

Харьковский национальный университет им. В.Н. Каразина

Факультет компьютерных наук

Лабораторная работа №8

По учебной дисциплине

**Математические методы и технологии тестирования и верификации  
программного обеспечения**

«Serenity»

Выполнил:

Студент группы КС-22

Филатов Виталий Витальевич

Проверил:

Доц. Малахов Сергей Витальевич

**Тема:** Изучение BDD (behavior driven development) на примере библиотеки Serenity BDD.

**Цель работы:** изучить подход для написания поведенческих тестов с использованием Serenity BDD.

## ВЫПОЛНЕНИЕ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ

```
public class SofasPage {
    private WebDriver webDriver;
    @FindBy(xpath = ".*[@id='producers-filter-block']/div[3]/span[3]/a")
    private WebElement amf;

    public SofasPage(WebDriver webDriver){
        this.webDriver = webDriver;
        PageFactory.initElements(webDriver, this);
    }

    public List<String> getSofasNames(){
        amf.click();
        List<WebElement> sofasNames =
webDriver.findElements(By.xpath(".*li[*]/article/div[2]/div[1]/a"));
        List<String> sofasStringNames = new LinkedList<>();
        for(WebElement sofaName : sofasNames){
            sofasStringNames.add(sofaName.getText());
        }

        return sofasStringNames;
    }
}
```

```
public class HomePage {
    private WebDriver webDriver;
    @FindBy(xpath=".*[@id='column-center']/section/div[5]/ul/li[1]/a")
    private WebElement sofas;

    public HomePage(WebDriver webDriver){
        this.webDriver = webDriver;
        PageFactory.initElements(webDriver, this);
    }

    public SofasPage choiceSofas(){
        sofas.click();
        return new SofasPage(webDriver);
    }
}
```

```
public class StepsForSerenity {
    private WebDriver driver;
    private HomePage homePage;
    private SofasPage sofasPage;
    private String pageSite = "https://pn.com.ua/";

    @Step ("The user visit a page {0}")
```

```

public void a_user_visit_a_page (String page){
    System.setProperty("webdriver.chrome.driver", "chromedriver.exe");
    driver = new ChromeDriver();
    homePage = new HomePage(driver);
    driver.get(pageSite);
}

@Step ("When the user chooses category Sofas {0}")
public void when_user_chooses_sofas_category(){
    sofasPage = homePage.choiceSofas();
}

@Step ("Producers of current goods equals to {0}")
public void producer_of_goods_equals_to_selected_producer(String producer){
    List<String> sofasNames = sofasPage.getSofasNames();
    assertThat(sofasNames).areExactly(sofasNames.size(), new
Condition<String>() {
        @Override
        public boolean matches(String s) {
            return s.startsWith(producer);
        }
    });

    driver.quit();
}

@RunWith(SerenityRunner.class)
public class Tests{
    @Steps
    private StepsForSerenity stepsForSerenity;

    @Test
    public void test() {
        stepsForSerenity.a_user_visit_a_page("https://pn.com.ua");
        stepsForSerenity.when_user_chooses_sofas_category();
        stepsForSerenity.producer_of_goods_equals_to_selected_producer("AMF");
    }
}

```

▼	✓ Tests (com.filatov)	17 s 784 ms
	✓ test	17 s 784 ms

Рисунок 1 – Успешное прохождение теста

Test		
Steps	Outcome	Duration
✓ The user visit a page https://pn.com.ua	SUCCESS	7,94s
✓ When the user chooses category Sofas	SUCCESS	3,94s
✓ Producers of current goods equals to AMF	SUCCESS	5,05s
	SUCCESS	17,5s

Рисунок 2 – Отчет Serenity о прохождении теста

✓ Test (com.filatov)	13 s 17 ms
✓ verifySubCategory	13 s 17 ms

Рисунок 3 – Успешное прохождение теста проекта, предложенного в работе

Verify Sub Category		
Steps	Outcome	Duration
✓ Given the user visits a page https://pn.com.ua	SUCCESS	0,15s
✓ When the user chooses Computer	SUCCESS	10,6s
✓ The user sees IT услуги subcategory	SUCCESS	1,69s
	SUCCESS	12,8s

Рисунок 4 – Отчет об успешном прохождении теста

## ВЫВОДЫ

В ходе данной работы я изучил подход для написания поведенческих тестов с использованием Serenity BDD. В основе лежит формула «given-when-then». Given (“имея, предполагая, при условии”) описывает начальную ситуацию, в которой находится пользователь в контексте работы с приложением. When (“когда”) описывает набор действий пользователя в данной ситуации. Then (“тогда”) описывает ожидаемое поведение приложения.