

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. Н. КАРАЗІНА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК
КАФЕДРА БЕЗПЕКИ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ

Лабораторна робота №8
з навчальної дисципліни
**«Математичні методи та технології тестування та верифікації
програмного забезпечення»**

Виконала:

студентка групи КС-23

Рузудженк С.Р.

Перевірив:

доцент

Нарєжній О. П.

Лабораторна робота №7
з навчальної дисципліни
«Математичні методи та технології тестування та верифікації
програмного забезпечення»

Тема: «Шаблон проектування Page Object»

Мета: вивчення BDD (behavior driven development) на прикладі бібліотеки Serenity BDD.

Хід роботи

Запускаємо *IntelliJIDEA*, створюємо проект *Maven*. Додаємо залежності до файлу *pom.xml*.

```
<dependencies>
  <dependency>
    <groupId>net.serenity-bdd</groupId>
    <artifactId>core</artifactId>
    <version>${serenity.version}</version>
  </dependency>
  <dependency>
    <groupId>net.serenity-bdd</groupId>
    <artifactId>serenity-junit</artifactId>
    <version>${serenity.version}</version>
  </dependency>
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-simple</artifactId>
    <version>1.6.1</version>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.assertj</groupId>
    <artifactId>assertj-core</artifactId>
    <version>1.7.0</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>junit</groupId>
```

```

        <artifactId>junit</artifactId>
        <version>RELEASE</version>
    </dependency>
</dependencies>

```

Формуємо проект з тестами: в першому тесті заходимо на сторінку <https://pn.com.ua/> та перевіряємо наявність елемента «ІТ услуги» на сторінці; у другому – заходимо на сторінку <https://pn.com.ua/> , обираємо категорію «Диваны», першого популярного виробника і перевіряємо, що всі відфільтровані товари належать даному виробнику. Для цього необхідно створити наступні класи:

- **HomePage** для керування діями у браузері;

```

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

```

Вказуємо поля класу:

```

private WebDriver driver;
@FindBy(
    xpath = ".*//*[@id='column-center']/section/div[1]/a")
private WebElement computerCategory;

@FindBy(
    xpath = ".*//*[@id='column-center']/section/div[5]/ul/li[1]/a")
private WebElement sofasCategory;

@FindBy(
    xpath = ".*//*[@id='producers-filter-block']/div[3]/span[3]/a")
private WebElement popularBrand;

@FindBy(
    xpath = ".*//*[@class = 'td-logo-block']/a/span[1]")
private WebElement home;

```

Оголосити елемент за допомогою анотації @FindBy. Пошук елементів відбувається шляхом використання методу *driver.findElement*.

Створюємо конструктор з параметром *driver*:

```

public HomePage(WebDriver driver) {
    this.driver = driver;
    PageFactory.initElements(driver, page: this);
}

```

Створюємо необхідні для коректної роботи методи:

```
public ComputerPage chooseComputerCategory(){
    this.computerCategory.click();
    return new ComputerPage(this.driver);
}

public ComputerPage chooseSofasCategory(){
    this.sofasCategory.click();
    return new ComputerPage(this.driver);
}

public ComputerPage choosePopularBrand(){
    this.popularBrand.click();
    return new ComputerPage(this.driver);
}
```

- *ComputerPage*, в якому, власне, реалізуємо тести;

Вказуємо поля класу:

```
private WebDriver driver;
@FindBy(
    xpath = ".*[@id='column-center']/section/div[4]/article[3]/div/div")
private WebElement ITService;
@FindBy(
    xpath = ".*[@class = 'td-logo-block']/a/span[1]")
private WebElement home;

@FindBy(xpath = ".*[@id='column-center']/" +
    "section/div[3]/*/li/*/div[2]/div[1]/*[contains(text(), 'AMF ')]")
private List<WebElement> list;

private ArrayList<String> stringList;
```

Конструктор:

```
public ComputerPage(WebDriver driver) {
    this.driver = driver;
    PageFactory.initElements(driver, this);
}
```

