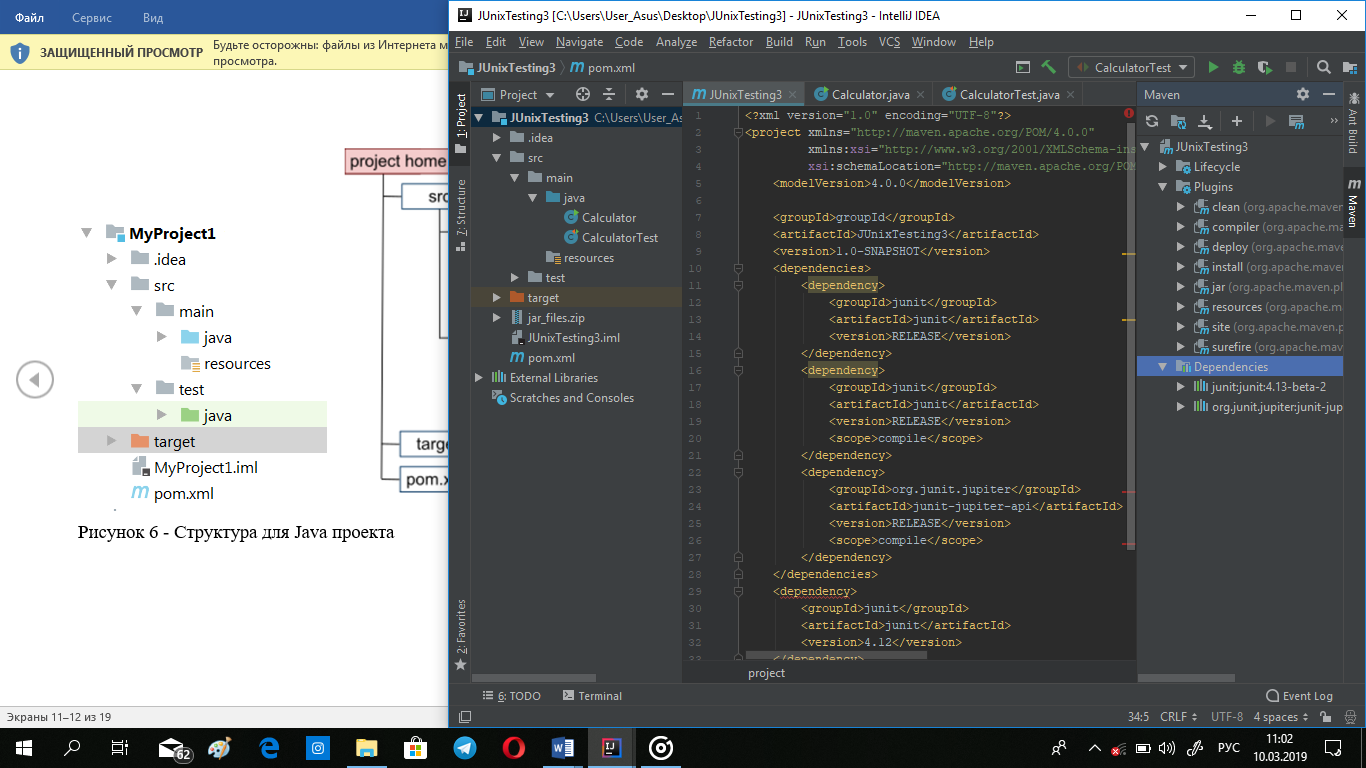
***3. Создать проект при помощи сборщика проектов Maven, подключить JUnit.***

Чтобы создать новый **Maven**-проект, в левом верхнем углу экрана переходим по команде **File 🡪 New 🡪 Progect…**. В открывшемся окне, в левой части, сразу

