

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Ижевский государственный университет имени М.Т. Калашникова»  
(ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)  
Институт непрерывного профессионального образования

Отчет по производственной практике  
по специальности 09.02.03 «Программирование в компьютерных системах»  
по профессиональному модулю ПМ.01 «Разработка программных модулей  
программного обеспечения для компьютерных систем»

Срок практики: с 03.06.2019 г. по 30.06.2019 г.

Работа защищена с оценкой \_\_\_\_\_.

Руководитель производственной практики от учебного заведения:

\_\_\_\_\_ Чернышев К.С «\_\_» \_\_\_\_\_ 2019 г.  
(подпись) (Ф.И.О.)

Студент:

\_\_\_\_\_ Тепляков Г.К «\_\_» \_\_\_\_\_ 2019 г.  
(подпись) (Ф.И.О.)

# Содержание

Задание (на производственную практику)	3
1. Работа с git	4
2. Создание пустого проекта в ide по выбору (рекомендуется intellij idea)	4
3. Добавление к проекту поддержки selenium webdriver	5
4. Добавление генерации allure отчетов	6
5. Создание page object	8
6. Написание 3 сценариев автоматизированного тестирования	9
6.1 Первый сценарий автоматизированного тестирования	9
6.2 Второй сценарий автоматизированного тестирования	12
6.3 Третий сценарий автоматизированного тестирования	16
6.4 Общее для трех сценариев тестирования	19

## Задание (на производственную практику)

В рамках производственной практики предлагается создать проект для автоматизированного тестирования портала на выбор:

1. Создание git репозитория на [gitlab.com](https://gitlab.com)
2. Создание пустого проекта в ide по выбору (рекомендуется intellij idea)
  - Для java: maven, junit
  - Для python: pytest
3. Добавление к проекту поддержки selenium webdriver
4. Добавление генерации allure отчетов
5. Создание page object
6. Написание 3 сценариев автоматизированного тестирования

## 1. Работа с git

Для начала работы с GitLab необходимо создать пользователя путём регистрации на сайте. Далее был создан репозиторий (рисунок 1), который в будущем будет использован для хранения в нём проекта.

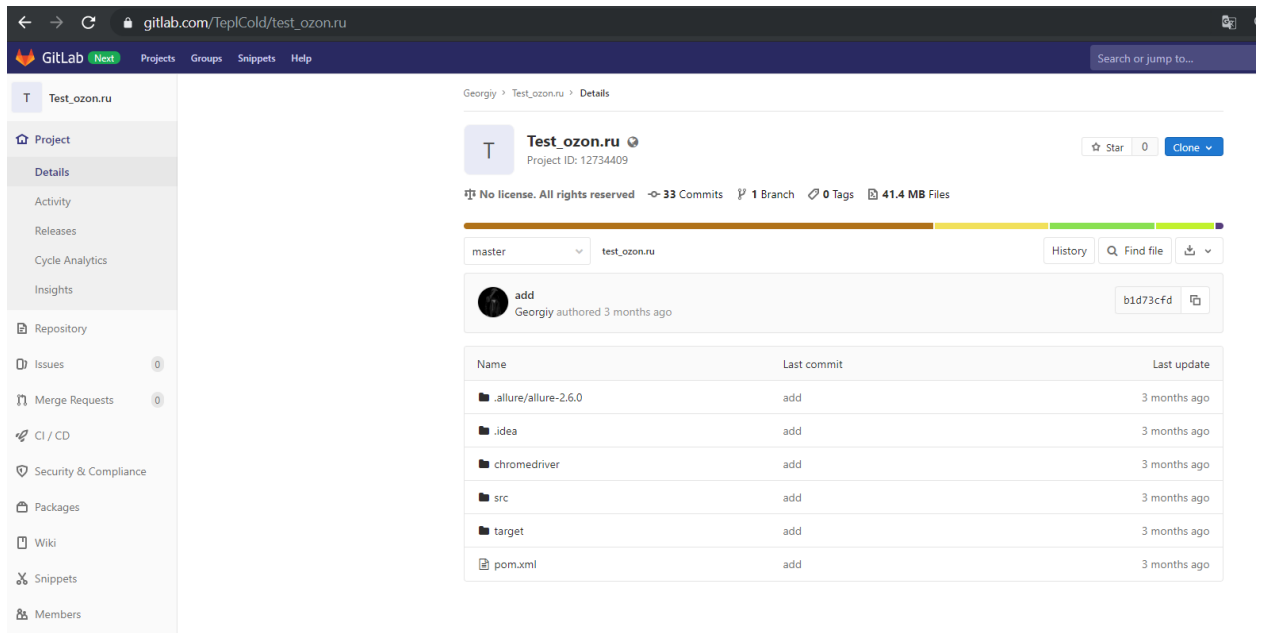


Рисунок 1 – Репозиторий на GitLab

## 2. Создание пустого проекта в ide по выбору (рекомендуется intellij idea)

Из-за того, что для выполнения работы был выбран язык программирования Java, то в дальнейшем была использована среда разработки IntelliJ IDEA. Для начала работы был создан проект под управлением Maven (рисунки 2, 3, 4).

