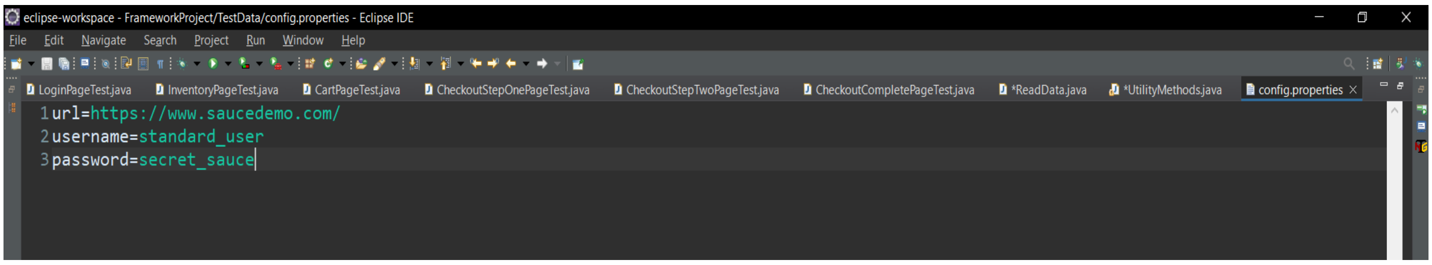
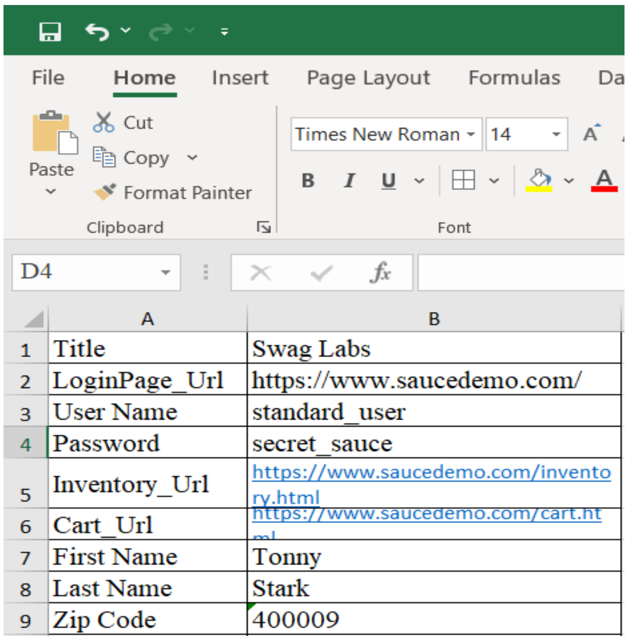
Situ

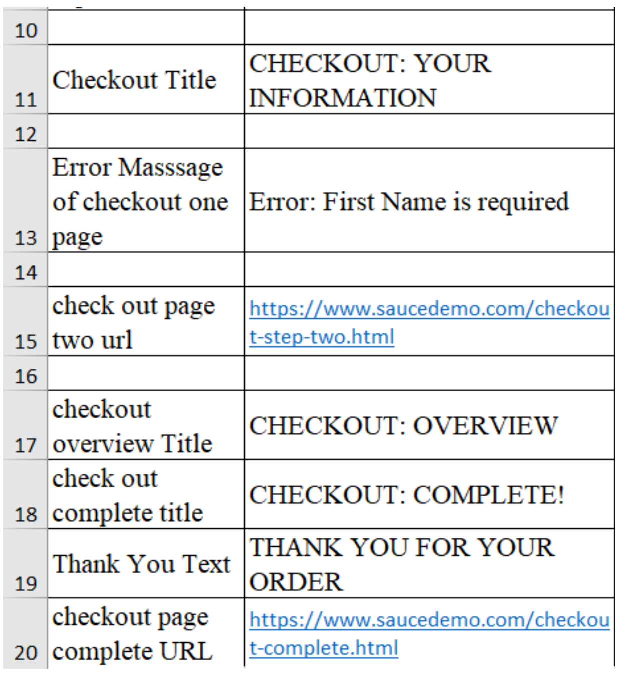
Property File:



```
url=https://www.saucedemo.com/
username=standard_user
password=secret_sauce
```

Excel File:

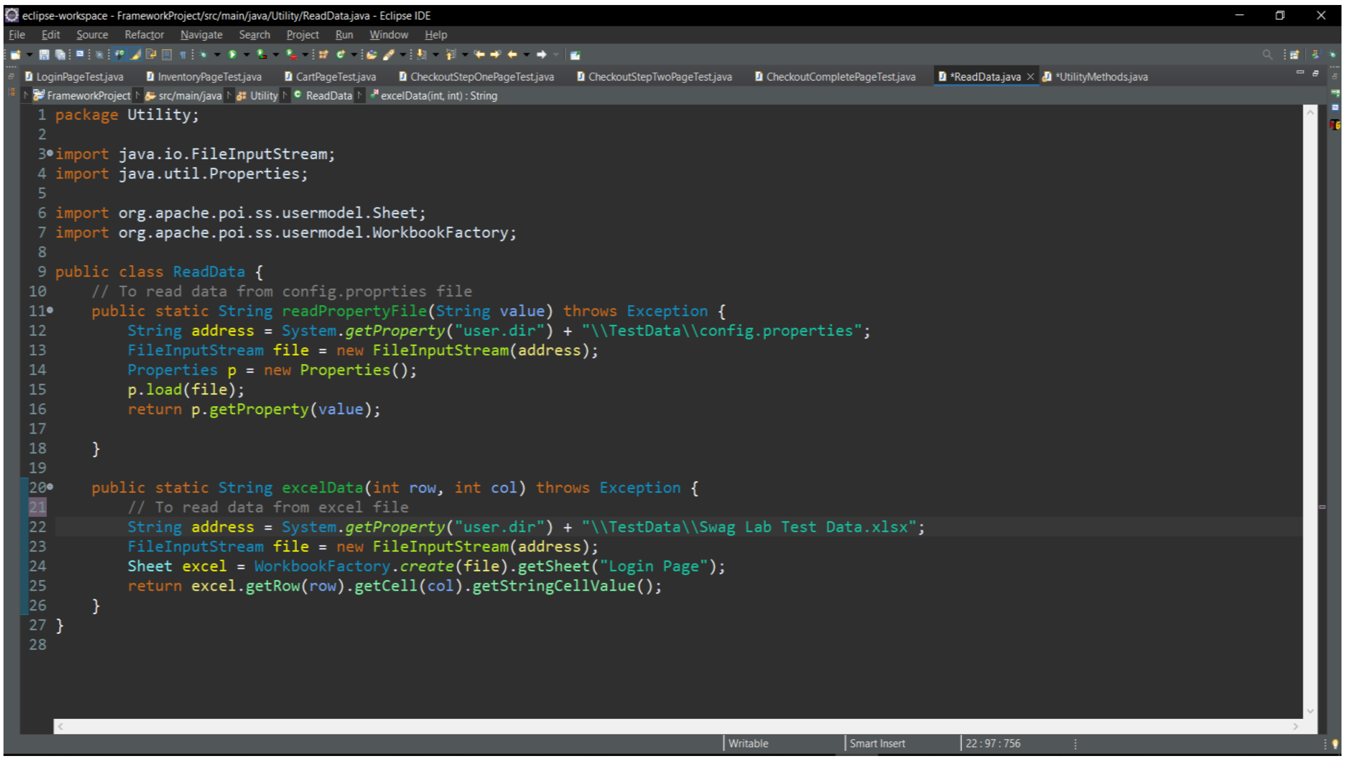




	A	B
1	Title	Swag Labs
2	LoginPage_Url	https://www.saucedemo.com/
3	User Name	standard_user
4	Password	secret_sauce
5	Inventory_Url	https://www.saucedemo.com/inventory.html
6	Cart_Url	https://www.saucedemo.com/cart.html
7	First Name	Tonny
8	Last Name	Stark
9	Zip Code	400009
10		
11	Checkout Title	CHECKOUT: YOUR INFORMATION
12		
13	Error Masssage of checkout one page	Error: First Name is required
14		
15	check out page two url	https://www.saucedemo.com/checkout-step-two.html
16		
17	checkout overview Title	CHECKOUT: OVERVIEW
18	check out complete title	CHECKOUT: COMPLETE!
19	Thank You Text	THANK YOU FOR YOUR ORDER
20	checkout page complete URL	https://www.saucedemo.com/checkout-complete.html

Utility Package:

1.ReadData class:

