Міністерство освіти і науки України

Харківський національний університет імені В.Н. Каразіна

Факультет комп’ютерних наук

**ЛАБОРАТОРНА РОБОТА № 5**

з дисципліни «Математичні методи та технології тестування та верифікації програмного забезпечення»

Тема: «Selenium WebDriver»

Виконав студент 2 курсу

групи КС-21

Безрук Юрій Русланович

Перевірив:

Доцент Нарєжній О. П.

Харків – 2020

Целью данной работы является изучение *Selenium WebDriver* – набора библиотек для различных языков программирования, позволяющего управлять браузером из программы, написанной на этом языке программирования.

*Selenium WebDriver* представляет собой надежный фреймворк автоматизации, способный работать с любым браузером.

Позволяет разработать большой тестовый набор, включающий тесты с достаточно сложной логикой поведения и проверок.

# ХОД РАБОТЫ

Для освоения работы с WebDriver был создан проект Maven в IntelliJ IDEA. В файле ­pom.xml указываем зависимости, необходимые для работы – собственно, selenium, и JUnit. Далее создаётся класс WebTesting, в котором и будет происходить тестирование согласно заданию. Сперва описываются ключевые необходимые элементы – объект класса WebDriver, с помощью которого будет производиться взаимодействие с браузером.

В методе с аннотацией @BeforeClass укажем местонахождение вебдрайвера и создадим его объект внутри программы, а в @AfterClass укажем, что драйвер нужно закрыть после окончания теста.

