# Задание 1 - Документация

Придумать формальные требования и подготовить план тестирования (тест-план) для проекта **Spring.PetClinic**. Это учебный проект, реализующий простую CRM-систему для работы ветеринарной клиники. Ссылка на репозиторий проекта и его описание (на английском) - <https://github.com/spring-projects/spring-petclinic>

1. Выписать репозиторий и собрать продукт по инструкции. Это поможет придумать требования для него.
2. Придумать 10 пользовательских историй для Spring.PetClinic. Это обобщенные сценарии поведения пользователей, построенные по шаблону *“****КАК*** *<роль в системе>* ***Я ХОЧУ*** *<воспользоваться одной из функций продукта>* ***ЧТОБЫ*** *<получить какую-то выгоду от продукта для себя>”.* Например “Как врач, я хочу залогиниться в системе, чтобы увидеть список моих пациентов на сегодня”.
3. Покрыть каждую пользовательскую историю набором тест-кейсов. В примере из п.1 их можно придумать несколько: *врач вводит правильный логин/пароль; врач вводит логин и неправильный пароль; врач вводит только логин, но не вводит пароль; врач вообще ничего не вводит* - система должна работать корректно во всех этих случаях.
4. Каждый тест-кейс должен состоять из:
   1. Названия - желательно, чтобы оно было кратким и отражало суть теста
   2. Начальных условия для выполнения теста.
   3. Шагов выполнения теста.
   4. Подробного описания ожидаемых результатов - то есть, формальные критерии того, что тест пройден.
5. Оформить работу, как тест-план. Этот документ может быть довольно сложным, но в данном задании, он будет содержать только:
   1. Общее описание продукта
   2. Ключевые требования к продукту (набор из 10 пользовательских историй)
   3. Тест-кейсы для покрытия пользовательских историй. Для нетривиальных тест-кейсов, желательно обоснование - зачем они нужны.

# 

# Задание 2 - Автоматические тесты

Реализовать автотесты для [ozon.ru](https://www.ozon.ru/). Авторизация на сайте не требуется для тестов. Весь проект автотестов разделен на несколько сценариев:

1. Находясь на главной странице [ozon.ru](https://www.ozon.ru/), перейти в каталог “Бытовая техника/Техника для кухни/Кофеварки и кофемашины”, выполнить поиск с диапазоном цен от 10000 до 11000 рублей. Проверить что в результате отобразились только кофеварки с ценами в этом диапазоне. Выбрать сортировку списка по цене, начиная с самого дешевого товара. Добавить первую (самую дешевую) кофеварку в корзину и перейти в нее. Увеличить количество кофеварок до 3 и проверить, что сумма увеличилась в 3 раза. Если нужного количества кофеварок нет в наличии - тест просто должен падать.
2. Сделать тест, аналогичный п.1, только добавить условие “Тип кофеварки=рожковая” и “Приготовление напитка=подогрев чашек”. Для этого, воспользоваться фильтром на сайте, в левой колонке.
3. По сценарию, аналогичному п.1, выбрать самую дешевую кофеварку и добавить ее в “Избранное”. Написать тест, который открывает раздел “Избранное” и проверяет, что кофеварка присутствует в нем и ее цена и скидка не изменились.

**Технологии:**  
Язык программирования - Java

Управление браузером - Selenium/Selenide

Сборка проектов - Maven

Фреймворк для тестирования - TestNG/JUnit

Репортинг - Allure 2

Контроль версий - Git (GitHub)

**Критерии оценки:**

1. Код должен соответствовать Java Code Conventions
2. Автотесты должны быть корректно разделены на тесты и сопутствующие классы.
3. Каждый тест проверяет конкретный сценарий и возвращает систему в исходное состояние.
4. Тесты должны быть стабильными должны запускаться в момент сборки maven’ом
5. Отчет allure должен содержать скриншоты
6. Задание должно быть опубликовано на GitHub

**Литература:**

Java Code Conventions: <http://www.magnumblog.space/other/131-translating-java-code-conventions>

Maven:

<http://proselyte.net/tutorials/maven/>

Allure:

<https://tech.yandex.ru/allure/>

TestNG:

<http://testng.org/doc/documentation-main.html>

Что нужно для старта:  
1. Установить Java JDK (версия 1.8+)  
<https://www.oracle.com/technetwork/java/javase/downloads/jdk11-downloads-5066655.html>

2. Установить Maven

<https://maven.apache.org/download.cgi>

3. Установить любую IDE для Java ([Eclipse](https://www.eclipse.org/downloads/packages/release/mars/2/eclipse-ide-java-ee-developers) or [Intellij Idea](https://www.jetbrains.com/idea/download/))

4. Установить [Git](https://git-scm.com/downloads) и зарегистрироваться на GitHub