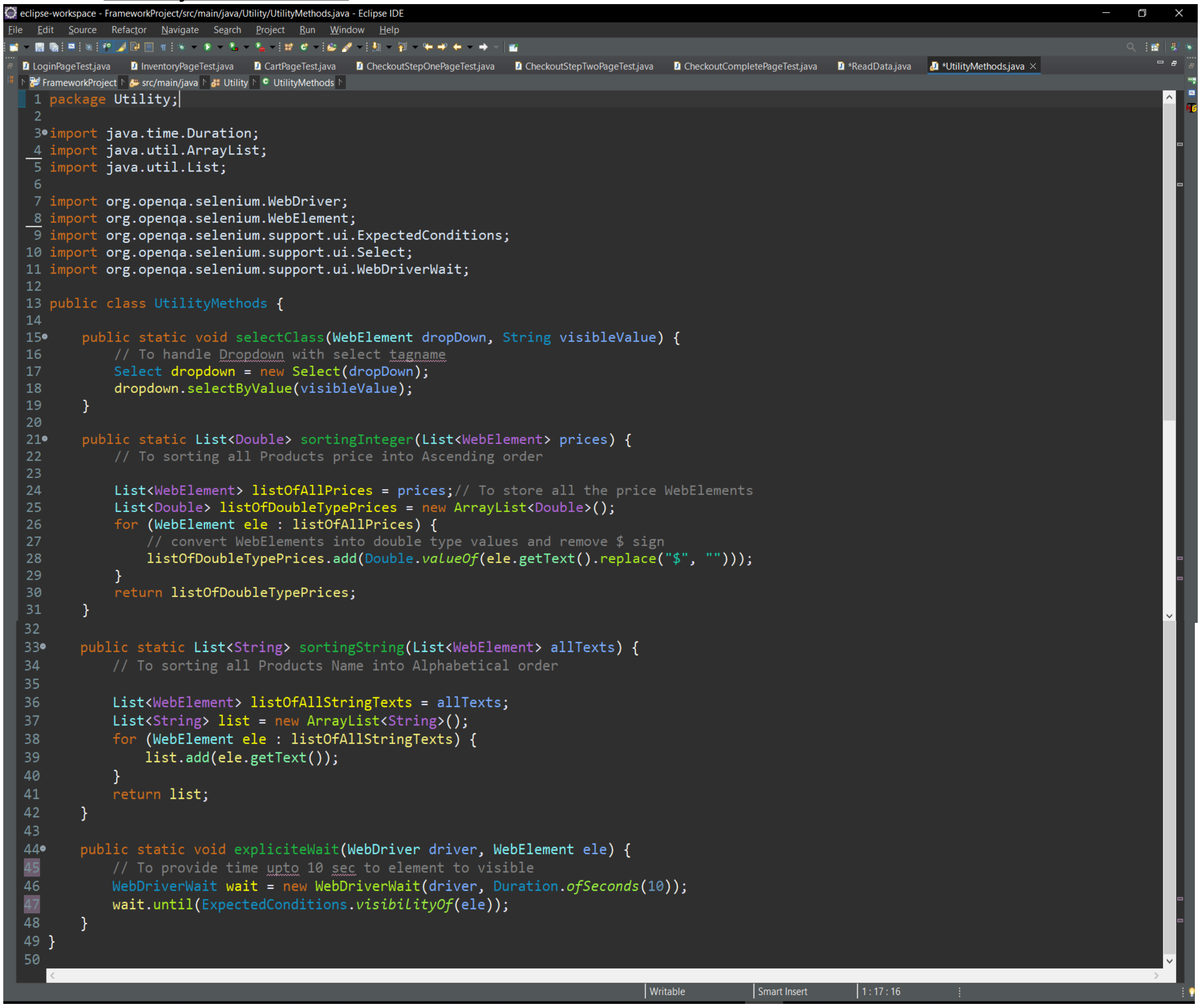


```java
package Utility;

import java.io.FileInputStream;
import java.util.Properties;

import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class ReadData {
    // To read data from config.proprties file
    public static String readPropertyFile(String value) throws Exception {
        String address = System.getProperty("user.dir") + "\\TestData\\config.properties";
        FileInputStream file = new FileInputStream(address);
        Properties p = new Properties();
        p.load(file);
        return p.getProperty(value);

    }

    public static String excelData(int row, int col) throws Exception {
        // To read data from excel file
        String address = System.getProperty("user.dir") + "\\TestData\\Swag Lab Test Data.xlsx";
        FileInputStream file = new FileInputStream(address);
        Sheet excel = WorkbookFactory.create(file).getSheet("Login Page");
        return excel.getRow(row).getCell(col).getStringCellValue();
    }
}
```

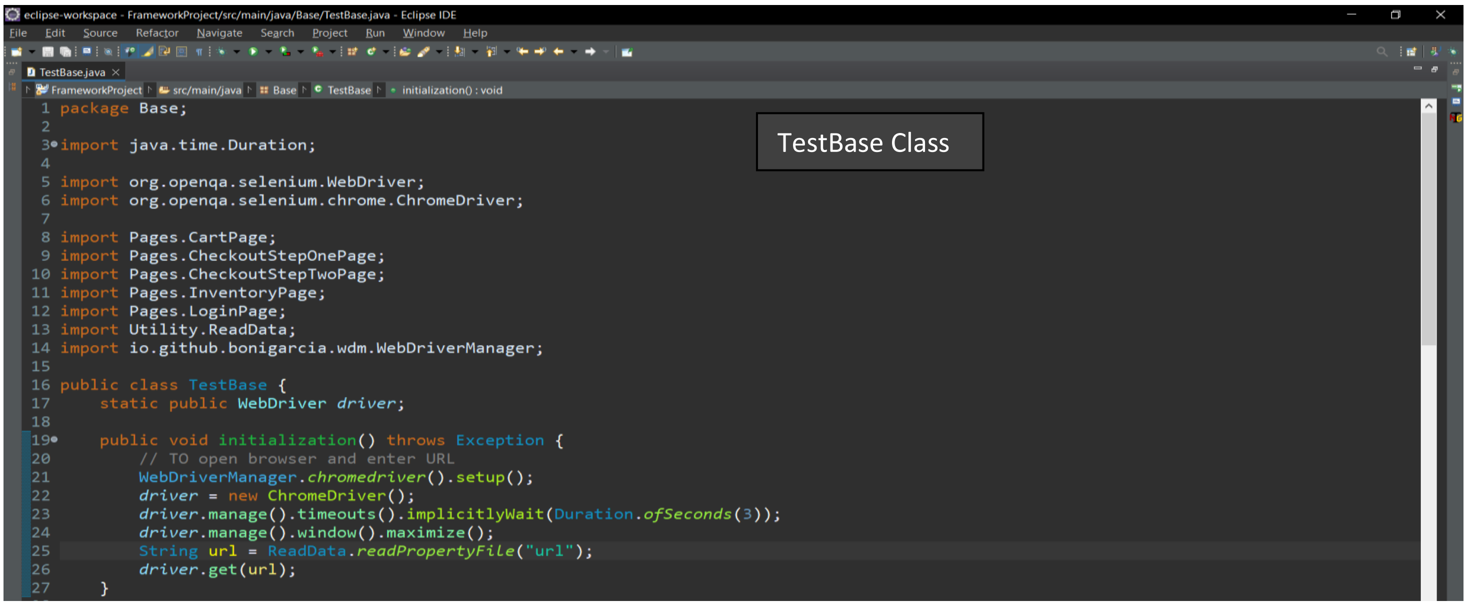
Testers don't make software; they make it better.

Situ

2.UtilityMethods class:


eclipse-workspace - FrameworkProject/src/main/java/Utility/UtilityMethods.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
LoginPageTest.java InventoryPageTest.java CartPageTest.java CheckoutStepOnePageTest.java CheckoutStepTwoPageTest.java CheckoutCompletePageTest.java *ReadData.java *UtilityMethods.java
FrameworkProject > src/main/java > Utility > UtilityMethods >
Writable Smart Insert 1 : 17 : 16

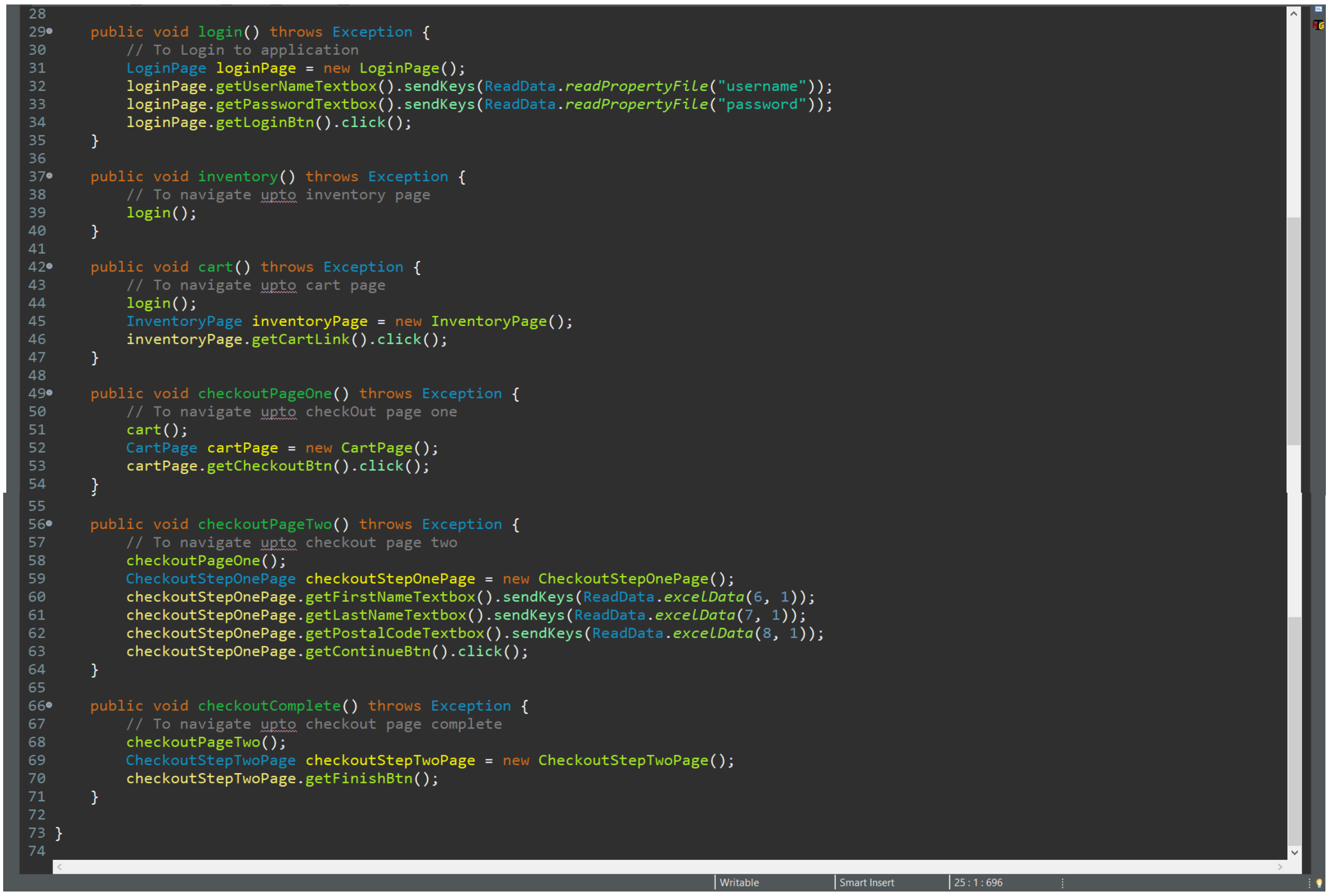
```java
package Utility;

import java.time.Duration;
import java.util.ArrayList;
import java.util.List;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.Select;
import org.openqa.selenium.support.ui.WebDriverWait;

public class UtilityMethods {

    public static void selectClass(WebElement dropDown, String visibleValue) {
        // To handle Dropdown with select tagname
        Select dropdown = new Select(dropDown);
        dropdown.selectByValue(visibleValue);
    }

    public static List<Double> sortingInteger(List<WebElement> prices) {
        // To sorting all Products price into Ascending order

        List<WebElement> listOfAllPrices = prices;// To store all the price WebElements
        List<Double> listOfDoubleTypePrices = new ArrayList<Double>();
        for (WebElement ele : listOfAllPrices) {
            // convert WebElements into double type values and remove $ sign
            listOfDoubleTypePrices.add(Double.valueOf(ele.getText().replace("$", "")));
        }
        return listOfDoubleTypePrices;
    }

    public static List<String> sortingString(List<WebElement> allTexts) {
        // To sorting all Products Name into Alphabetical order

        List<WebElement> listOfAllStringTexts = allTexts;
        List<String> list = new ArrayList<String>();
        for (WebElement ele : listOfAllStringTexts) {
            list.add(ele.getText());
        }
        return list;
    }

    public static void expliciteWait(WebDriver driver, WebElement ele) {
        // To provide time upto 10 sec to element to visible
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));
        wait.until(ExpectedConditions.visibilityOf(ele));
    }
}
```