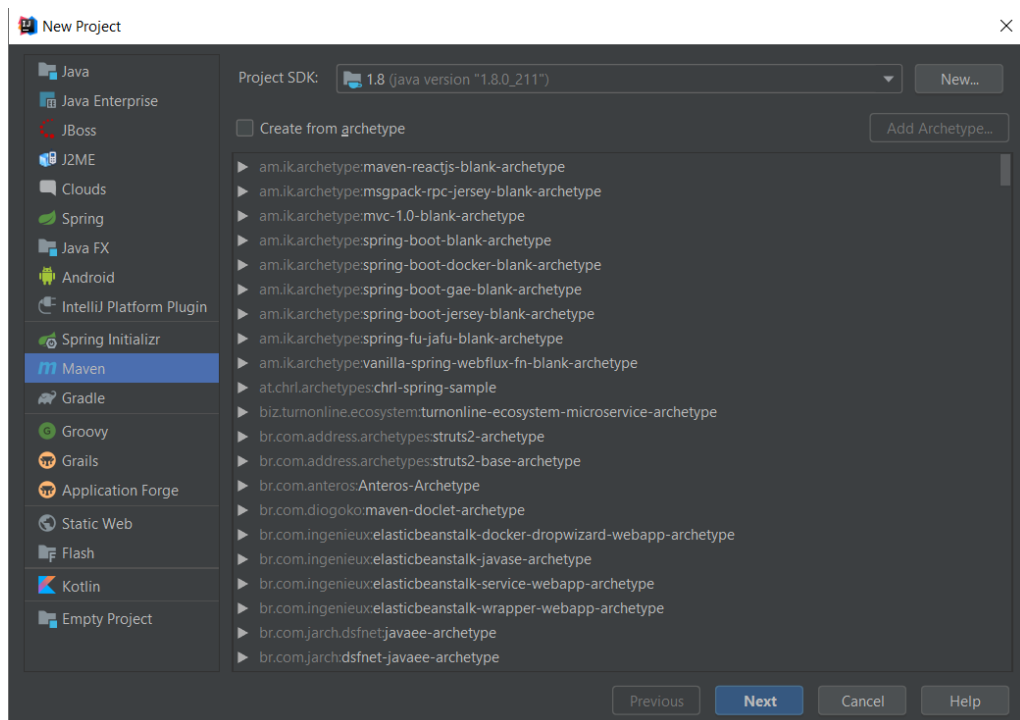


Рисунок 2 – создание проекта под управлением Maven

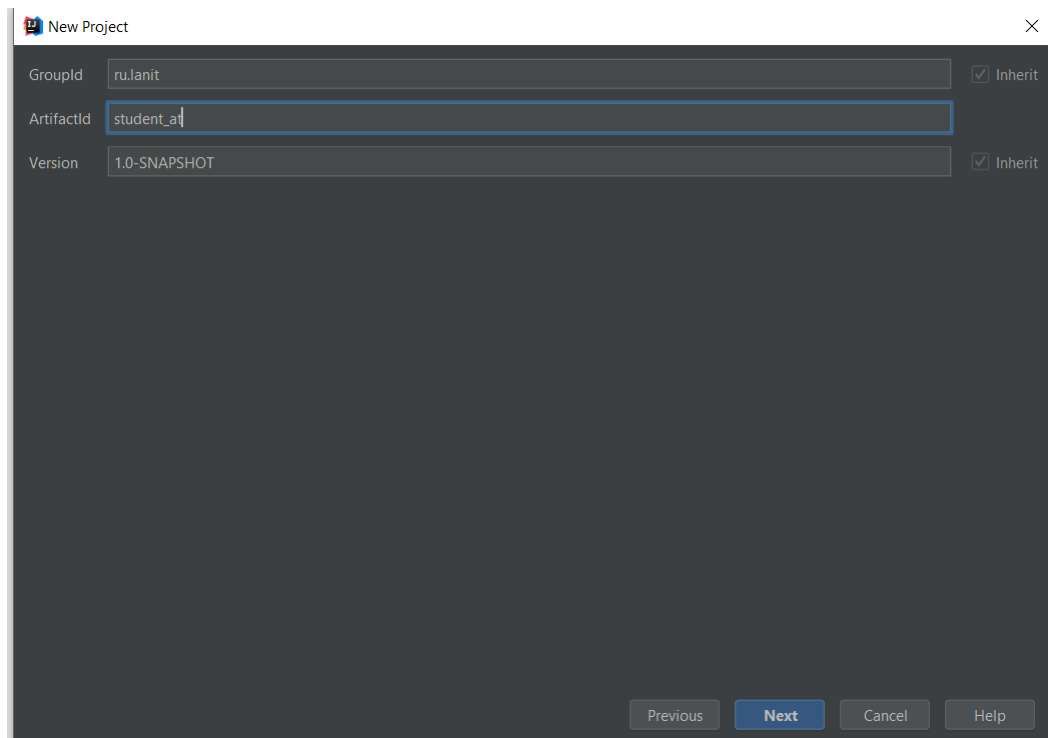


Рисунок 3 – создание проекта под управлением Maven ввод GroupId и ArtifactId

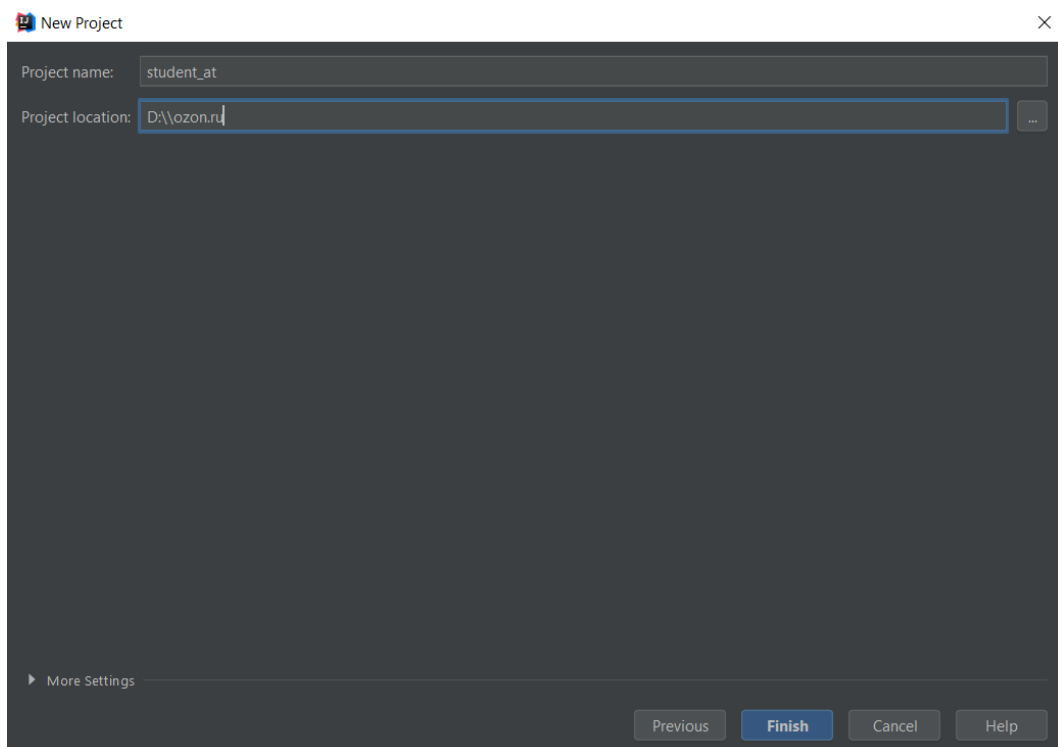


Рисунок 4 – создание проекта под управлением Maven ввод Project name и Project location

