Харьковский национальный университет им. В.Н. Каразина

Факультет компьютерных наук

# Лабораторная работа №7

По учебной дисциплине

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

«Шаблон проектирования Page Object»

|  |  |
| --- | --- |
|  | Выполнил:  Студент группы КС-22  Филатов Виталий Витальевич  Проверил:  Доц. Малахов Сергей Витальевич |

**Тема:** шаблон проектирования Page Object.

**Цель работы:** написание тестового сценария с применением шаблона проектирования Page Object.

**ВЫПОЛНЕНИЕ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ**

**public class** HomePage {  
 **private** WebDriver **driver**;  
 @FindBy(xpath = **".//\*[@id=\"column-center\"]/section/div[1]/a"**)  
 **private** WebElement **computerCategory**;  
  
 **public** HomePage(WebDriver driver) {  
 **this**.**driver** = driver;  
 PageFactory.*initElements*(driver, **this**);  
 }  
  
 **public** ComputerPage choiceComputerCategory(){  
 **computerCategory**.click();  
 **return new** ComputerPage(**driver**);  
 }  
}

**public class** ComputerPage {  
 **private** WebDriver **driver**;  
 @FindBy(xpath = **".//\*[@id=\"column-center\"]/section/div[4]/article[3]/div/div"**)  
 **private** WebElement **ITService**;  
  
 **public** ComputerPage(WebDriver driver){  
 **this**.**driver** = driver;  
 PageFactory.*initElements*(driver, **this**);  
 }  
  
 **public** String getTextITService() {  
 **return ITService**.getText();  
 }  
}

**public class** TestFromDocument {  
 **private** String **siteHomePage** = **"https://pn.com.ua/"**;  
 **private** WebDriver **driver**;  
 **private** HomePage;  
 **private** ComputerPage;  
  
 @Before  
 **public void** testBeforeClass() {  
 System.*setProperty*(**"webdriver.gecko.driver"**, System.*getProperty*(**"user.dir"**) + **"\\geckodriver.exe"**);  
 **driver** = **new** FirefoxDriver();  
 **driver**.get(**siteHomePage**);  
 }  
  
 @Test  
 **public void** testComputerFactory() {  
 **homePage** = PageFactory.*initElements*(**driver**, HomePage.**class**);  
 **computerPage** = **homePage**.choiceComputerCategory();  
 System.***out***.println(**computerPage**.getTextITService());  
 **computerPage**.getTextITService();  
 **driver**.manage().timeouts().implicitlyWait(2, TimeUnit.***MINUTES***);  
 *assertEquals*(**true**, **computerPage**.getTextITService().contains(**"IT услуги"**));  
 }  
}

**public class** HomePage {  
 **private** 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** SofasPage {  
 **private** 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** Tests {  
 **static** WebDriver;  
  
 @BeforeClass  
 **public static void** webDriverInitialisation() {  
 System.*setProperty*(**"webdriver.gecko.driver"**, System.*getProperty*(**"user.dir"**) + **"\\geckodriver.exe"**);  
 *webDriver* = **new** FirefoxDriver();  
 *webDriver*.get(**"https://pn.com.ua"**);  
 }

@Test  
 **public void** test(){  
 HomePage homePage = **new** HomePage(*webDriver*);  
 List<String> sofasNames = homePage.choiceSofas().getSofasNames();  
 **for**(String sofaName : sofasNames){  
 Assert.*assertEquals*(**true**, sofaName.startsWith(**"AMF"**));  
 }  
 }  
}



Рисунок 1 – успешное прохождение теста



Рисунок 2 – успешное завершение теста из лабораторной работы

**ВЫВОДЫ**

В ходе данной лабораторной работы я ознакомился с шаблоном Page Object и использовал его для написания теста. Page Object как бы моделирует страницы тестируемого приложения в качестве объектов в коде. В результате его использования получаются отдельные классы, отвечающие за работу с HTML каждой конкретной веб-страницы. Такой подход значительно уменьшает объем повторяющегося кода, потому что одни и те же объекты страниц можно использовать в различных тестах.