eclipse-workspace - FrameworkProject/src/main/java/Base/TestBase.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
TestBase.java
FrameworkProject > src/main/java > Base > TestBase > initialization() : void
TestBase Class

```java
package Base;

import java.time.Duration;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

import Pages.CartPage;
import Pages.CheckoutStepOnePage;
import Pages.CheckoutStepTwoPage;
import Pages.InventoryPage;
import Pages.LoginPage;
import Utility.ReadData;
import io.github.bonigarcia.wdm.WebDriverManager;

public class TestBase {
    static public WebDriver driver;

    public void initialization() throws Exception {
        // TO open browser and enter URL
        WebDriverManager.chromedriver().setup();
        driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(3));
        driver.manage().window().maximize();
        String url = ReadData.readPropertyFile("url");
        driver.get(url);
    }
```

Testers don't make software; they make it better.

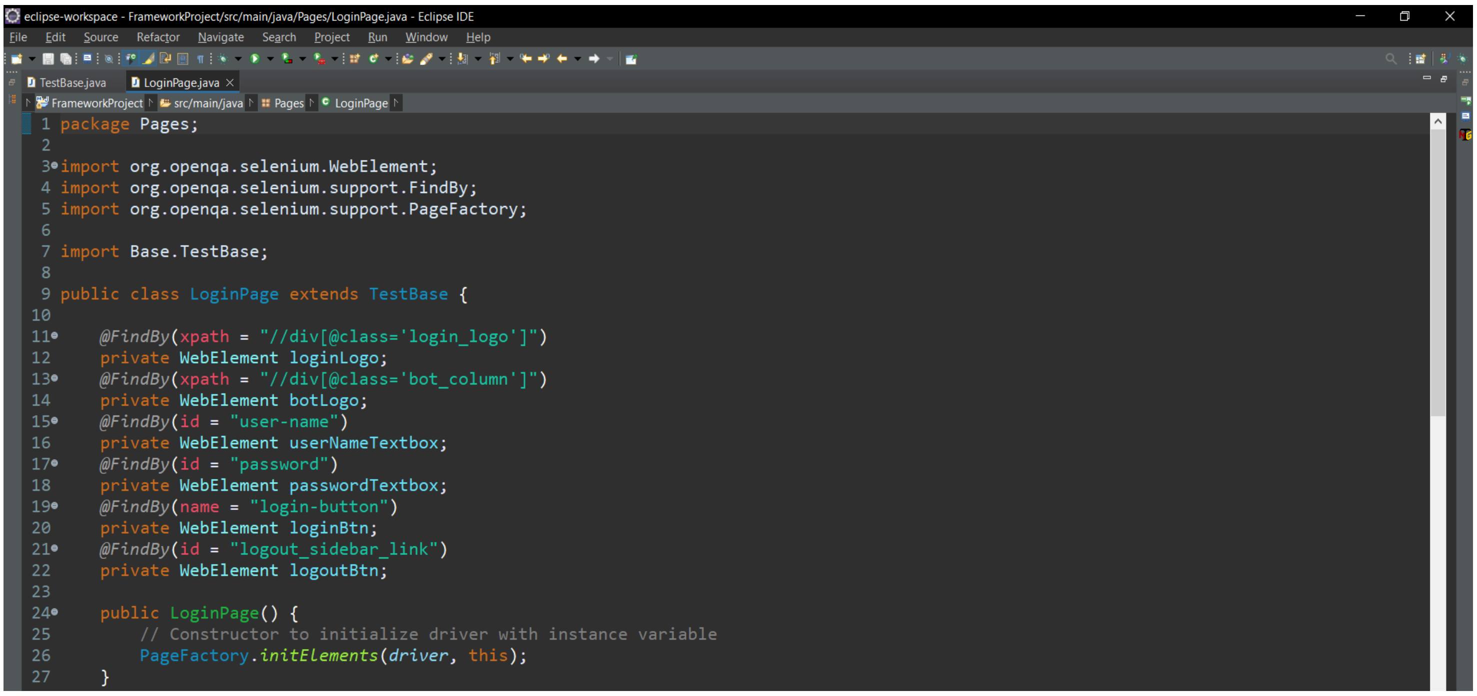
Situ



```java
    public void login() throws Exception {
        // To Login to application
        LoginPage loginPage = new LoginPage();
        loginPage.getUserNameTextbox().sendKeys(ReadData.readPropertyFile("username"));
        loginPage.getPasswordTextbox().sendKeys(ReadData.readPropertyFile("password"));
        loginPage.getLoginBtn().click();
    }

    public void inventory() throws Exception {
        // To navigate upto inventory page
        login();
    }

    public void cart() throws Exception {
        // To navigate upto cart page
        login();
        InventoryPage inventoryPage = new InventoryPage();
        inventoryPage.getCartLink().click();
    }

    public void checkoutPageOne() throws Exception {
        // To navigate upto checkOut page one
        cart();
        CartPage cartPage = new CartPage();
        cartPage.getCheckoutBtn().click();
    }

    public void checkoutPageTwo() throws Exception {
        // To navigate upto checkout page two
        checkoutPageOne();
        CheckoutStepOnePage checkoutStepOnePage = new CheckoutStepOnePage();
        checkoutStepOnePage.getFirstNameTextbox().sendKeys(ReadData.excelData(6, 1));
        checkoutStepOnePage.getLastNameTextbox().sendKeys(ReadData.excelData(7, 1));
        checkoutStepOnePage.getPostalCodeTextbox().sendKeys(ReadData.excelData(8, 1));
        checkoutStepOnePage.getContinueBtn().click();
    }

    public void checkoutComplete() throws Exception {
        // To navigate upto checkout page complete
        checkoutPageTwo();
        CheckoutStepTwoPage checkoutStepTwoPage = new CheckoutStepTwoPage();
        checkoutStepTwoPage.getFinishBtn();
    }

}
```