Методи:

```
public String getTextITService() {
    return this.ITService.getText();
}

public ArrayList getElements(){
    stringList = new ArrayList<String>();
    for(int i = 0; i < this.list.size(); ++i) {
        this.stringList.add((this.list.get(i)).getText());
    }
    return stringList;
}

public boolean toCompare(){
    boolean condition = true;
    for(int i = 0; i < this.list.size(); ++i) {
        System.out.println(this.stringList.get(i));
        condition = ((this.stringList.get(i)).equals((this.list.get(i)).getText()));
    }
    return condition;
}
```

- *StepsForSerenity*, який містить кроки проходження кожного тесту;

```
import Pages.ComputerPage;
import Pages.HomePage;
import net.thucydides.core.annotations.Step;
import org.junit.Assert;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
```

Поля класу:

```
public HomePage homePage;
public WebDriver driver;
public ComputerPage computerPage;
public String siteHomePage = "https://pn.com.ua/";
```

Кроки першого тесту:

```
@Step("Given the user visits the page {0}")
public void user_visits_page(String homePage){
    this.siteHomePage = homePage;
}

@Step("When the user chooses the computer category {0}")
public void user_chooses_computerCategory(){
    System.setProperty("webdriver.chrome.driver",
        "C:\\Users\\Anna\\Desktop\\it\\chromedriver.exe");
    WebDriver driver = new ChromeDriver();
    homePage = new HomePage(driver);
    driver.get(siteHomePage);
    computerPage = homePage.chooseComputerCategory();
}

@Step("When the user sees {0} subCategory")
public void user_sees_subCategory_ITservice(String subCategory){
    String string = computerPage.getTextITService().substring( beginIndex: 0, endIndex: 21);
    Assert.assertEquals( message: "Test ComputerCategory failed",
        computerPage.getTextITService(), string);
    driver.quit();
}
```

Кроки другого тесту:

```
@Step("User chooses sofas and popularBrand categories")
public void user_chooses_sofasCategory_and_the_brandCategory(){
    System.setProperty("webdriver.chrome.driver",
        "C:\\Users\\Anna\\Desktop\\it\\chromedriver.exe");
    this.driver = new ChromeDriver();
    homePage = new HomePage(driver);
    driver.get(siteHomePage);
    computerPage = homePage.chooseSofasCategory();
    computerPage = homePage.choosePopularBrand();
}

@Step("Comparing the elements for checking if all elements belongs to AMF")
public void user_sees_elements_of_AMFbrand(){
    computerPage.getElements();
    Assert.assertTrue( message: "Something have gone wrong!",
        this.computerPage.toCompare());
}
```