### 3. Добавление к проекту поддержки selenium webdriver

Selenium предоставляет возможность производить автоматизированные тесты сайтов. Для этого нужно подключить WebDriver (листинг 1).

```

package testing;

import org.junit.Before;
import org.openqa.selenium.chrome.ChromeDriver;

public class WebDriverSettings {

    public ChromeDriver driver;
    @Before
    public void setUp() {

        System.setProperty("webdriver.chrome.driver", "D:/test_ozon.ru/chromedriver/chromedriver.exe");
        driver = new ChromeDriver();
    }
}

```

Листинг 1 – подключение WebDriver

#### 4. Добавление генерации allure отчетов

К Maven проекту, к которому уже подключен плагин Surefire, можно подключить плагин Allure Report, который будет создавать отчёты на основе проведённого перед этим тестом. Для получения отчёта нужно выполнить две команды в консольной строке IntelliJ IDE: `mvn clean test` (проводит тесты, удаляя данные предыдущих тестов) (рисунок 5) и `mvn allure:serve` (Создаёт временный отчёт теста) (рисунок 6) или `mvn allure:report` (Создаёт файл отчёта теста) (рисунок 7). Сами отчёты, создаваемые в allure report, можно увидеть на рисунках 8, 9 и 10.



Рисунок 5 – ввод команды clean test

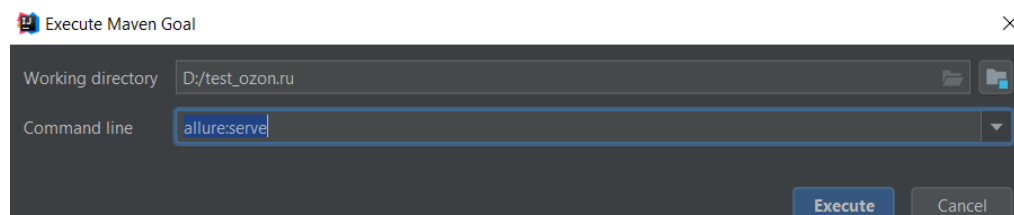


Рисунок 6 – ввод команды allure:serve

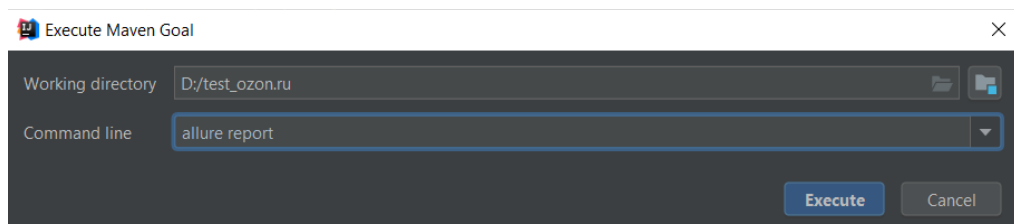


Рисунок 7 – ввод команды allure report

Suites				testing.FirstTest.Test1	
order	name	duration	status	<div>Passed</div> <div>тест для проверки:поиск, добавление и удаление из корзины товара</div>	
Filter by status: 0 0 3 0 0					
▼ testing.FirstTest				1	
	✓ #1 тест для проверки:поиск, добавление и удаление из корзины товара	25s 645ms			
> testing.SecondTest				1	
> testing.ThirdTest				1	

testing.FirstTest.Test1	
<div>Passed</div> <div>тест для проверки:поиск, добавление и удаление из корзины товара</div>	
Overview	History Retries
Severity: normal	
Duration: 25s 645ms	
Execution	
▼ Test body	
✓ Заходим на сайт Ozon 1 parameter, 1 attachment	10s 141ms
driver ChromeDriver: chrome on XP (e27224e9f8d338a326d8f57d295...	
> скрин: Главная страница	504.2 KB ✕
✓ Поиск товара 'Bluetooth-гарнитура Xiaomi Mini Earphones' 1 parameter, 1 attachment	5s 575ms
driver ChromeDriver: chrome on XP (e27224e9f8d338a326d8f57d295...	
> скрин: Поиск товара по названию	165.9 KB ✕
✓ добавление в корзину 1 parameter, 1 attachment	944ms
driver ChromeDriver: chrome on XP (e27224e9f8d338a326d8f57d295...	
> скрин: добавление в корзину	116 KB ✕
✓ переход в Корзину 1 parameter, 1 attachment	2s 029ms
driver ChromeDriver: chrome on XP (e27224e9f8d338a326d8f57d295...	
> скрин: товара в корзине	66.5 KB ✕
✓ нажатие на кнопку удаление из Корзины 1 parameter, 1 attachment	762ms
driver ChromeDriver: chrome on XP (e27224e9f8d338a326d8f57d295...	
> скрин: удаления из Корзины	85.3 KB ✕
✓ подтверждение удаления из Корзины 1 parameter, 1 attachment	652ms
driver ChromeDriver: chrome on XP (e27224e9f8d338a326d8f57d295...	
> скрин: корзины после удаления	90.5 KB ✕
✓ Выход из браузера 1 parameter	1s 876ms
driver ChromeDriver: chrome on XP (e27224e9f8d338a326d8f57d295...	

