1. Создать Maven проект, подключить WebDriver, Junit 5
2. Реализовать тест кейс:
   1. Перед запуском тестового метода вывести сообщение «Запуск теста»
   2. Зайти на yandex.ru
   3. Кликнуть на «Маркет»
   4. В поле поиска ввести «ноутбуки»
   5. Проверить что поле поиска содержит «ноутбуки»
   6. Нажать «Найти»
   7. В поле Цена ввести значения 100000 – 200000
   8. Выбрать производителей «Apple», «ASUS», «HP», «Xiaomi»
   9. Проверить, что соответствующие галочки проставились
   10. Выбрать процессор «Core i7»
   11. Проверить что цены в таблице в интервале 100 – 200 тысяч
   12. Очистить поле поиска
   13. Ввести «Зеленый слоник»
   14. Нажать найти
   15. Найти товар содержащие в имени «Толстовка», кликнуть на него
   16. Перейти на yandex.ru
   17. При окончании теста вывести сообщение «Тест окончен»
   18. При какой либо ошибке в тесте вывести «Упс, что-то пошло не так» и стектрейс ошибки
3. Как реализовывать:
   1. Прочитать про Junit 5, использовать аннотации @Test, @BeforeAll, @BeforeEach,@AfterAll, @AfterEach и т.д. Использовать assert’ы Junit.

<https://junit.org/junit5/docs/current/user-guide/>

* 1. Прочитать про WebDriver, как запускать, основные методы, класс WebElement, аннотация @FindBy, класс WebDriverWait. При написании тестов не использовать Thread.sleep().

<http://toolsqa.com/selenium-tutorial>

* 1. Для поиска элементов на странице использовать xpath, не копировать что генерит браузер, писать самому

<http://www.cheat-sheets.org/saved-copy/Locators_table_1_0_2.pdf>

* 1. Прочитать про паттерн Page Object, Chain Of Invocations. Использовать их при написании теста.

<https://habr.com/company/jugru/blog/338836>

* 1. Запустить тесты через Maven
  2. Залить получившееся на GitHub