Writable | Smart Insert | 25 : 1 : 696

<u>LoginPage Class:</u>



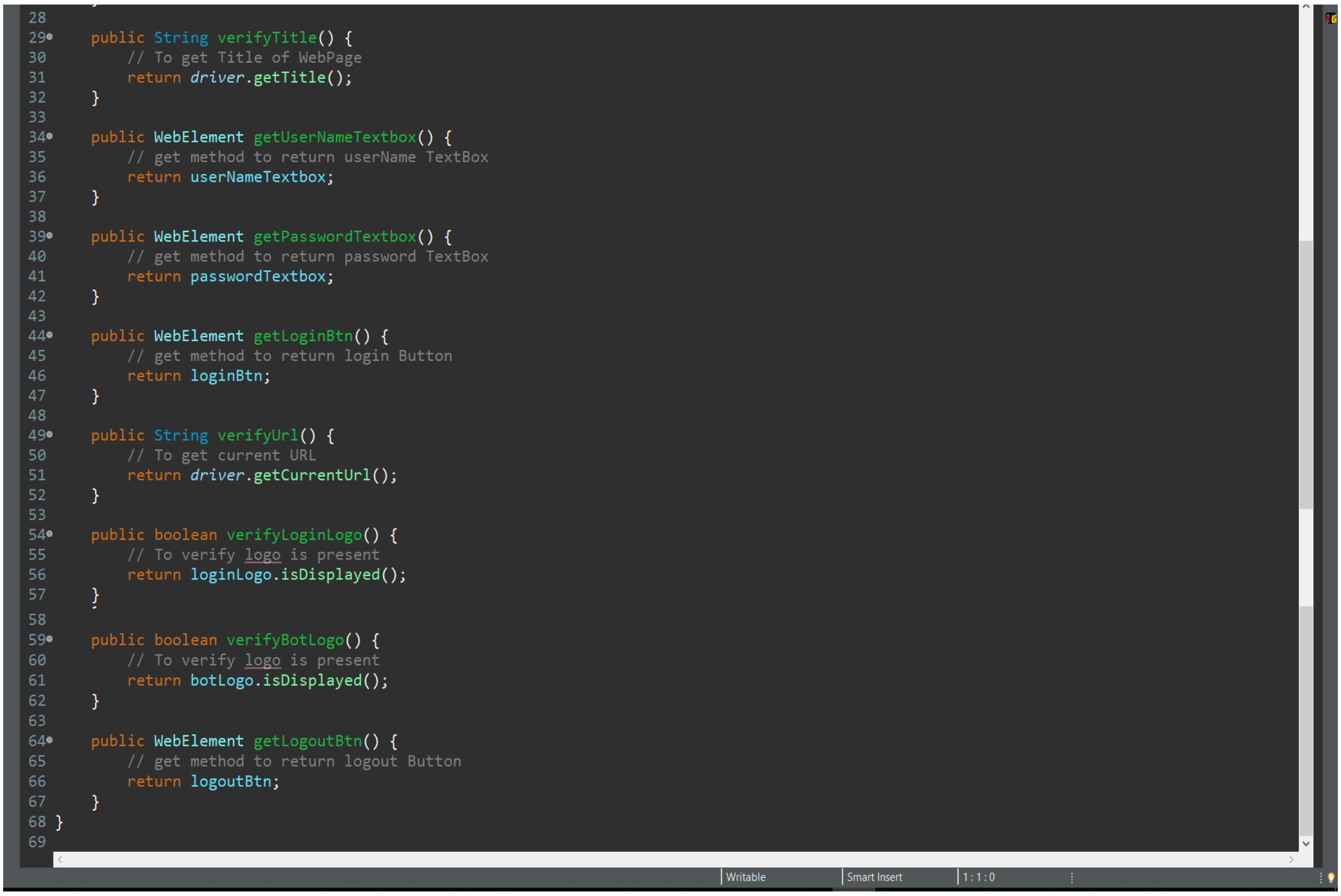
eclipse-workspace - FrameworkProject/src/main/java/Pages/LoginPage.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help
TestBase.java
LoginPage.java
FrameworkProject > src/main/java > Pages > LoginPage >

```java
package Pages;

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

import Base.TestBase;

public class LoginPage extends TestBase {

    @FindBy(xpath = "//div[@class='login_logo']")
    private WebElement loginLogo;
    @FindBy(xpath = "//div[@class='bot_column']")
    private WebElement botLogo;
    @FindBy(id = "user-name")
    private WebElement userNameTextbox;
    @FindBy(id = "password")
    private WebElement passwordTextbox;
    @FindBy(name = "login-button")
    private WebElement loginBtn;
    @FindBy(id = "logout_sidebar_link")
    private WebElement logoutBtn;

    public LoginPage() {
        // Constructor to initialize driver with instance variable
        PageFactory.initElements(driver, this);
    }
```

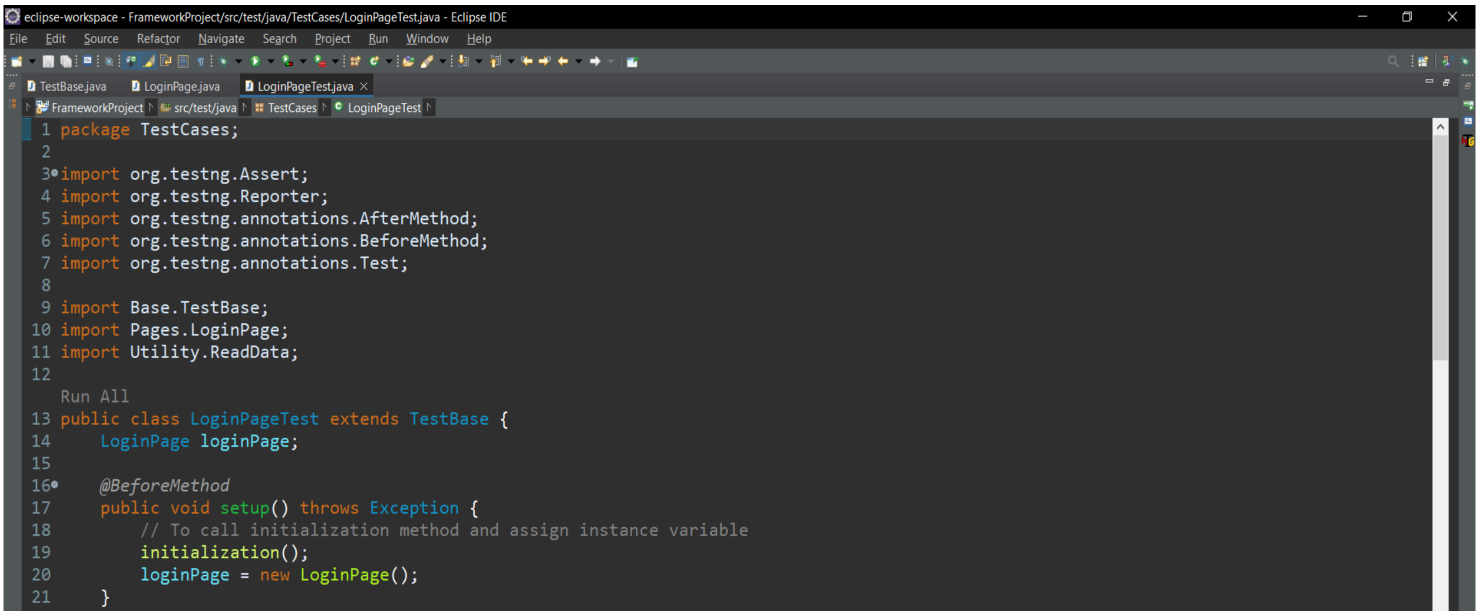
Testers don't make software; they make it better.

Situ



```
    public String verifyTitle() {
        // To get Title of WebPage
        return driver.getTitle();
    }

    public WebElement getUserNameTextbox() {
        // get method to return userName TextBox
        return userNameTextbox;
    }

    public WebElement getPasswordTextbox() {
        // get method to return password TextBox
        return passwordTextbox;
    }

    public WebElement getLoginBtn() {
        // get method to return login Button
        return loginBtn;
    }

    public String verifyUrl() {
        // To get current URL
        return driver.getCurrentUrl();
    }

    public boolean verifyLoginLogo() {
        // To verify logo is present
        return loginLogo.isDisplayed();
    }

    public boolean verifyBotLogo() {
        // To verify logo is present
        return botLogo.isDisplayed();
    }

    public WebElement getLogoutBtn() {
        // get method to return logout Button
        return logoutBtn;
    }
}
```