Рисунок 8 – отчет создаваемый allure report – тест 1

Suites				testing.SecondTest.Test2	
order	name	duration	status	<div>Passed</div> <div>тест для проверки:поиск, добавление, сравнения товаров</div>	
Filter by status: 0 0 3 0 0					
▼ testing.FirstTest				1	
	✓ #1 тест для проверки:поиск, добавление и удаление из корзины товара	25s 645ms			
▼ testing.SecondTest				1	
	✓ #1 тест для проверки:поиск, добавление, сравнения товаров	37s 753ms			
> testing.ThirdTest				1	

testing.SecondTest.Test2	
<div>Passed</div> <div>тест для проверки:поиск, добавление, сравнения товаров</div>	
Overview	History Retries
Severity: normal	
Duration: 37s 753ms	
Execution	
▼ Test body	
✓ Заходим на сайт Ozon 1 parameter, 1 attachment	6s 980ms
driver ChromeDriver: chrome on XP (34805be9032948789e33c37502b009c)	
> скрин: Главная страница	503.2 KB ✕
✓ Поиск товара 'Минимойка Karcher K 3 EU' 1 parameter, 2 attachments	4s 050ms
driver ChromeDriver: chrome on XP (34805be9032948789e33c37502b009c)	
> скрин: Поиск товара 'Минимойка Karcher K 3 EU'	136.9 KB ✕
> скрин: страницы с результатом поиска Karcher_K_3	101 KB ✕
✓ Переход на страницу с товаром Karcher_K_3 1 parameter, 1 attachment	3s 476ms
driver ChromeDriver: chrome on XP (34805be9032948789e33c37502b009c)	
> скрин: страницы с товаром Karcher_K_3	300.6 KB ✕
✓ добавить в сравнение KarcherK3 1 parameter, 1 attachment	857ms
driver ChromeDriver: chrome on XP (34805be9032948789e33c37502b009c)	
> скрин: добавление в сравнение KarcherK3	294.2 KB ✕
✓ Поиск товара 'Минимойка Karcher K4 Compact, 1.637-500.0, желтый, черный' 1 parameter, 2 attachments	3s 922ms
driver ChromeDriver: chrome on XP (34805be9032948789e33c37502b009c)	
> скрин: Поиск товара 'Минимойка Karcher K4 Compact'	165.2 KB ✕
> скрин: страницы после поиска Karcher K4 Compact	128.5 KB ✕
✓ Переход на страницу с товаром Karcher K4 Compact 1 parameter, 1 attachment	2s 954ms
driver ChromeDriver: chrome on XP (34805be9032948789e33c37502b009c)	
> скрин: страницы с товаром Karcher K4 Compact	360 KB ✕
✓ добавить в сравнение Karcher K4 Compact 1 parameter, 1 attachment	978ms
driver ChromeDriver: chrome on XP (34805be9032948789e33c37502b009c)	
> скрин: добавление в сравнение Karcher K4 Compact	371.7 KB ✕
✓ перейти на страницу сравнение товаров 1 parameter, 2 attachments	8s 049ms
driver ChromeDriver: chrome on XP (34805be9032948789e33c37502b009c)	
> скрин: наведение на роупر меню	331.7 KB ✕
> скрин: страницы 'сравнение товаров'	124.5 KB ✕
✓ очистить список товаров сравнения 1 parameter, 1 attachment	620ms
driver ChromeDriver: chrome on XP (34805be9032948789e33c37502b009c)	
> скрин: страницы после очищения списка сравнения	73 KB ✕
✓ Выход из браузера 1 parameter	3s 848ms

Рисунок 9 – отчет создаваемый allure report – тест 2

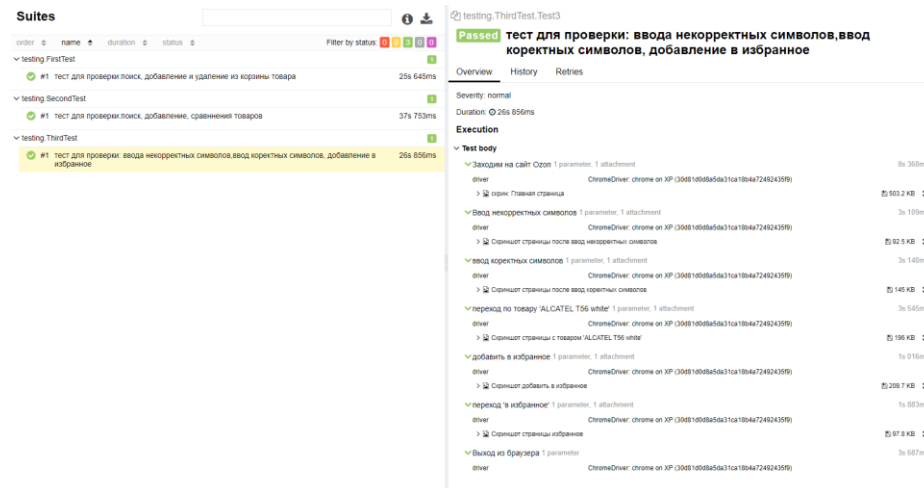


Рисунок 10 – отчет создаваемый allure report – тест 3

## 5. Создание page object

Page object представляет из себя макет страницы. Такой макет можно сделать, используя инструментальный selenium. Рассмотрим на примере page object, на примере страницы поиска. Для начала объявим веб-элементов, которые будут использоваться на сайте, в ходе тестирования (листинг 2). В page object с использованием selenium принято использовать конструкцию `@FindBy(свойство искомого элемента)`. Это нужно для того, чтобы использовать одни и те же переменные, которые были инициализированы однажды при заходе на страницу, многократно.

```
@FindBy(xpath =
"//*[@id=\"__layout\"]/div[1]/div[1]/header[1]/div[1]/div[3]/div[1]/div[2]
]/div[1]/div[1]/div[1]/div[1]/input[1]")
private WebElement searchBox;
```

