**Practical no 3**

**AIM:** Write and test a program to login a specific web page using JUnit.

**Theory**

**What is Junit**

The JUnit Platform serves as a foundation for launching testing frameworks on the JVM.

It also defines the TestEngine API for developing a testing framework that runs on the

platform. Furthermore, the platform provides a Console Launcher to launch the platform

from the command line and a JUnit 4 based Runner for running any TestEngine on the

platform in a JUnit 4 based environment. First-class support for the JUnit Platform also

exists in popular IDEs like eclipse.

**Annotations Used in Junit**

**@Test** Denotes that a method is a test method. Unlike JUnit 4’s @Test annotation, this

annotation does not declare any attributes, since test extensions in JUnit Jupiter

operate based on their own dedicated annotations. Such methods are inherited

unless they are overridden.

**@BeforeAll** Denotes that the annotated method should be executed before all @Test,

@RepeatedTest, @ParameterizedTest, and @TestFactory methods in the current

class; analogous to JUnit 4’s @BeforeClass.

**@AfterAll** Denotes that the annotated method should be executed after all @Test,

@RepeatedTest, @ParameterizedTest, and @TestFactory methods in the current

class; analogous to JUnit 4’s @AfterClass.

**@BeforeEach** Denotes that the annotated method should be executed before each

@Test, @RepeatedTest, @ParameterizedTest, or @TestFactory method in the current class; analogous to JUnit 4’s @Before. Such methods are inherited unless they

are overridden.

**@AfterEach** Denotes that the annotated method should be executed after each @Test,

@RepeatedTest, @ParameterizedTest, or @TestFactory method in the current class;

analogous to JUnit 4’s @After. Such methods are inherited unless they are overridden.

**Steps for Testing a site using Junit**

Step-1 :- Create a new Project and give it a name.

Step-2 :- After the initialization create a new package inside the project and give it a

name.

Step-3 :- To include the libraries and ”jar” files, Unzip the zipped archives,next ”right

click on the project icon → BuildP ath → Conf igureBuildP ath..”Step-4 :- Here we will choose ”Add External JARs...” button and navigate to the ”.jar”

file we downloaded/extracted and select them to include in our project.

Step-5 :- Save the configuration. You have successfully installed selenium for java.

Step-6 :- For linking the chrome driver you need to include it in the code by adding a

Line System.setProperty("webdriver.chrome.driver","chromedriver path");

Step-7 :- Next create a new JUnit Test Case file give it a name.

Step-8 :- Import the following libraries

import org.junit.Test;

import org.openqa.selenium.By;

import org.openqa.selenium.chrome.ChromeDriver;

import org.openqa.selenium.WebDriver;

import static org.junit.Assert.\*;

import org.junit.Before;

import org.junit.After;

import junit.framework.Assert;

Step-9 :- Add the code.

**Code:-**

|  |
| --- |
| import static org.junit.Assert.\*;  import org.junit.After;  import org.junit.Assert;  import org.junit.Before;  import org.junit.Test;  import org.openqa.selenium.By;  import org.openqa.selenium.WebDriver;  import org.openqa.selenium.chrome.ChromeDriver;  public class JunitEx {  WebDriver driver = null;  @Before  public void setup() {  System.setProperty("webdriver.chrome.driver" , "E:\\tycs\\stqa prac\\prac3\\chromedriver\_win32\\chromedriver.exe" );  driver = new ChromeDriver();  driver.manage().window().maximize();  }    @Test  public void test() throws InterruptedException {    driver.get("http://thedemosite.co.uk/savedata.php");  driver.findElement(By.xpath("//input[@name='username']")).sendKeys("krunal71");  Thread.sleep(1000);  driver.findElement(By.xpath("//input[@name='password']")).sendKeys("kd713");  Thread.sleep(1000);  driver.findElement(By.xpath("//input[@name='FormsButton2']")).click();  Thread.sleep(1000);  Thread.sleep(2000);  Assert.assertTrue("invalid credential" , driver.getTitle().contains("Add a user - FREE PHP code and SQL"));  System.out.println(" page title is verified user is able to login ");  }    @Test  public void demotest() throws InterruptedException {    driver.get("http://demo.guru99.com/test/newtours/");  driver.findElement(By.xpath("//input[@name='userName']")).sendKeys("krunal713");  Thread.sleep(1000);  driver.findElement(By.xpath("//input[@name='password']")).sendKeys("kd713");  Thread.sleep(1000);  driver.findElement(By.xpath("//input[@name='submit']")).click();  Thread.sleep(1000);  Thread.sleep(2000);  Assert.assertTrue("invalid credential" , driver.getTitle().contains("Welcome: Mercury Tours"));  System.out.println(" page title is verified user is able to login ");  }    @After  public void aftertest(){  driver.quit();  }  } |

**Output:**







 