<u>LoginPageTest Class:</u>

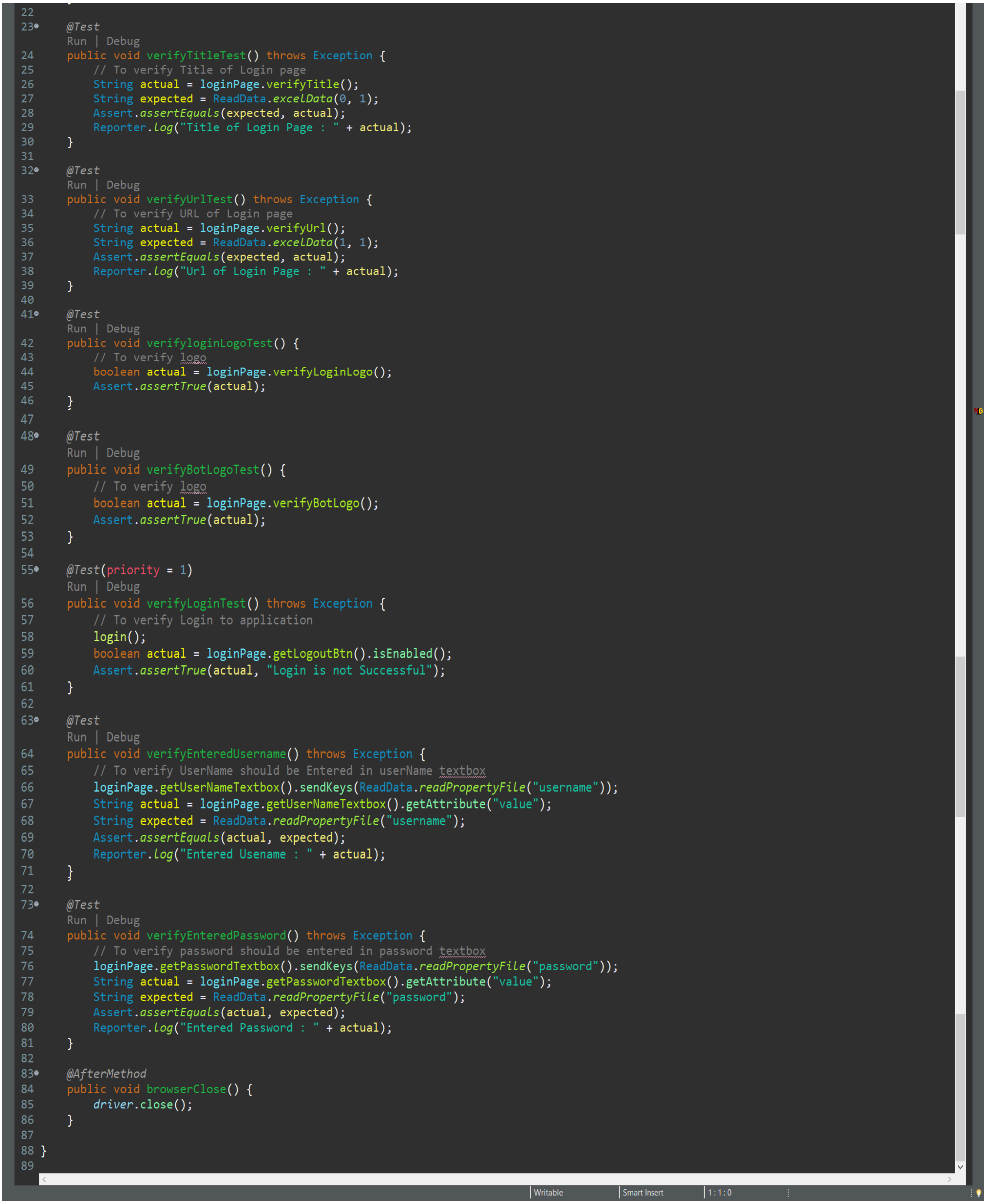


eclipse-workspace - FrameworkProject/src/test/java/TestCases/LoginPageTest.java - Eclipse IDE

```
package TestCases;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.LoginPage;
import Utility.ReadData;

Run All
public class LoginPageTest extends TestBase {
    LoginPage loginPage;

    @BeforeMethod
    public void setup() throws Exception {
        // To call initialization method and assign instance variable
        initialization();
        loginPage = new LoginPage();
    }
```

Testers don't make software; they make it better.

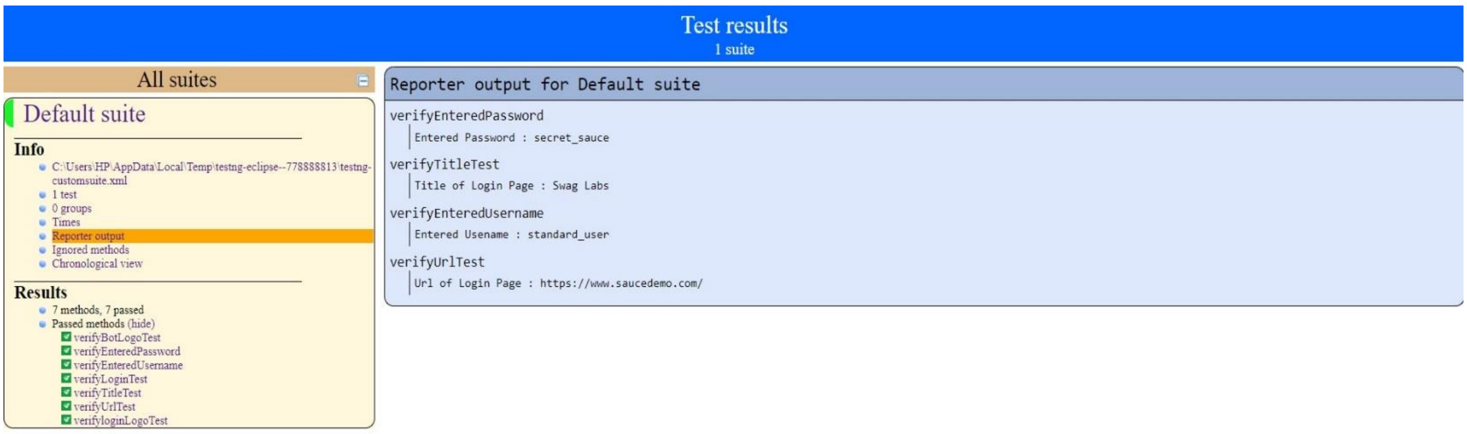
Situ



```
    @Test
    Run | Debug
    public void verifyTitleTest() throws Exception {
        // To verify Title of Login page
        String actual = loginPage.verifyTitle();
        String expected = ReadData.excelData(0, 1);
        Assert.assertEquals(expected, actual);
        Reporter.log("Title of Login Page : " + actual);
    }

    @Test
    Run | Debug
    public void verifyUrlTest() throws Exception {
        // To verify URL of Login page
        String actual = loginPage.verifyUrl();
        String expected = ReadData.excelData(1, 1);
        Assert.assertEquals(expected, actual);
        Reporter.log("Url of Login Page : " + actual);
    }

    @Test
    Run | Debug
    public void verifyloginLogoTest() {
        // To verify logo
        boolean actual = loginPage.verifyLoginLogo();
        Assert.assertTrue(actual);
    }

    @Test
    Run | Debug
    public void verifyBotLogoTest() {
        // To verify logo
        boolean actual = loginPage.verifyBotLogo();
        Assert.assertTrue(actual);
    }

    @Test(priority = 1)
    Run | Debug
    public void verifyLoginTest() throws Exception {
        // To verify Login to application
        login();
        boolean actual = loginPage.getLogoutBtn().isEnabled();
        Assert.assertTrue(actual, "Login is not Successful");
    }

    @Test
    Run | Debug
    public void verifyEnteredUsername() throws Exception {
        // To verify UserName should be Entered in userName textbox
        loginPage.getUserNameTextbox().sendKeys(ReadData.readPropertyFile("username"));
        String actual = loginPage.getUserNameTextbox().getAttribute("value");
        String expected = ReadData.readPropertyFile("username");
        Assert.assertEquals(actual, expected);
        Reporter.log("Entered Usename : " + actual);
    }

    @Test
    Run | Debug
    public void verifyEnteredPassword() throws Exception {
        // To verify password should be entered in password textbox
        loginPage.getPasswordTextbox().sendKeys(ReadData.readPropertyFile("password"));
        String actual = loginPage.getPasswordTextbox().getAttribute("value");
        String expected = ReadData.readPropertyFile("password");
        Assert.assertEquals(actual, expected);
        Reporter.log("Entered Password : " + actual);
    }

    @AfterMethod
    public void browserClose() {
        driver.close();
    }

}
```

Testers don't make software; they make it better.

LoginPageTest Class Report:



Test results
1 suite

All suites

Default suite

Info

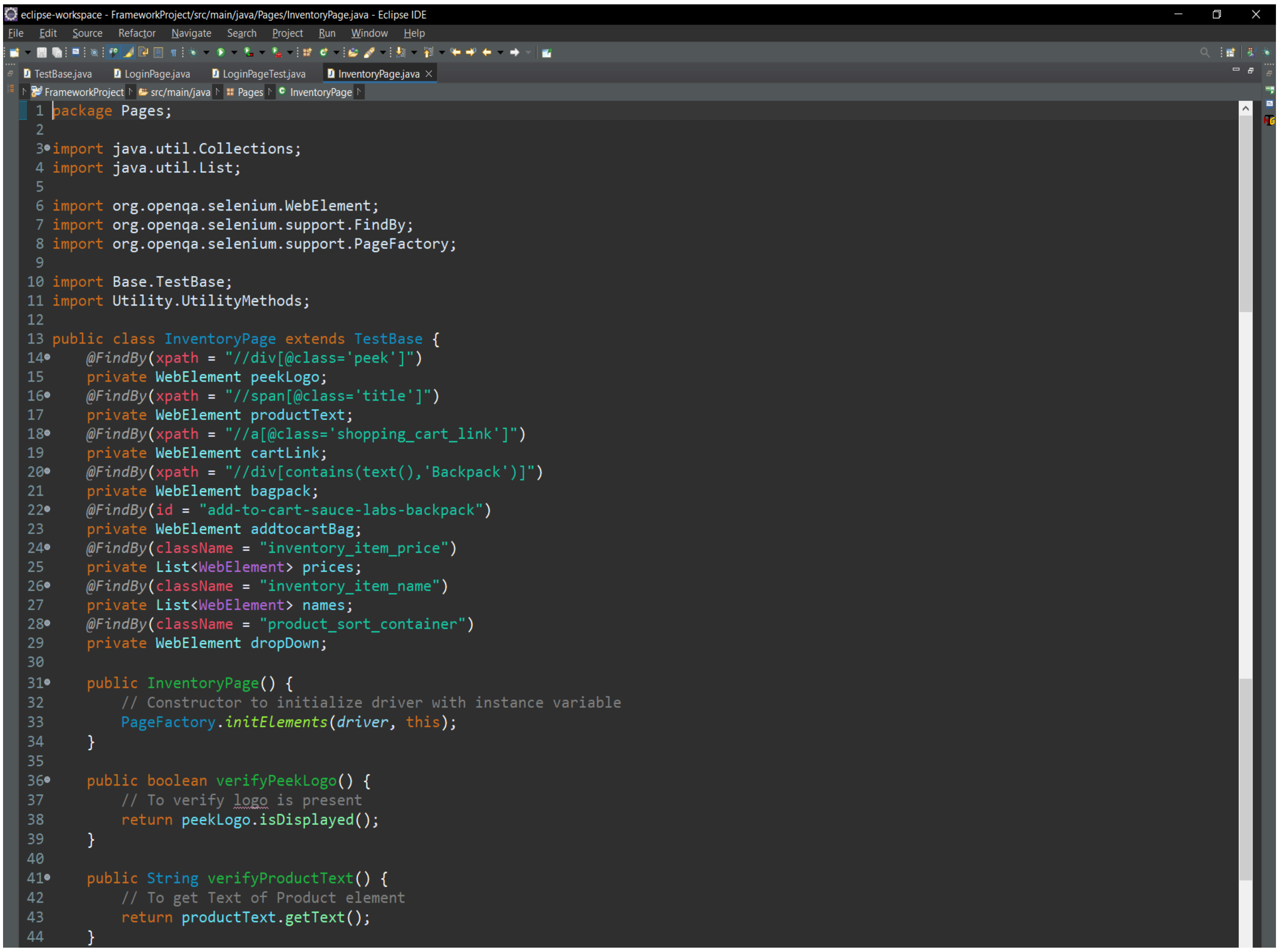
- C:\Users\HP\AppData\Local\Temp\testng-eclipse--778888813\testng-customsuite.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 7 methods, 7 passed
- Passed methods (hide)
  - ✓ verifyBotLogoTest
  - ✓ verifyEnteredPassword
  - ✓ verifyEnteredUsername
  - ✓ verifyLoginTest
  - ✓ verifyTitleTest
  - ✓ verifyUrlTest
  - ✓ verifyloginLogoTest

```
Reporter output for Default suite
verifyEnteredPassword
    Entered Password : secret_sauce
verifyTitleTest
    Title of Login Page : Swag Labs
verifyEnteredUsername
    Entered Usename : standard_user
verifyUrlTest
    Url of Login Page : https://www.saucedemo.com/
```