Листинг 2 – объявление переменных веб-элементов

Теперь нужно написать функцию (листинг 3), которая будет создавать экземпляр page object во время теста (листинг 4). Это нужно для того, чтобы инициализировать веб-элементы, помеченные с помощью `@FindBy`, а также использовать другие функции, которые присутствуют в page object.

```
public Search(WebDriver driver) {
    if (!driver.getCurrentUrl().contains(OZON_URL)) {
        throw new IllegalStateException("This is not the page you are
expected");
    }
    PageFactory.initElements(driver, this);
    this.waitFor = new WebDriverWait(driver, 3, 300);
}
```

Листинг 3 – Функция для инициализации элементов на страницы

```
Search search = new Search(driver);
```

Листинг 4 – Создание экземпляра во время теста

И в конце идут функции, которые реализуют действия с элементами на странице (листинг 5). Из вызова данных функции и состоят тесты.



```

public void Search(String item){
    searchBox.clear();
    waitFor.until(ExpectedConditions.visibilityOf(searchBox));
    searchBox.clear();
    searchBox.sendKeys(item);
    searchBox.sendKeys(Keys.ENTER);
    searchBox.clear();
}

```

Листинг 5 – Функции, производящие действия на странице

## 6. Написание 3 сценариев автоматизированного тестирования

### 6.1 Первый сценарий автоматизированного тестирования

```

package testing;
import ForTest.ForFirstTest;
import ForTest.Search;
import io.qameta.allure.Attachment;
import io.qameta.allure.Step;
import io.qameta.allure.junit4.DisplayName;
import org.apache.commons.io.output.ByteArrayOutputStream;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.TestWatcher;
import org.junit.runner.Description;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.remoteAugmenter;
import ru.yandex.qatools.ashot.AShot;
import ru.yandex.qatools.ashot.Screenshot;
import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.io.IOException;
import static org.junit.Assert.assertEquals;

public class FirstTest extends WebDriverSettings {

    @Attachment(value = "{name}", type = "image/png")
    public static byte[] getFullScreenshot(WebDriver driver, String name)
    throws IOException {
        WebDriver augmented = new Augmenter().augment(driver);
        Screenshot shot = new AShot().takeScreenshot(augmented);
        BufferedImage img = shot.getImage();
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ImageIO.write(img, "png", baos);
        baos.flush();
        byte[] imageInByte = baos.toByteArray();
        baos.close();
        return imageInByte;
    }

    @Rule
    public TestWatcher screenshotOnFailure = new TestWatcher() {
        @Override
        protected void failed(Throwable e, Description description) {
            try {
                getFullScreenshot(driver, "Скриншот при ошибке");
            } catch (IOException f) { }
        }
    };

    @Step("Заходим на сайт Ozon")
    public static void IntoOzon(WebDriver driver) throws IOException
    {

```

```

        driver.get("https://www.ozon.ru/");
        assertEquals("OZON – интернет-магазин. Миллионы товаров по
выгодным ценам", driver.getTitle());

        getFullScreenshot(driver, "скрин: Главная страница");
    }

    @Step("Поиск товара 'Bluetooth-гарнитура Xiaomi Mini Earphones'")
    public static void Search (WebDriver driver) throws IOException
    {
        Search search = new Search(driver);
        search.Search("Bluetooth-гарнитура Xiaomi Mini Earphones");

        getFullScreenshot(driver, "скрин: Поиск товара по названию");
    }

    @Step("добавление в корзину")
    public static void InGarbage(WebDriver driver) throws IOException
    {
        JavascriptExecutor jse = ((JavascriptExecutor) driver);
        jse.executeScript("scroll(0, 320);");
        ForFirstTest InGarbage = new ForFirstTest(driver);
        InGarbage.InGarbage();

        getFullScreenshot(driver, "скрин: добавление в корзину");
    }

    @Step("переход в Корзину")
    public static void Basket(WebDriver driver) throws IOException,
    InterruptedException {
        ForFirstTest Basket = new ForFirstTest(driver);
        Basket.Basket();

        Thread.sleep(100);
        JavascriptExecutor jse = ((JavascriptExecutor) driver);
        jse.executeScript("scroll(0, 300);");
        getFullScreenshot(driver, "скрин: товара в корзине");
    }

    @Step("нажатие на кнопку удаление из Корзину")
    public static void Deletion(WebDriver driver) throws IOException
    {
        ForFirstTest Deletion = new ForFirstTest(driver);
        Deletion.Deletion();

        getFullScreenshot(driver, "скрин: удаления из Корзины");
    }

    @Step("подтверждение удаления из Корзины")
    public static void Confirm(WebDriver driver) throws IOException
    {
        ForFirstTest Confirm = new ForFirstTest(driver);
        Confirm.Confirm();

        getFullScreenshot(driver, "скрин: корзины после удаления");
    }

    @Step("Выход из браузера")
    public static void WaitAndQuit(WebDriver driver) throws
    InterruptedException
    {
        Thread.sleep(1000);
        driver.close();
    }

```

```

        driver.quit();
        System.out.println("test close");
    }

    @Test
    @DisplayName("тест для проверки:поиск, добавление и удаление из корзины товара")
    public void Test1() throws InterruptedException, IOException
    {
        IntoOzon(driver);
        Search(driver);
        InGarbage(driver);
        Basket(driver);
        //ScreenBasket(driver);
        Deletion(driver);
        Confirm(driver);
        WaitAndQuit(driver);
    }
}

```

#### Листинг 7 - листинг файла FirstTest.java

```

package ForTest;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;

public class ForFirstTest {
    private static String OZON_URL = "https://www.ozon.ru";

    WebDriverWait waitFor;

    @FindBy(xpath = "//*[@id=\"addToCartWithCount\"]/button[1]")
    private WebElement InGarbage;

    @FindBy(xpath =
"//*[@id=\"__layout\"]/div[1]/div[1]/header[1]/div[2]/div[3]/div[4]")
    private WebElement Basket;

    @FindBy(xpath =
"//*[@id=\"__layout\"]/div[1]/div[2]/div[1]/div[1]/div[2]/div[1]/div[2]/div[1]/div[2]/div[2]/div[2]/div[2]/div[3]/div[2]/span[1]/button[1]")
    private WebElement Deletion;

    @FindBy(xpath =
"//*[@id=\"__layout\"]/div[1]/div[2]/div[1]/div[1]/div[2]/div[1]/div[2]/div[2]/div[1]/div[1]/div[1]/div[1]/section[1]/div[3]/button[1]")
    private WebElement Confirm;

    public ForFirstTest(WebDriver driver) {
        if (!driver.getCurrentUrl().contains(OZON_URL)) {
            throw new IllegalStateException("This is not the page you are expected");
        }
        PageFactory.initElements(driver, this);
        this.waitFor = new WebDriverWait(driver, 3, 300);
    }

    public void InGarbage() { InGarbage.click(); }
    public void Basket() { Basket.click(); }
}