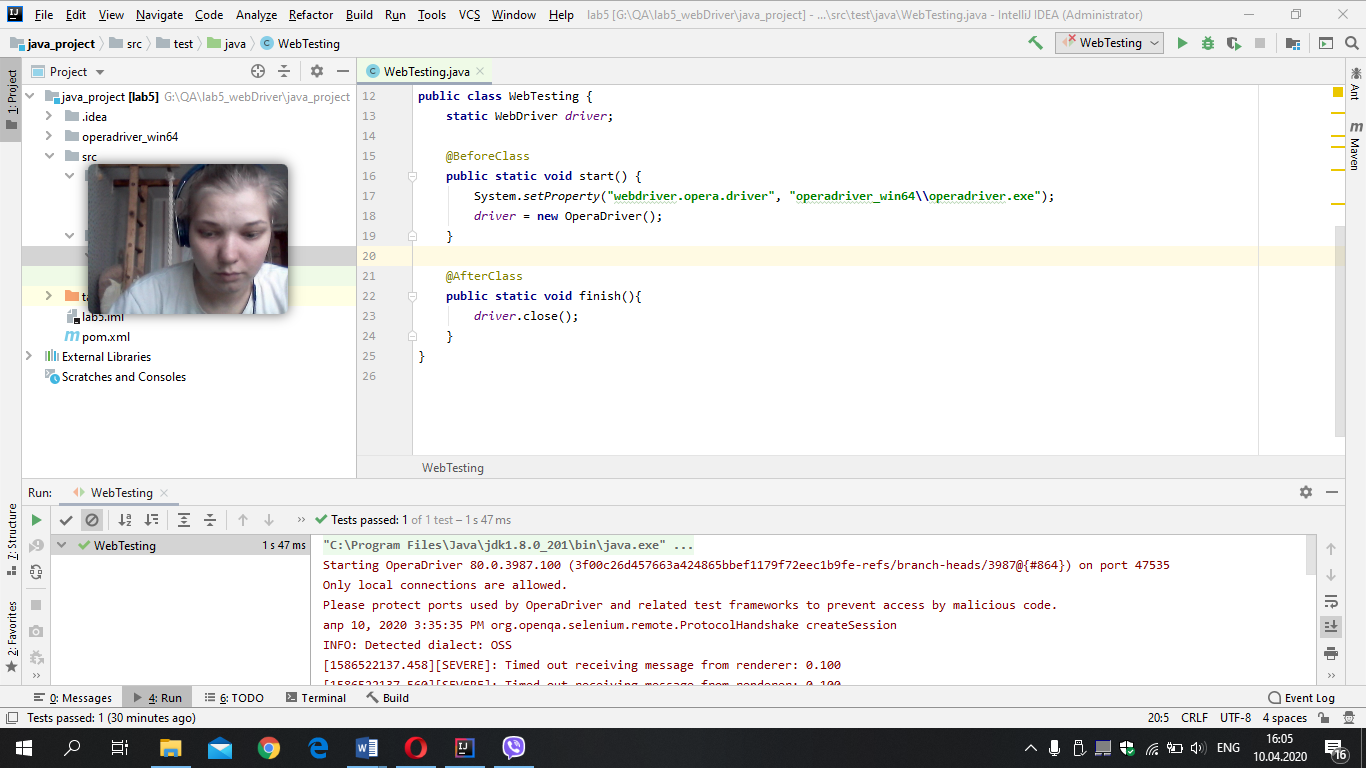


Рисунок 1 - Начало создания класса

Таким образом, при запуске тестов автоматически будет создаваться объект класса WebDriver, а при окончании всех тестов он будет закрываться. Эта часть является общей и необходимой для всех тестов.

В первом тесте необходимо запустить свою страницу (указанную в вариантах), перейти на первого популярного производителя, и проверить, что все отфильтрованные товары являются товарами выбранного производителя. Переход на страницу осуществляется при помощи метода вебдрайвера get(String url); Обращения к элементам на странице производятся при помощи XPath. Получаем WebElement производителя, кликаем на него, и проверяем, что все полученные результаты содержат в имени подстроку-имя производителя.