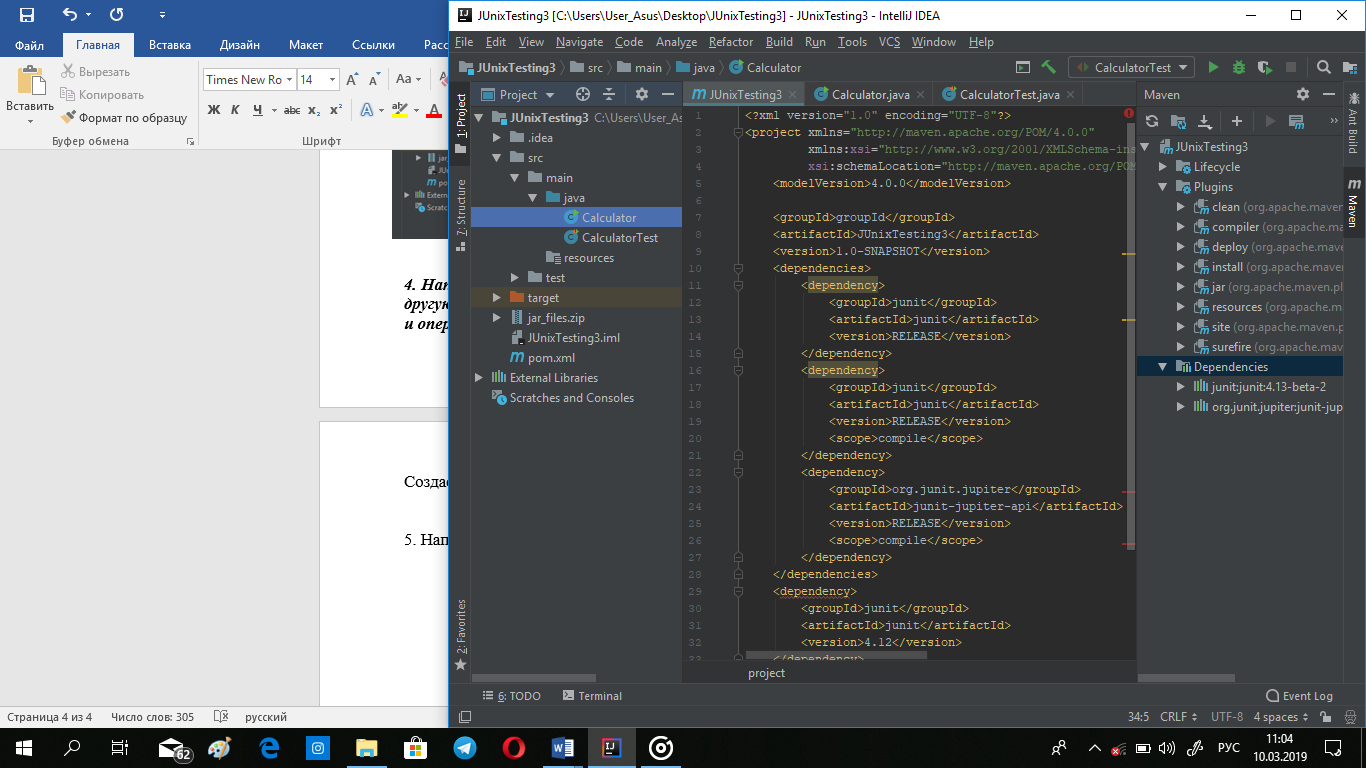
выбираем категорию **Maven,** в основном окне отобразятся архетипы. Дальнейшее создание проекта аналогично з.2.

Также к уже созданному проекту можно подключить фреймверк **Maven**, правой кнопкой мыши нажав на корневую папку проекта и выбрав **Add Framework Support**… В окне выбрать **Maven** и подтвердить.