```

```

        public void Deletion(){ Deletion.click();}
        public void Confirm(){Confirm.click();}
    }

```

#### Листинг 8 - листинг файла ForFirstTest.java

### 6.2 Второй сценарий автоматизированного тестирования

```

package testing;

import ForTest.ForSecondTest;
import ForTest.Search;
import io.qameta.allure.Attachment;
import io.qameta.allure.Step;
import io.qameta.allure.junit4.DisplayName;
import org.apache.commons.io.output.ByteArrayOutputStream;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.TestWatcher;
import org.junit.runner.Description;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.remote.Augmenter;
import ru.yandex.qatools.ashot.AShot;
import ru.yandex.qatools.ashot.Screenshot;
import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.io.IOException;
import static org.junit.Assert.assertEquals;

public class SecondTest extends WebDriverSettings {

    @Attachment(value = "{name}", type = "image/png")
    public static byte[] getFullScreenshot(WebDriver driver, String name)
    throws IOException {
        WebDriver augmented = new Augmenter().augment(driver);
        Screenshot shot = new AShot().takeScreenshot(augmented);
        BufferedImage img = shot.getImage();
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ImageIO.write(img, "png", baos);
        baos.flush();
        byte[] imageInByte = baos.toByteArray();
        baos.close();
        return imageInByte;
    }

    @Rule
    public TestWatcher screenshotOnFailure = new TestWatcher() {
        @Override
        protected void failed(Throwable e, Description description) {
            try {
                getFullScreenshot(driver, "Скриншот при ошибке");
            } catch (IOException f) { }
        }
    };

    @Step("Заходим на сайт Ozon")
    public static void IntoOzon(WebDriver driver) throws IOException
    {
        driver.get("https://www.ozon.ru/");
        assertEquals("OZON – интернет-магазин. Миллионы товаров по выгодным ценам", driver.getTitle());
    }

```

```

        getFullScreenshot(driver, "скрин: Главная страница");
    }

    @Step("Поиск товара 'Минимойка Karcher K 3 EU'")
    public static void Search_Karcher_K_3 (WebDriver driver) throws
    IOException, InterruptedException {
        Search search = new Search(driver);
        search.Search("Минимойка Karcher K 3 EU");

        getFullScreenshot(driver, "скрин: Поиск товара 'Минимойка Karcher
K 3 EU'");

        Thread.sleep(100);
        JavascriptExecutor jse = ((JavascriptExecutor) driver);
        jse.executeScript("scroll(0, 300);");
        getFullScreenshot(driver, "скрин: страницы с результатом поиска
Karcher_K_3");
    }

    @Step("Переход на страницу с товаром Karcher_K_3")
    public static void description_Karcher_K_3 (WebDriver driver) throws
    IOException
    {
        ForSecondTest description_Karcher_K_3 = new
        ForSecondTest(driver);
        description_Karcher_K_3.description();

        JavascriptExecutor jse = ((JavascriptExecutor) driver);
        jse.executeScript("scroll(0, 300);");
        getFullScreenshot(driver, "скрин: страницы с товаром
Karcher_K_3");
    }

    @Step("добавить в сравнение KarcherK3")
    public static void ComparisonKarcherK3(WebDriver driver) throws
    IOException
    {
        ForSecondTest ComparisonKarcherK3 = new ForSecondTest(driver);
        ComparisonKarcherK3.Comparison();

        getFullScreenshot(driver, "скрин: добавление в сравнение
KarcherK3");
    }

    @Step("Поиск товара 'Минимойка Karcher K4 Compact, 1.637-500.0,
желтый, черный'")
    public static void Search_Karcher_K4_Compact (WebDriver driver )
    throws IOException, InterruptedException {
        Search search = new Search(driver);
        search.Search("Минимойка Karcher K4 Compact, 1.637-500.0, желтый,
черный");

        getFullScreenshot(driver, "скрин: Поиск товара 'Минимойка Karcher
K4 Compact'");

        Thread.sleep(100);
        JavascriptExecutor jse = ((JavascriptExecutor) driver);
        jse.executeScript("scroll(0, 300);");
        getFullScreenshot(driver, "скрин: страницы после поиска Karcher
K4 Compact");
    }

```

```

        @Step("Переход на страницу с товаром Karcher K4 Compact")
        public static void description_Karcher_K4_Compact (WebDriver driver)
        throws IOException
        {
            ForSecondTest description_Karcher_K4_Compact = new
            ForSecondTest(driver);
            description_Karcher_K4_Compact.description();

            JavascriptExecutor jse = ((JavascriptExecutor) driver);
            jse.executeScript("scroll(0, 300);");
            getFullScreenshot(driver, "скрин: страницы с товаром Karcher K4
            Compact");
        }

        @Step("добавить в сравнение Karcher K4 Compact")
        public static void ComparisonKarcher_K4_Compact(WebDriver
        driver)throws IOException
        {
            ForSecondTest Comparison_Karcher_K4_Compact = new
            ForSecondTest(driver);
            Comparison_Karcher_K4_Compact.Comparison();

            getFullScreenshot(driver, "скрин: добавление в сравнение Karcher
            K4 Compact");
        }

        @Step("перейти на страницу сравнение товаров")
        public static void Comparison(WebDriver driver)throws IOException
        {
            WebElement myElement =
            driver.findElement(By.xpath("//*[@id=\"__layout\"]/div[1]/div[1]/header[1
            ]/div[2]/div[3]/div[1]/div[1]"));
            Actions builder = new Actions(driver);
            builder.moveToElement(myElement).build().perform();
            getFullScreenshot(driver, "скрин: наведение на popup menu");

            driver.findElement(By.xpath("//*[@id=\"__layout\"]/div[1]/div[1]/header[1
            ]/div[2]/div[3]/div[1]/div[1]/div[2]/div[2]/a[1]")).click();

            JavascriptExecutor jse = ((JavascriptExecutor) driver);
            jse.executeScript("scroll(0, 200);");
            getFullScreenshot(driver, "скрин: страницы 'сравнение товаров'");
        }

        @Step("очистить список товаров сравнения")
        public static void ComparisonClear(WebDriver driver)throws
        IOException
        {
            ForSecondTest Claer = new ForSecondTest(driver);
            Claer.Claer();

            getFullScreenshot(driver, "скрин: страница после очищения списка
            сравнения");
        }

        @Step("Выход из браузера")
        public static void WaitAndQuit(WebDriver driver) throws
        InterruptedException
        {
            Thread.sleep(1000);
            driver.close();
            driver.quit();
        }