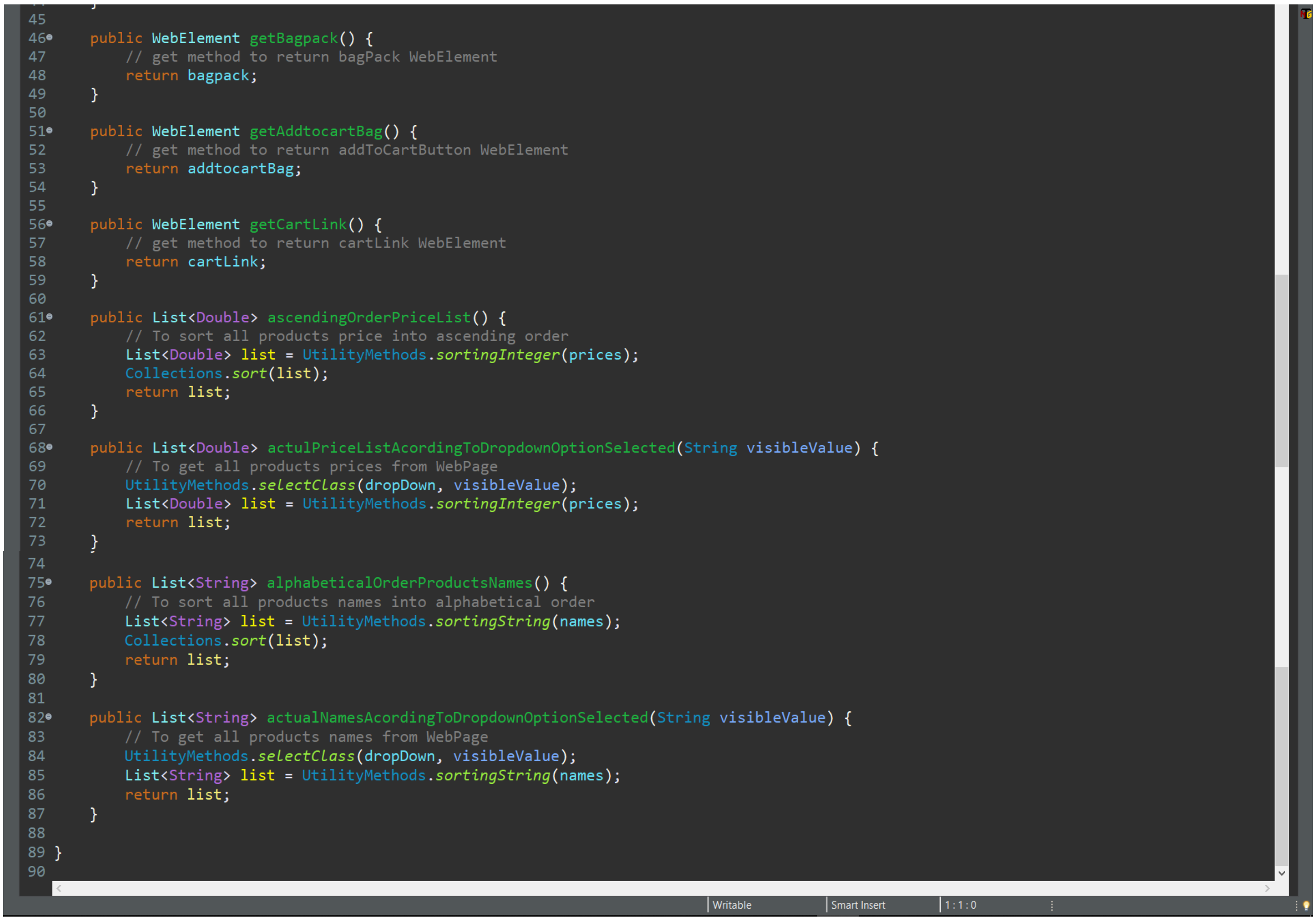
InventoryPage Class:



eclipse-workspace - FrameworkProject/src/main/java/Pages/InventoryPage.java - Eclipse IDE

```java
package Pages;

import java.util.Collections;
import java.util.List;

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

import Base.TestBase;
import Utility.UtilityMethods;

public class InventoryPage extends TestBase {
    @FindBy(xpath = "//div[@class='peek']")
    private WebElement peekLogo;
    @FindBy(xpath = "//span[@class='title']")
    private WebElement productText;
    @FindBy(xpath = "//a[@class='shopping_cart_link']")
    private WebElement cartLink;
    @FindBy(xpath = "//div[contains(text(),'Backpack')]")
    private WebElement bagpack;
    @FindBy(id = "add-to-cart-sauce-labs-backpack")
    private WebElement addtocartBag;
    @FindBy(className = "inventory_item_price")
    private List<WebElement> prices;
    @FindBy(className = "inventory_item_name")
    private List<WebElement> names;
    @FindBy(className = "product_sort_container")
    private WebElement dropDown;

    public InventoryPage() {
        // Constructor to initialize driver with instance variable
        PageFactory.initElements(driver, this);
    }

    public boolean verifyPeekLogo() {
        // To verify logo is present
        return peekLogo.isDisplayed();
    }

    public String verifyProductText() {
        // To get Text of Product element
        return productText.getText();
    }
```

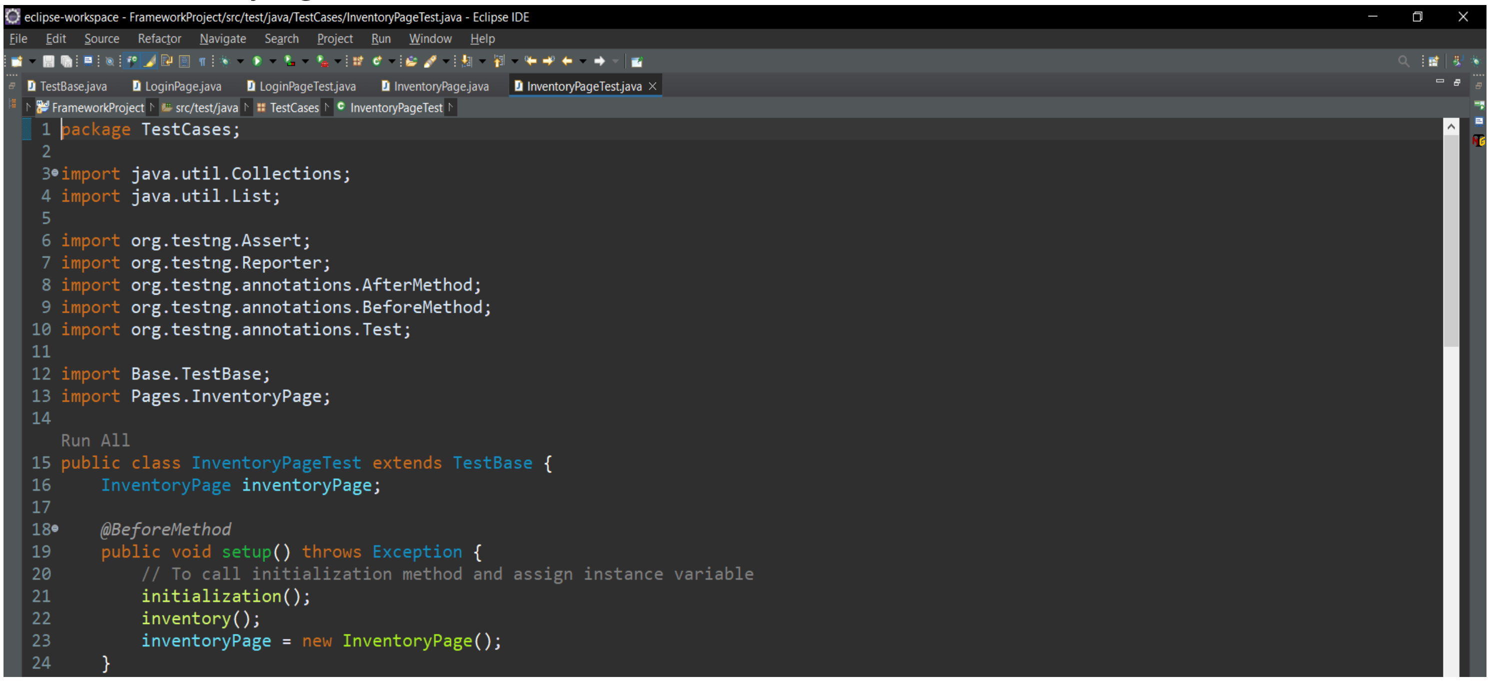
Situ



```
    public WebElement getBagpack() {
        // get method to return bagPack WebElement
        return bagpack;
    }

    public WebElement getAddtocartBag() {
        // get method to return addToCartButton WebElement
        return addtocartBag;
    }

    public WebElement getCartLink() {
        // get method to return cartLink WebElement
        return cartLink;
    }

    public List<Double> ascendingOrderPriceList() {
        // To sort all products price into ascending order
        List<Double> list = UtilityMethods.sortingInteger(prices);
        Collections.sort(list);
        return list;
    }

    public List<Double> actulPriceListAcordingToDropdownOptionSelected(String visibleValue) {
        // To get all products prices from WebPage
        UtilityMethods.selectClass(dropDown, visibleValue);
        List<Double> list = UtilityMethods.sortingInteger(prices);
        return list;
    }

    public List<String> alphabeticalOrderProductsNames() {
        // To sort all products names into alphabetical order
        List<String> list = UtilityMethods.sortingString(names);
        Collections.sort(list);
        return list;
    }

    public List<String> actualNamesAcordingToDropdownOptionSelected(String visibleValue) {
        // To get all products names from WebPage
        UtilityMethods.selectClass(dropDown, visibleValue);
        List<String> list = UtilityMethods.sortingString(names);
        return list;
    }

}
```