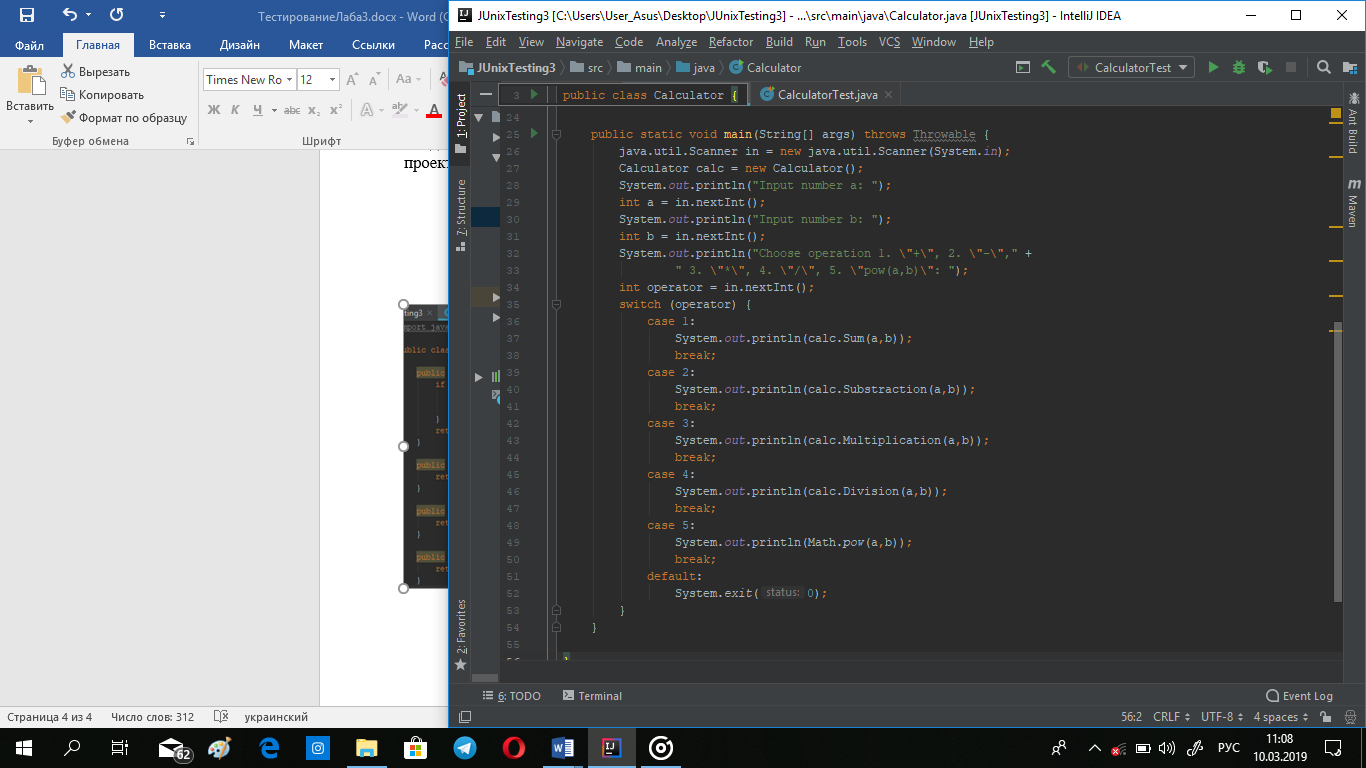
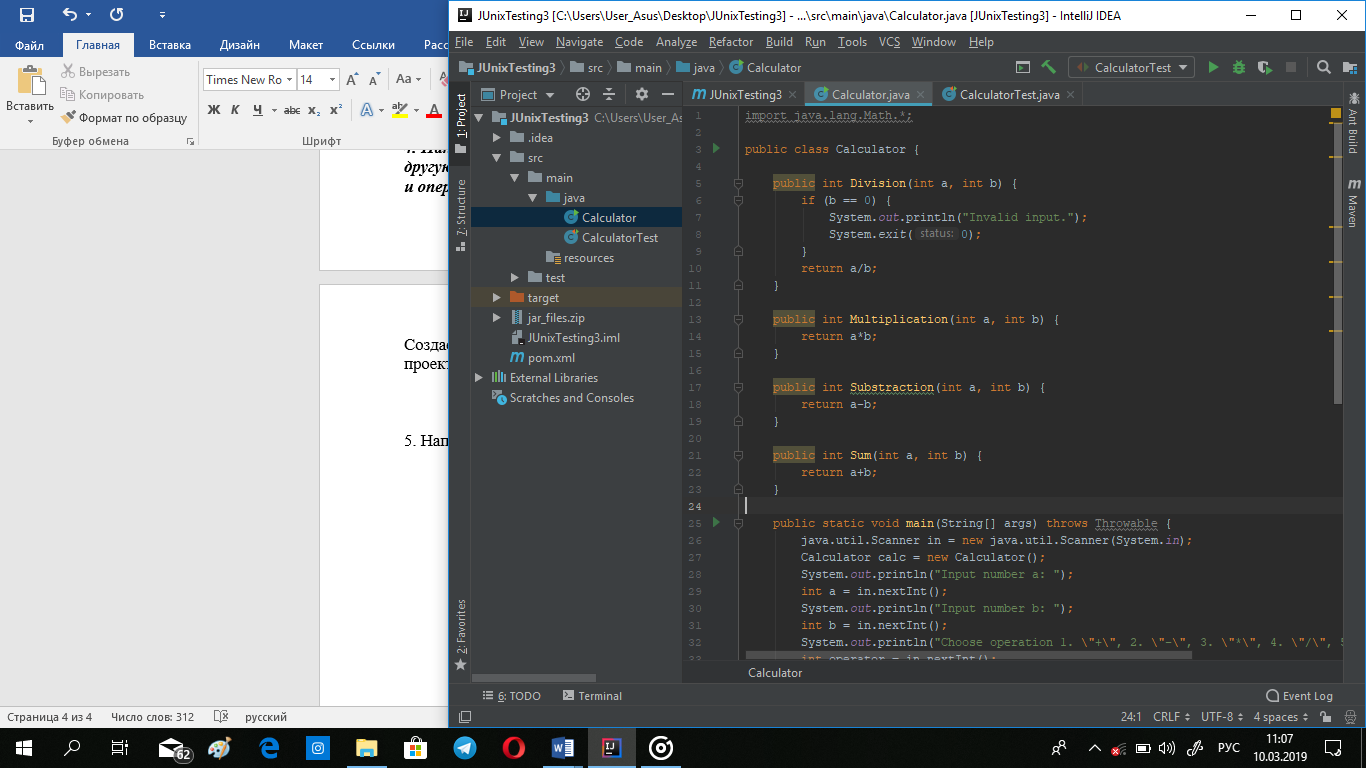