```

```

        System.out.println("test close");
    }

    @Test
    @DisplayName("тест для проверки:поиск, добавление, сравнения товаров")
    public void Test2() throws InterruptedException, IOException
    {
        IntoOzon(driver);
        Search_Karcher_K_3(driver);
        description_Karcher_K_3(driver);
        ComparisonKarcherK3(driver);
        Search_Karcher_K4_Compact(driver);
        description_Karcher_K4_Compact(driver);
        ComparisonKarcher_K4_Compact(driver);
        Comparison(driver);
        ComparisonClear(driver);
        WaitAndQuit(driver);
    }
}

```

**Листинг 9 - листинг файла SecondTest.java**

```

package ForTest;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;

public class ForSecondTest {
    private static String OZON_URL = "https://www.ozon.ru";

    WebDriverWait waitFor;

    @FindBy(xpath = "//*[@id=\"name\"]")
    private WebElement description;

    @FindBy(xpath =
"//*[@id=\"__layout\"]/div[1]/div[1]/div[4]/div[1]/div[2]/div[3]/div[1]/div[2]/div[2]/div[1]/span[1]/button[1]")
    private WebElement Comparison;

    @FindBy(xpath =
"//*[@id=\"PageCenter\"]/div[2]/div[1]/div[1]/div[2]/div[2]/div[1]/div[1]/div[1]/div[1]/span[1]")
    private WebElement clear;

    public ForSecondTest(WebDriver driver) {
        if (!driver.getCurrentUrl().contains(OZON_URL)) {
            throw new IllegalStateException("This is not the page you are expected");
        }
        PageFactory.initElements(driver, this);
        this.waitFor = new WebDriverWait(driver, 3, 300);
    }
    public void description() {description.click();}
    public void Comparison() {Comparison.click();}
    public void Claer() {clear.click();}
}

```

**Листинг 10 - листинг файла ForSecondTest.java**

### 6.3 Третий сценарий автоматизированного тестирования

```
package testing;

import ForTest.Search;
import ForTest.ForThirdTest;
import io.qameta.allure.Attachment;
import io.qameta.allure.Step;
import io.qameta.allure.junit4.DisplayName;
import org.apache.commons.io.output.ByteArrayOutputStream;
import org.junit.Rule;
import org.junit.Test;
import org.junit.rules.TestWatcher;
import org.junit.runner.Description;
import org.openqa.selenium.*;
import org.openqa.selenium.remote.Augmenter;
import ru.yandex.qatools.ashot.AShot;
import ru.yandex.qatools.ashot.Screenshot;
import javax.imageio.ImageIO;
import java.awt.image.BufferedImage;
import java.io.IOException;
import static org.junit.Assert.assertEquals;

public class ThirdTest extends WebDriverSettings {

    @Attachment(value = "{name}", type = "image/png")
    public static byte[] getFullScreenshot(WebDriver driver, String name)
    throws IOException {
        WebDriver augmented = new Augmenter().augment(driver);
        Screenshot shot = new AShot().takeScreenshot(augmented);
        BufferedImage img = shot.getImage();
        ByteArrayOutputStream baos = new ByteArrayOutputStream();
        ImageIO.write(img, "png", baos);
        baos.flush();
        byte[] imageInByte = baos.toByteArray();
        baos.close();
        return imageInByte;
    }

    @Rule
    public TestWatcher screenshotOnFailure = new TestWatcher() {
        @Override
        protected void failed(Throwable e, Description description) {
            try {
                getFullScreenshot(driver, "Скриншот: ошибка");
            } catch (IOException f) { }
        }
    };

    @Step("Заходим на сайт Ozon")
    public static void IntoOzon(WebDriver driver) throws IOException
    {
        driver.get("https://www.ozon.ru/");
        assertEquals("OZON – интернет-магазин. Миллионы товаров по выгодным ценам", driver.getTitle());

        getFullScreenshot(driver, "скрин: Главная страница");
    }

    /*@Step("перейти на страницу с регистрацией")
    public static void check_in(WebDriver driver) throws IOException
    {
        WebElement myElement =
        driver.findElement(By.xpath("//*[@id=\"__layout\"]/div[1]/div[1]/header[1
```



```

]/div[2]/div[3]/div[1]/div[1]"));
        Actions builder = new Actions(driver);
        builder.moveToElement(myElement).build().perform();

driver.findElement(By.xpath("//*[@id=\"__layout\"]/div[1]/div[1]/header[1]
]/div[2]/div[3]/div[1]/div[1]/div[2]/div[2]/span[1]")).click();
    }

    @Step("Создать аккаунта как юрлицо")
    public static void entity(WebDriver driver) throws IOException
    {
        ForThirdTest entity = new ForThirdTest(driver);
        entity.entity();
    }

    @Step("Ввод имени и фамилии")
    public static void FI(WebDriver driver) throws IOException
    {
        ForThirdTest FI = new ForThirdTest(driver);
        FI.FI("Иван Иванович");
    }

    @Step("Ввод некорректной почты")
    public static void email(WebDriver driver) throws IOException
    {
        ForThirdTest email = new ForThirdTest(driver);
        email.email("htrhtrh");
    }

    @Step("Ввод некорректных символов")
    public static void InvalidCharacters (WebDriver driver) throws
IOException
    {
        Search InvalidCharacters = new Search(driver);
        InvalidCharacters.Search("ase6+2le");

        JavascriptExecutor jse = ((JavascriptExecutor) driver);
        getFullScreenshot(driver, "Скриншот страницы после ввод
некорректных символов");
    }

    @Step("Ввод коректных символов")
    public static void Correctly (WebDriver driver) throws IOException
    {
        Search correctly = new Search(driver);
        correctly.Search("Телефон ALCATEL T56 white");

        getFullScreenshot(driver, "Скриншот страницы после ввод коректных
символов");
    }

    @Step("переход по товару 'ALCATEL T56 white'")
    public static void Transition(WebDriver driver) throws IOException
    {
        JavascriptExecutor jse = ((JavascriptExecutor) driver);
        jse.executeScript("scroll(0, 250);");
        ForThirdTest transition = new ForThirdTest(driver);
        transition.transition();

        getFullScreenshot(driver, "Скриншот страницы с товаром 'ALCATEL
T56 white'");
    }

    @Step("добавить в избранное")