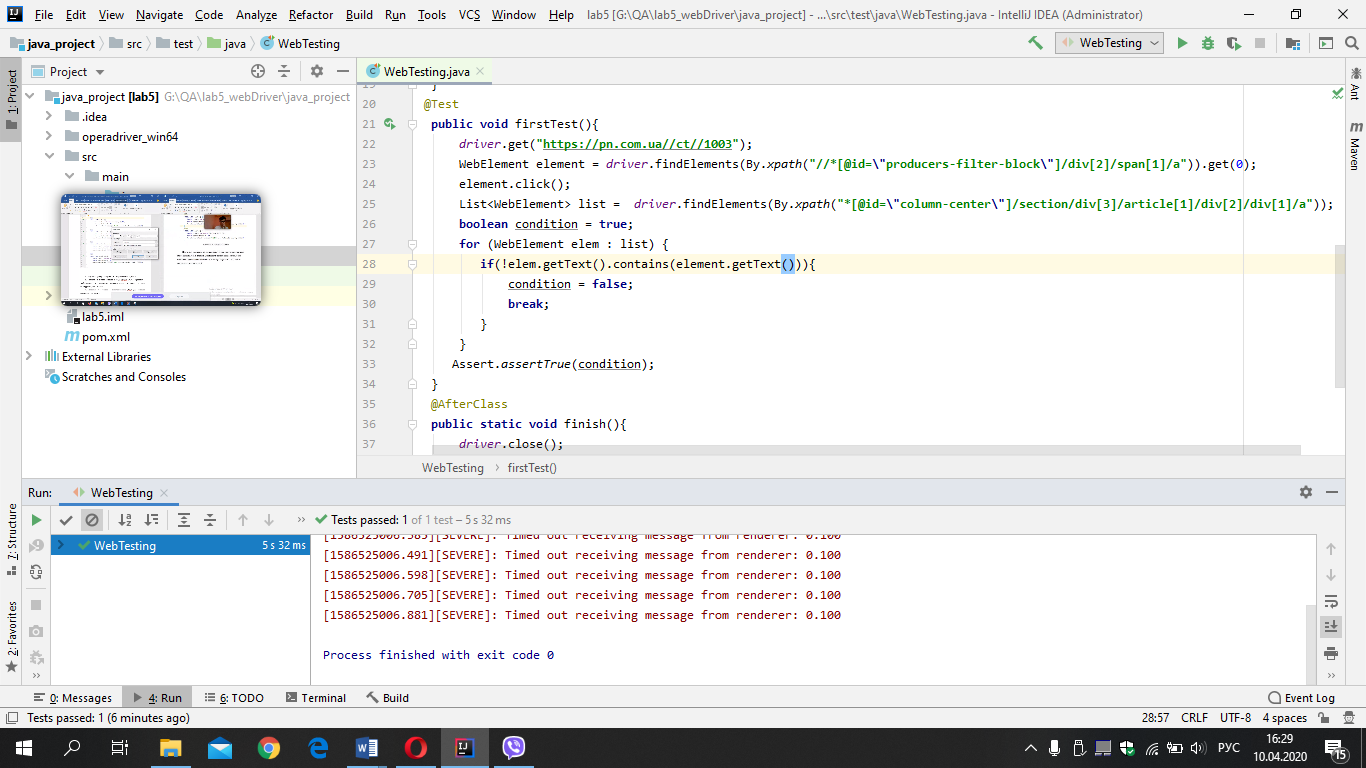


Рисунок 2 - первый тест

Во втором тесте необходимо сверить изначально предлагаемое количество результатов при выборе производителя с полученным после нажатия на фильтр. Для этого сравниваем подпись возле фильтра с посчитанным количеством результатов после его применения.

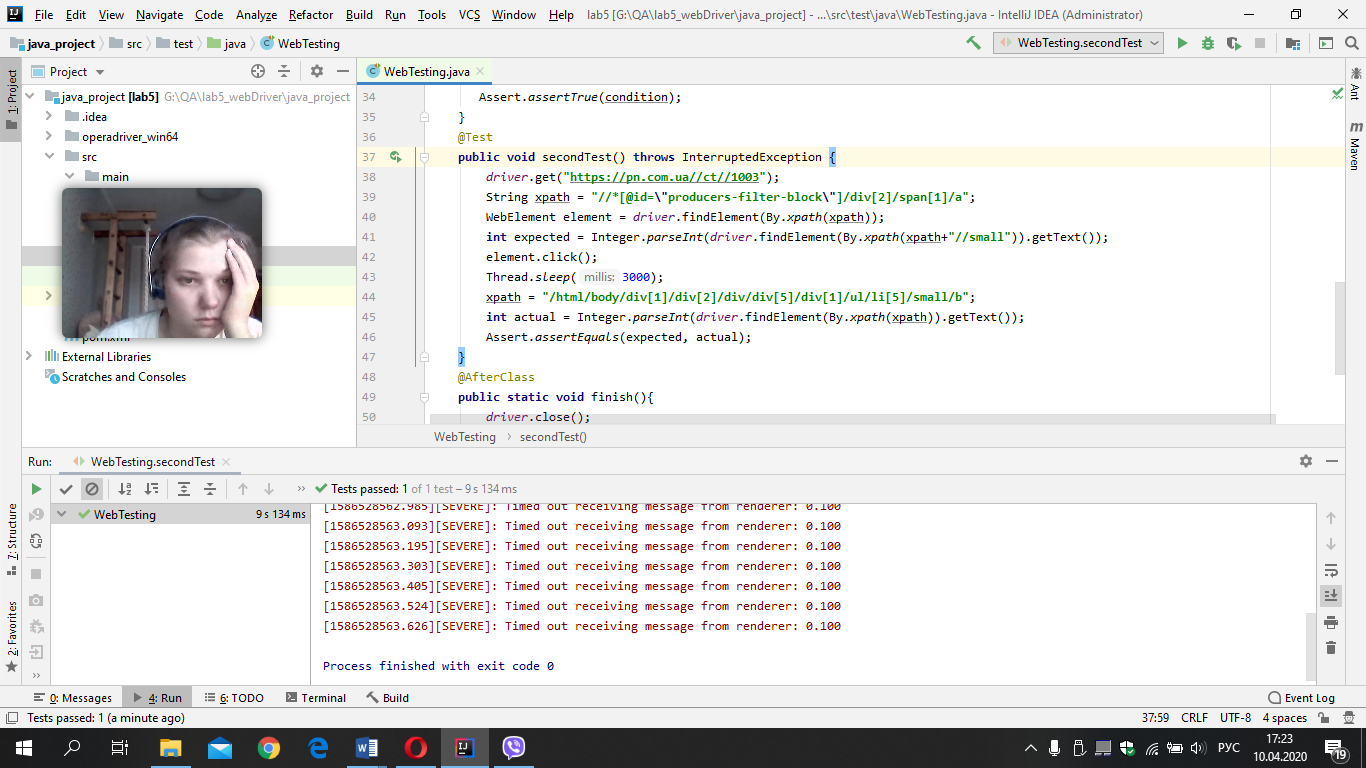


Рисунок 3 - второй тест

В третьем тесте выполняем проверку добавления конкретно двух выбранных товаров на страницу сравнения, и проверка перехода на данную страницу (наличие в адресной строке указания compare).