Так будет выглядеть проект после всех ранее описанных действий:

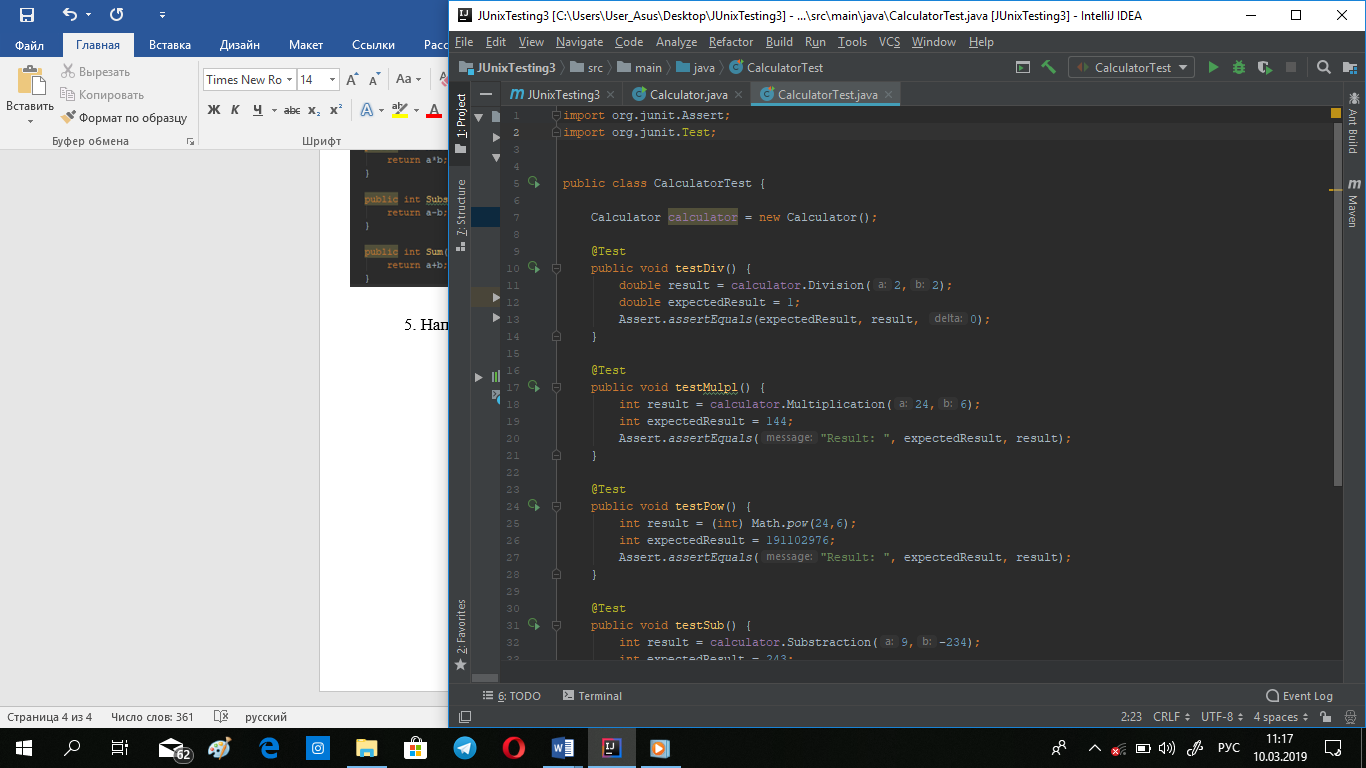
***4. Написать калькулятор в среде IntelliJ IDEA (можно использовать любую другую программу, написанную ранее). Ввод чисел (принимает целые числа) и операцию осуществлять из консоли.***