Writable | Smart Insert | 1 : 1 : 0

InventoryPageTest Class:



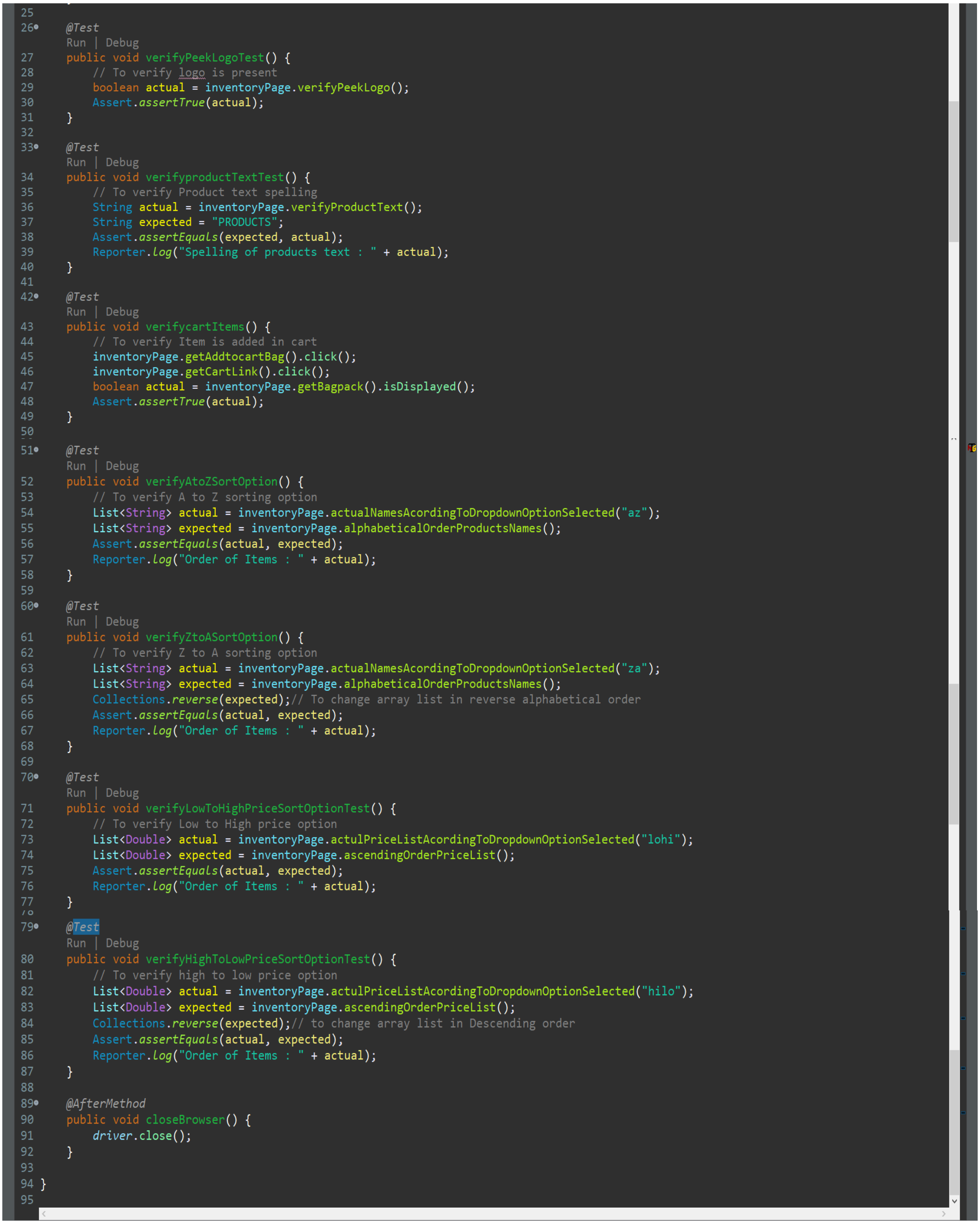
eclipse-workspace - FrameworkProject/src/test/java/TestCases/InventoryPageTest.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help
TestBase.java LoginPage.java LoginPageTest.java InventoryPage.java InventoryPageTest.java
FrameworkProject > src/test/java > TestCases > InventoryPageTest

```
package TestCases;

import java.util.Collections;
import java.util.List;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.InventoryPage;

Run All
public class InventoryPageTest extends TestBase {
    InventoryPage inventoryPage;

    @BeforeMethod
    public void setup() throws Exception {
        // To call initialization method and assign instance variable
        initialization();
        inventory();
        inventoryPage = new InventoryPage();
    }
```

Testers don't make software; they make it better.

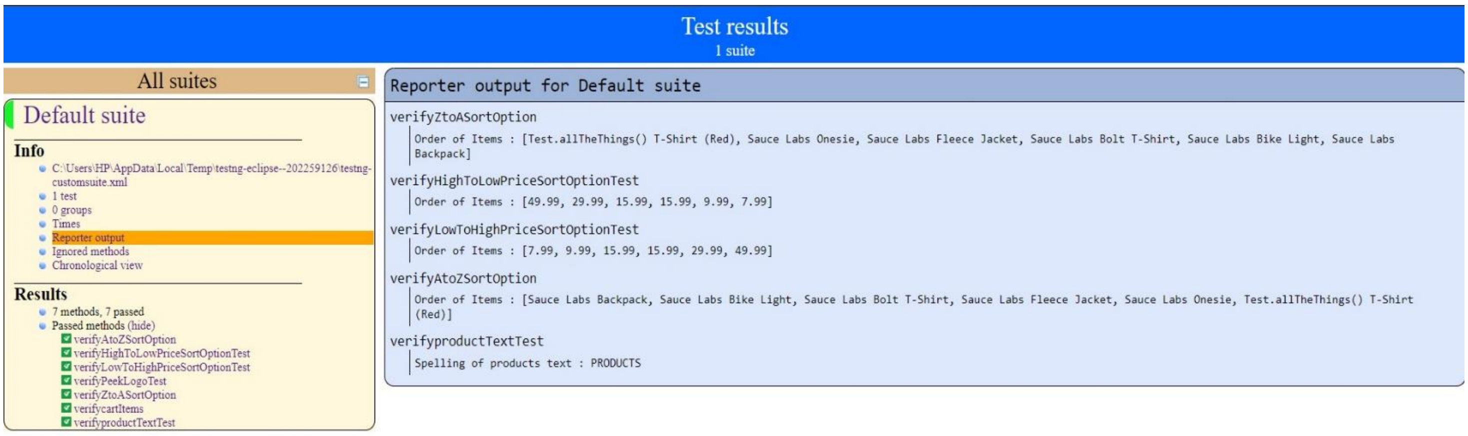
Situ



```
    @Test
    Run | Debug
    public void verifyPeekLogoTest() {
        // To verify logo is present
        boolean actual = inventoryPage.verifyPeekLogo();
        Assert.assertTrue(actual);
    }

    @Test
    Run | Debug
    public void verifyproductTextTest() {
        // To verify Product text spelling
        String actual = inventoryPage.verifyProductText();
        String expected = "PRODUCTS";
        Assert.assertEquals(expected, actual);
        Reporter.log("Spelling of products text : " + actual);
    }

    @Test
    Run | Debug
    public void verifycartItems() {
        // To verify Item is added in cart
        inventoryPage.getAddtocartBag().click();
        inventoryPage.getCartLink().click();
        boolean actual = inventoryPage.getBagpack().isDisplayed();
        Assert.assertTrue(actual);
    }

    @Test
    Run | Debug
    public void verifyAtoZSortOption() {
        // To verify A to Z sorting option
        List<String> actual = inventoryPage.actualNamesAcordingToDropdownOptionSelected("az");
        List<String> expected = inventoryPage.alphabeticalOrderProductsNames();
        Assert.assertEquals(actual, expected);
        Reporter.log("Order of Items : " + actual);
    }

    @Test
    Run | Debug
    public void verifyZtoASortOption() {
        // To verify Z to A sorting option
        List<String> actual = inventoryPage.actualNamesAcordingToDropdownOptionSelected("za");
        List<String> expected = inventoryPage.alphabeticalOrderProductsNames();
        Collections.reverse(expected);// To change array list in reverse alphabetical order
        Assert.assertEquals(actual, expected);
        Reporter.log("Order of Items : " + actual);
    }

    @Test
    Run | Debug
    public void verifyLowToHighPriceSortOptionTest() {
        // To verify Low to High price option
        List<Double> actual = inventoryPage.actulPriceListAcordingToDropdownOptionSelected("lohi");
        List<Double> expected = inventoryPage.ascendingOrderPriceList();
        Assert.assertEquals(actual, expected);
        Reporter.log("Order of Items : " + actual);
    }

    @Test
    Run | Debug
    public void verifyHighToLowPriceSortOptionTest() {
        // To verify high to low price option
        List<Double> actual = inventoryPage.actulPriceListAcordingToDropdownOptionSelected("hilo");
        List<Double> expected = inventoryPage.ascendingOrderPriceList();
        Collections.reverse(expected);// to change array list in Descending order
        Assert.assertEquals(actual, expected);
        Reporter.log("Order of Items : " + actual);
    }

    @AfterMethod
    public void closeBrowser() {
        driver.close();
    }

}
```

Testers don't make software; they make it better.