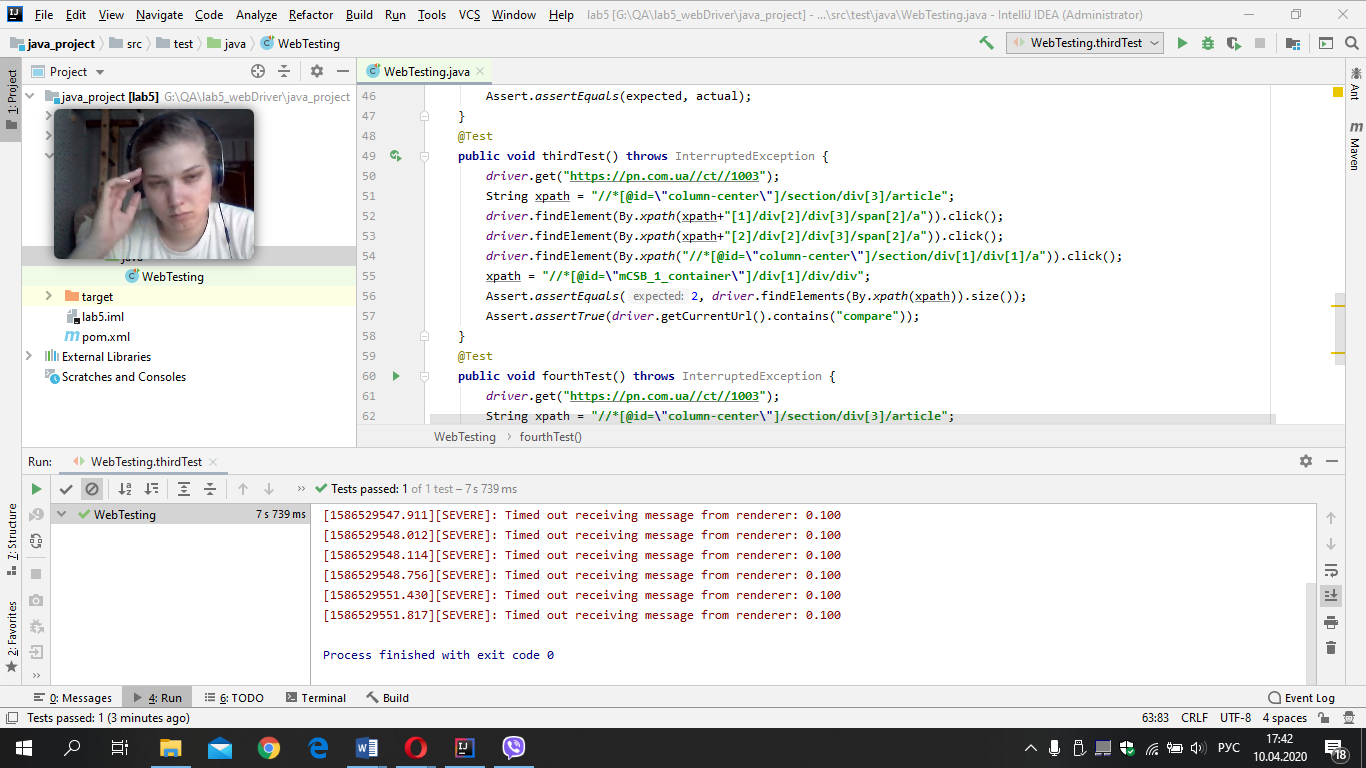


Рисунок 4 - третий тест

В четвертом тесте требуется применить фильтр сортировки товаров по возрастанию цены, и проверить, что они отсортированы. Для этой проверки получаем элементы из списка, и выбираем их цены, вырезая из строк ненужные части. После чего используя Integer.parseInt(), переводим строки с подготовленными цифрами в числа, и проверяем, отсортированы ли они.