Создаем классы **Calculator** и **CalculatorTest** в папке проектов (scr 🡪 main 🡪 java).

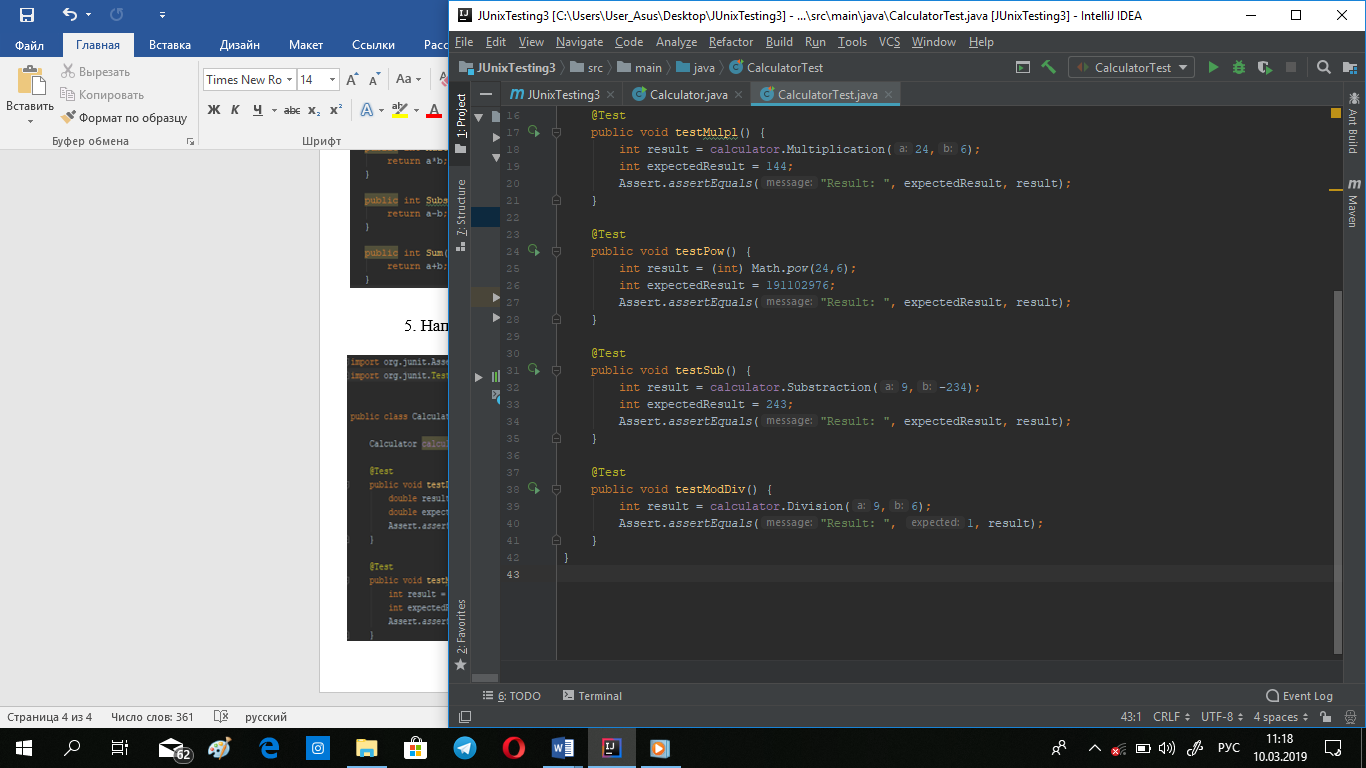
В классе **Calculator** пишем методы Division(), Multiplication(), Substraction(), Sum() и main(): спрашиваем у пользователя два целых числа, сохраняем их в переменные a и b, затем пользователь выбирает, какую операцию он хотел бы совершить с числами. С помощью оператора множественного ветвления switch() выводим на экран тот результат, который потребовал пользователь.