```

```

public static void Favorites(WebDriver driver) throws IOException
{
    ForThirdTest favorites = new ForThirdTest(driver);
    favorites.favorites();

    getFullScreenshot(driver, "Скриншот добавить в избранное");
}

@Step("переход 'в избранное'")
public static void ToFavorites(WebDriver driver) throws IOException
{
    ForThirdTest toFavorites = new ForThirdTest(driver);
    toFavorites.toFavorites();

    JavascriptExecutor jse = ((JavascriptExecutor) driver);
    jse.executeScript("scroll(0, 300);");
    getFullScreenshot(driver, "Скриншот страницы избранное");
}

@Step("Выход из браузера")
public static void WaitAndQuit(WebDriver driver) throws
InterruptedException
{
    Thread.sleep(1000);
    driver.close();
    driver.quit();
    System.out.println("test close");
}

@Test
@DisplayName("тест для проверки: ввода некорректных символов, ввод
корректных символов, добавление в избранное")
public void Test3() throws InterruptedException, IOException
{
    IntoOzon(driver);
    InvalidCharacters(driver);
    Correctly(driver);
    Transition(driver);
    Favorites(driver);
    ToFavorites(driver);
    WaitAndQuit(driver);
}
}

```

Листинг 11 - листинг файла ThirdTest.java

```

package ForTest;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.WebDriverWait;

public class ForThirdTest {
    private static String OZON_URL = "https://www.ozon.ru";

    WebDriverWait waitFor;

    @FindBy(xpath = "//*[@id=\"name\"]")
    private WebElement transition;

    @FindBy(xpath =
"//*[@id=\"__layout\"]/div[1]/div[1]/div[4]/div[1]/div[2]/div[3]/div[1]/d

```

```

iv[2]/div[1]/button[1]")
    private WebElement favorites;

    @FindBy(xpath =
"//*[@id="__layout"]/div[1]/div[1]/header[1]/div[2]/div[3]/div[3]/a[1]/
span[1]")
    private WebElement toFavorites;

    /*@FindBy(xpath =
"//*[@id="__layout"]/div[1]/div[1]/header[1]/div[2]/div[3]/div[1]/div[2
]/div[1]/div[1]/div[1]/form[1]/div[1]/div[3]/a[2]")
    private WebElement entity;*/

    /*@FindBy(xpath =
"//*[@id="__layout"]/div[1]/div[1]/header[1]/div[2]/div[3]/div[1]/div[2
]/div[1]/div[1]/div[1]/form[1]/div[1]/div[2]/input[1]")
    private WebElement FI;

    @FindBy(xpath =
"//*[@id="__layout"]/div[1]/div[1]/header[1]/div[2]/div[3]/div[1]/div[2
]/div[1]/div[1]/div[1]/form[1]/div[1]/div[3]")
    private WebElement email;

    @FindBy(xpath =
"//*[@id="__layout"]/div[1]/div[1]/header[1]/div[2]/div[3]/div[1]/div[2
]/div[1]/div[1]/div[1]/form[1]/div[1]/div[4]")
    private WebElement password;*/

    public ForThirdTest(WebDriver driver) {
        if (!driver.getCurrentUrl().contains(OZON_URL)) {
            throw new IllegalStateException("This is not the page you are
expected");
        }
        PageFactory.initElements(driver, this);
        this.waitFor = new WebDriverWait(driver, 3, 300);
    }

    public void transition(){transition.click();}
    public void favorites(){favorites.click();}
    public void toFavorites(){toFavorites.click();}
}

```

Листинг 12 - листинг файла ForThirdTest.java

#### 6.4 Общее для трех сценариев тестирования

```

package ForTest;

import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.FindBy;
import org.openqa.selenium.support.PageFactory;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Search {
    private static String OZON_URL = "https://www.ozon.ru";

    WebDriverWait waitFor;

    @FindBy(xpath =
"//*[@id="__layout"]/div[1]/div[1]/header[1]/div[2]/div[2]/div[1]/div[1
]/div[2]/div[1]/div[1]/div[1]/div[1]/input[1]")
    private WebElement searchBox;

```

```

    public Search(WebDriver driver) {
        if (!driver.getCurrentUrl().contains(OZON_URL)) {
            throw new IllegalStateException("This is not the page you are
expected");
        }
        PageFactory.initElements(driver, this);
        this.waitFor = new WebDriverWait(driver, 3, 300);
    }

    public void Search(String item) {
        waitFor.until(ExpectedConditions.visibilityOf(searchBox));
        searchBox.clear();
        searchBox.sendKeys(item);
        searchBox.sendKeys(Keys.ENTER);
        searchBox.clear();
    }
}

```

Листинг 13 – листинг файла Search.java