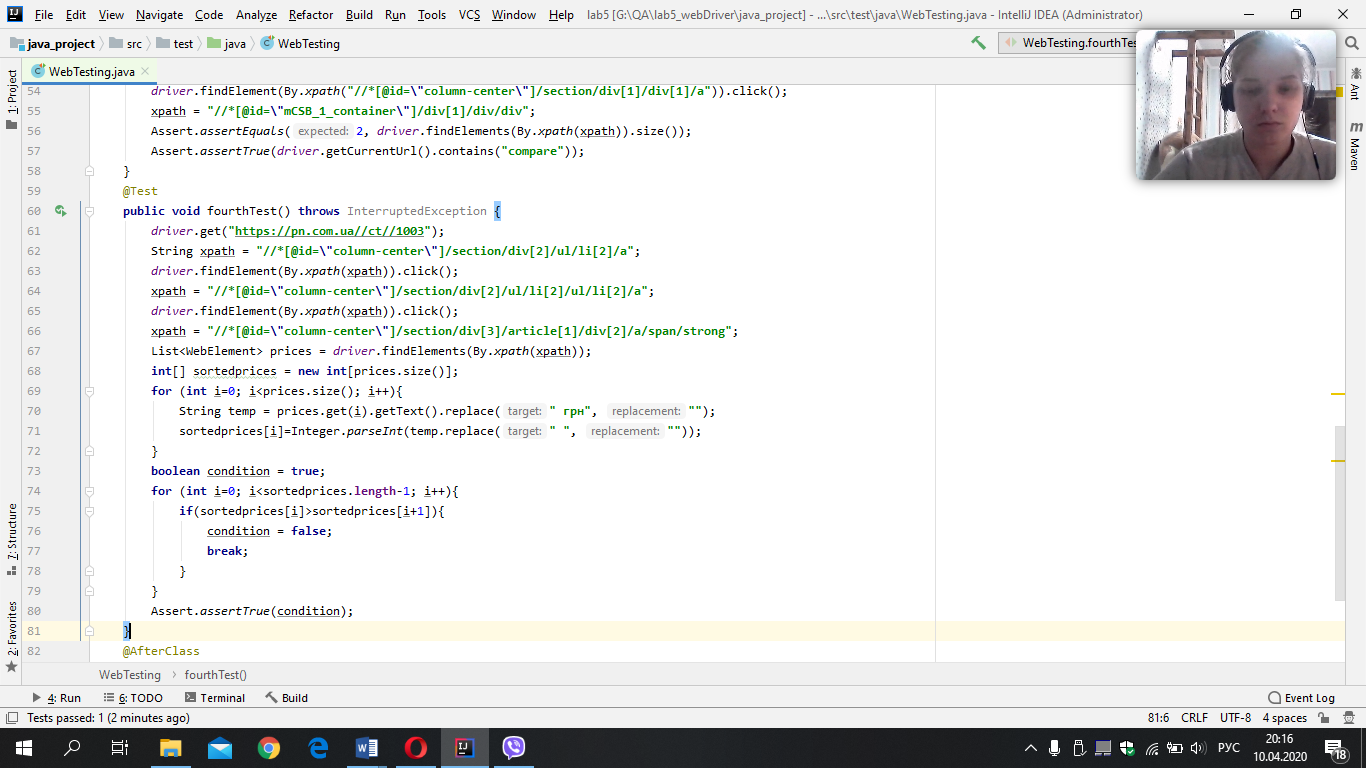


Рисунок 5 - четвертый тест

В пятом тесте от пользователя требуется добавить в сравнение первые 4 товара, после чего нажать кнопку «удалить сравнение», подтвердить действие и таким образом попасть на главную страницу. Проверить, что он находится на главной странице. Для подтверждения действия необходимо выполнить переключение на всплывающее сообщение (driver.swichTo().alert()), и выполнить на этом объекте метод accept(). Для проверки возврата сравниваем текущий URL с исходным.