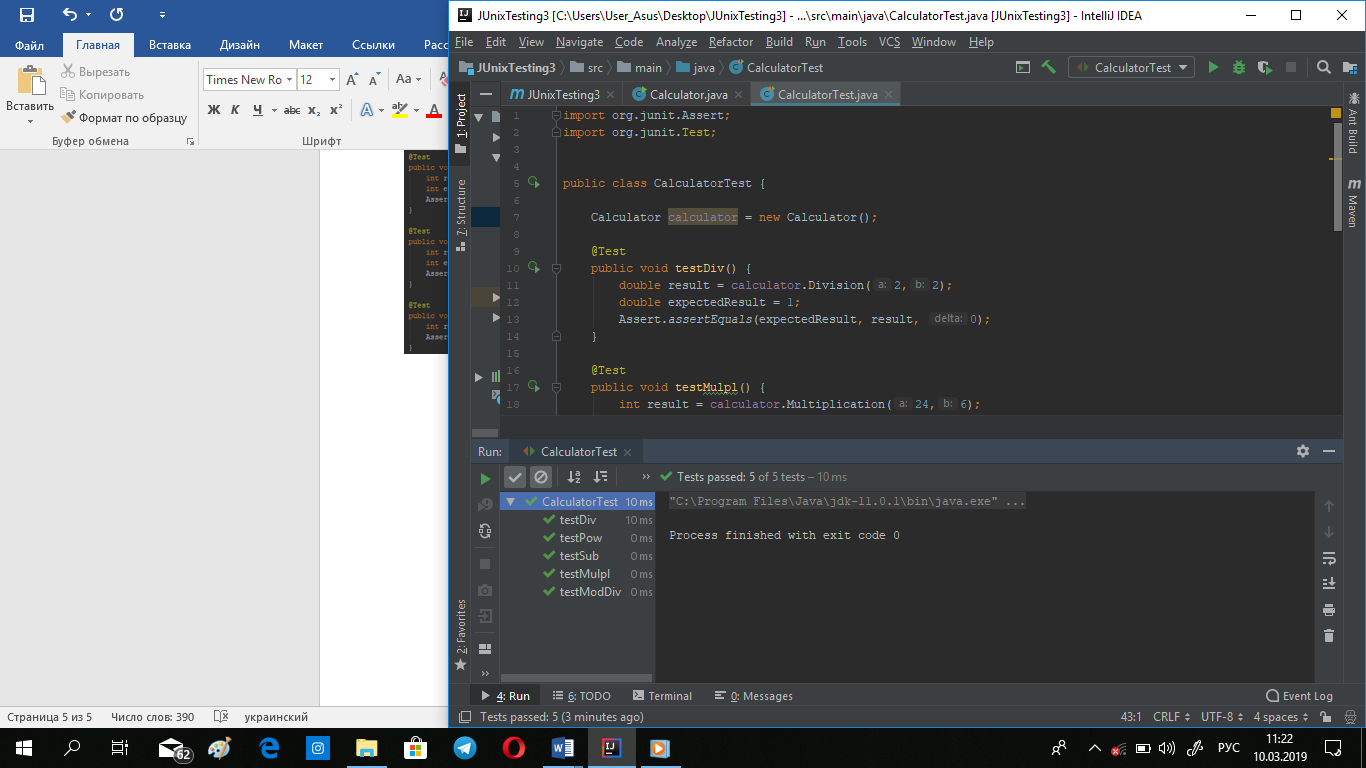


***5. Написать 4-5 тестов с использованием JUnit.***



В классе CalculatorTest пишем пять тестов на пять операций, обозначая их атрибутом @Test: testDiv(), testMulpl(), testPow(), testSub(), testModDiv(). Запускаем класс и на консоли видно, что все тесты пройдены успешно.





***ВЫВОД***

Выполнив лабораторную работу, я изучила фреймверка для модульного тестирования JUnit, ознакомилась со сборщиками проектов, а также самостоятельно написала автоматизированные тестов для программного обеспечения (в моем случае – для программы «Калькулятор»).