- *TestSerenity*, у якому містяться тести;

```
import net.serenitybdd.junit.runners.SerenityRunner;
import net.thucydides.core.annotations.Steps;
import org.junit.Test;
import org.junit.runner.RunWith;
import org.openqa.selenium.WebDriver;
```

```

@RunWith(SerenityRunner.class)
public class TestSerenity {

    WebDriver driver;
    String siteHomePage = "https://pn.com.ua/";

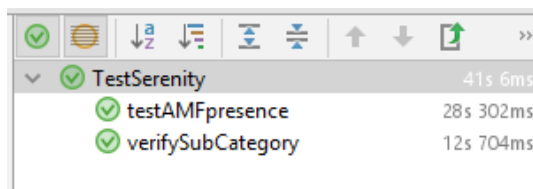
    @Steps
    StepsForSerenity stepsForSerenity;

    @Test
    public void verifySubCategory() {
        stepsForSerenity.user_visits_page(siteHomePage);
        stepsForSerenity.user_chooses_computerCategory();
        stepsForSerenity.user_sees_subCategory_ITservice("ІТ услуги");
    }

    @Test
    public void testAMFpresence() {
        stepsForSerenity.user_visits_page(siteHomePage);
        stepsForSerenity.user_chooses_sofasCategory_and_the_brandCategory();
        stepsForSerenity.user_sees_elements_of_AMFbrand();
    }
}

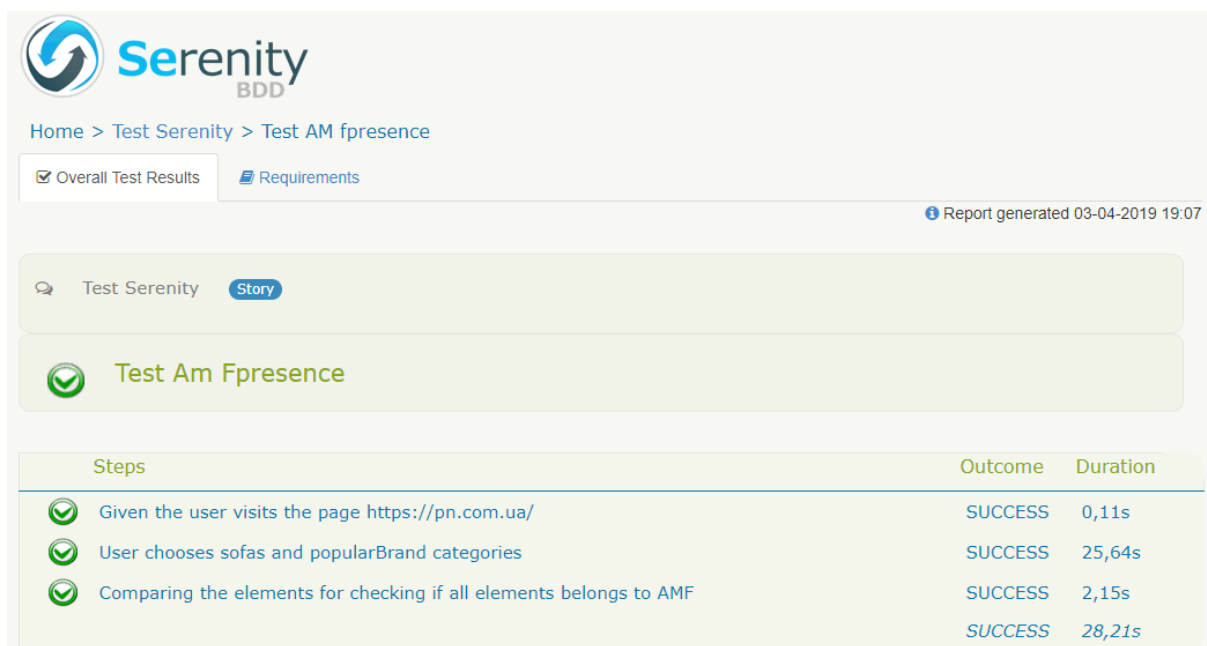
```

Можемо побачити, що всі тести пройшли успішно:



Test Name	Duration
TestSerenity	41s 6ms
testAMFpresence	28s 302ms
verifySubCategory	12s 704ms

Формуємо репорти тестів:



Serenity BDD

Home > Test Serenity > Test AM fpresence

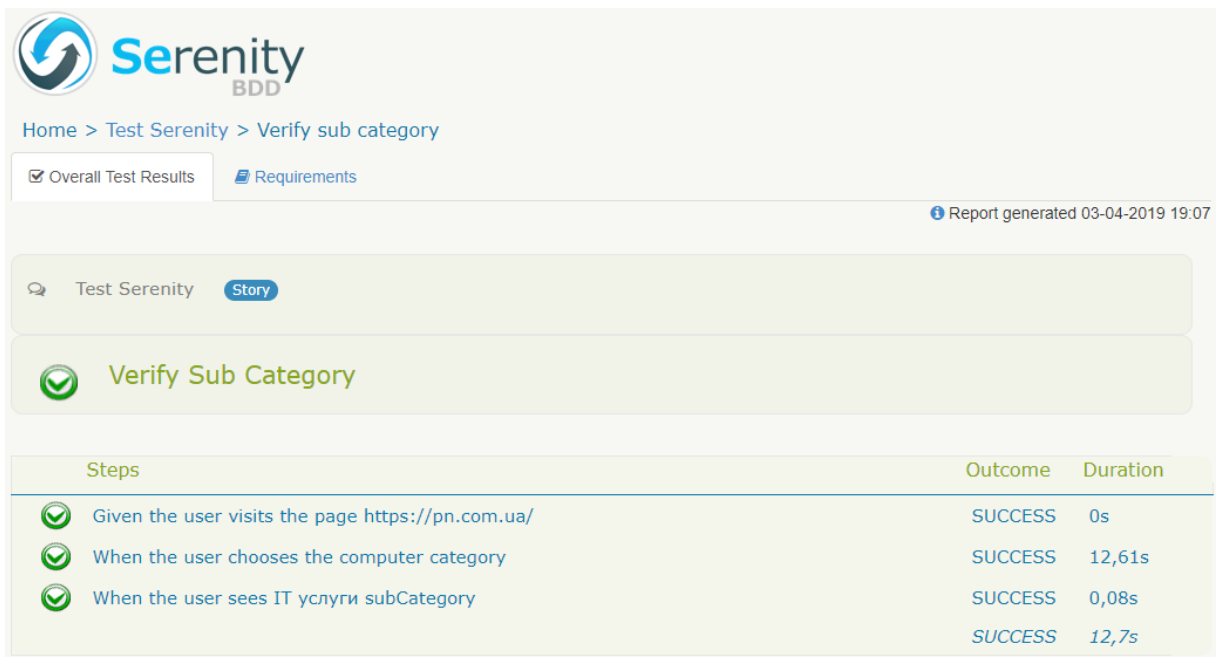
☒ Overall Test Results ☐ Requirements

Report generated 03-04-2019 19:07

Test Serenity **Story**

Test Am Fpresence

Steps	Outcome	Duration
Given the user visits the page https://pn.com.ua/	SUCCESS	0,11s
User chooses sofas and popularBrand categories	SUCCESS	25,64s
Comparing the elements for checking if all elements belongs to AMF	SUCCESS	2,15s
	SUCCESS	28,21s



The screenshot displays the Serenity BDD web interface. At the top, the Serenity BDD logo is visible. Below it, the breadcrumb navigation shows 'Home > Test Serenity > Verify sub category'. There are two tabs: 'Overall Test Results' (selected) and 'Requirements'. A timestamp indicates the report was generated on 03-04-2019 at 19:07. The main section shows a 'Test Serenity' button and a 'Story' button. Below this, a green checkmark icon is next to the title 'Verify Sub Category'. A table follows, detailing the test steps, their outcomes, and durations.

Steps	Outcome	Duration
Given the user visits the page https://pn.com.ua/	SUCCESS	0s
When the user chooses the computer category	SUCCESS	12,61s
When the user sees IT услуги subCategory	SUCCESS	0,08s
	SUCCESS	12,7s

Висновки

Отже, **Serenity** –це фреймворк з відкритим вихідним кодом для написання функціональних тестів, в який з коробки вбудована система генерації звітів про виконанні тестах, що включає такі можливості:

- Інтеграція з іншими популярними фреймворками за допомогою плагінів (JUnit, Cucumber, JBehave, Rest-Assured, Selenium і інші)
- Безліч вбудованих можливостей (паралельне виконання, збереження скріншотів для UI тестів, Data Driven testing)
- Формування детального звіту
- Інтеграція з системами збірки (maven, gradle, ant)

Таким чином, під час виконання лабораторної роботи, було вивчено підхід до написання поведінкових тестів за допомогою **SerenityBDD**.

У результаті виконання лабораторної роботи було написано два тести, що пройшли успішно. Після виконання тестів було створено репорт про проходження тестів.