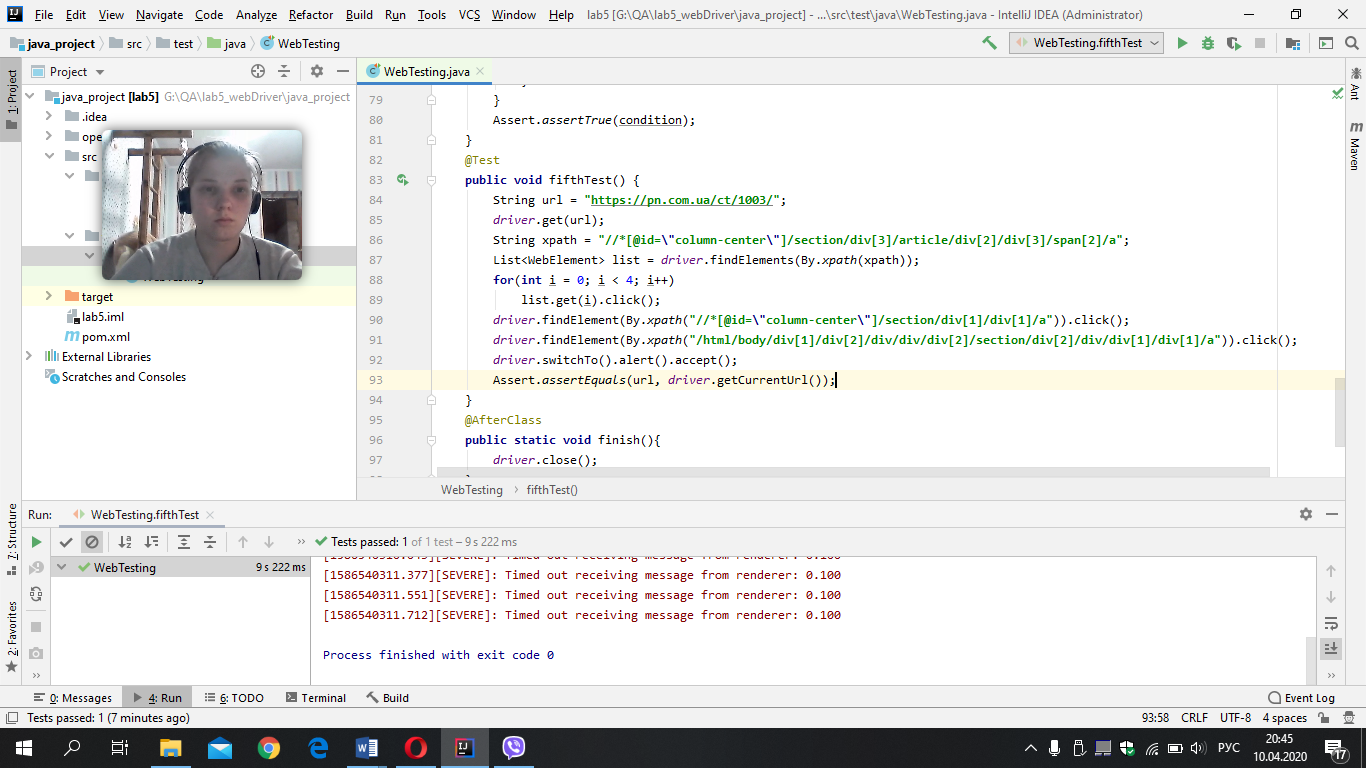


Рисунок 6 - пятый тест

# ВЫВОДЫ

Итак, SeleniumWebDriver представляет собой набор библиотек для различных языков программирования, он позволяет управлять браузером по программе, написанной на данном языке программирования. Это надежный фреймворк автоматизации, который способен работать с любым браузером. С его помощью можно разрабатывать крупные тестовые наборы, в том числе и тесты по достаточно сложной логикой поведения и проверок.

Таким образом, в ходе выполнения лабораторной работы было изучено фреймворк SeleniumWebDriver. Были разработаны пять тестов, с помощью которых выполняется проверка корректности работы сайта. Все тесты были пройдены успешно, ошибок в работе не обнаружено. Исходный код всех тестов сохранен внутри класса WebTesting в виде отдельных методов.