Inventory Report:

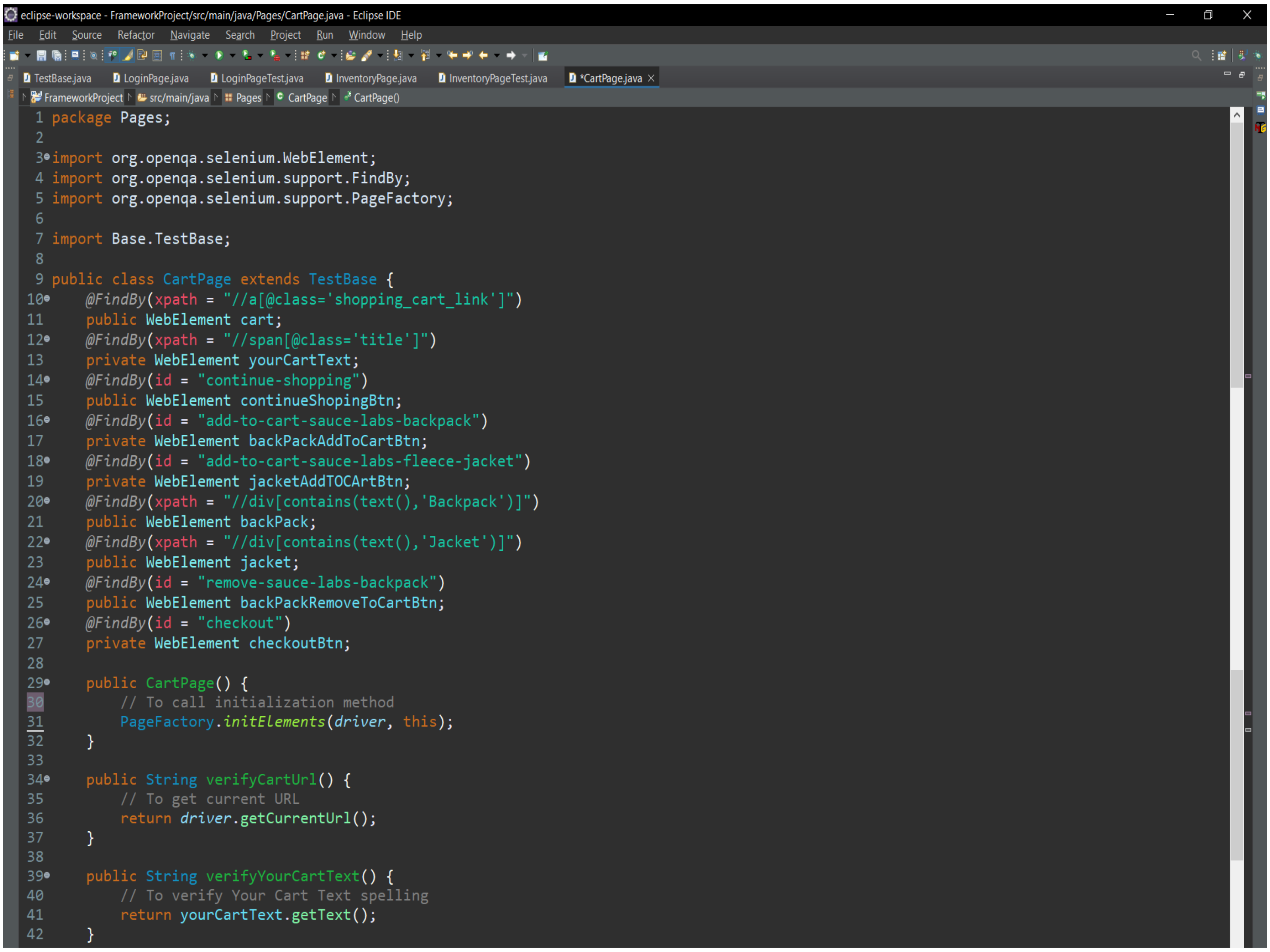


Info
- C:\Users\HP\AppData\Local\Temp\testng-eclipse--202259126\testng-customsuite.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results
- 7 methods, 7 passed
- Passed methods (hide)
  - ✓ verifyAtoZSortOption
  - ✓ verifyHighToLowPriceSortOptionTest
  - ✓ verifyLowToHighPriceSortOptionTest
  - ✓ verifyPeekLogoTest
  - ✓ verifyZtoASortOption
  - ✓ verifycartItems
  - ✓ verifyproductTextTest

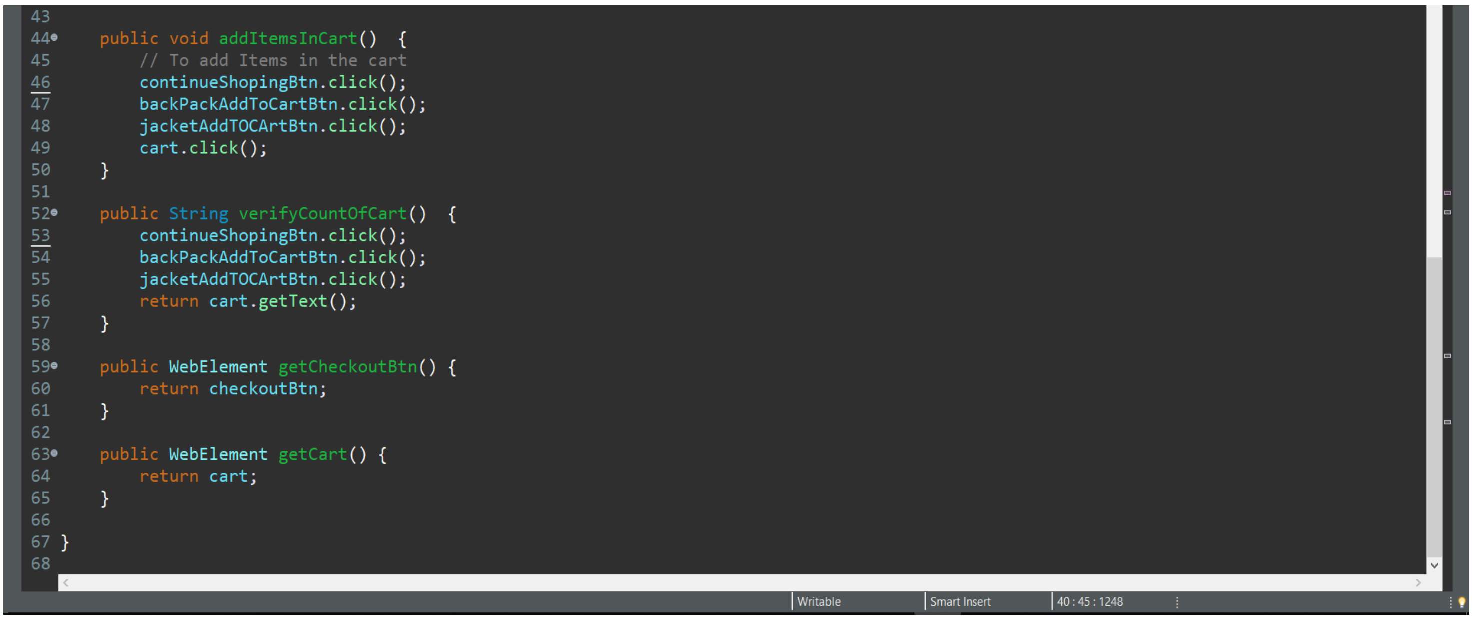
```
Reporter output for Default suite

verifyZtoASortOption
    Order of Items : [Test.allTheThings() T-Shirt (Red), Sauce Labs Onesie, Sauce Labs Fleece Jacket, Sauce Labs Bolt T-Shirt, Sauce Labs Bike Light, Sauce Labs
    Backpack]
verifyHighToLowPriceSortOptionTest
    Order of Items : [49.99, 29.99, 15.99, 15.99, 9.99, 7.99]
verifyLowToHighPriceSortOptionTest
    Order of Items : [7.99, 9.99, 15.99, 15.99, 29.99, 49.99]
verifyAtoZSortOption
    Order of Items : [Sauce Labs Backpack, Sauce Labs Bike Light, Sauce Labs Bolt T-Shirt, Sauce Labs Fleece Jacket, Sauce Labs Onesie, Test.allTheThings() T-Shirt
    (Red)]
verifyproductTextTest
    Spelling of products text : PRODUCTS
```

CartPage Class:



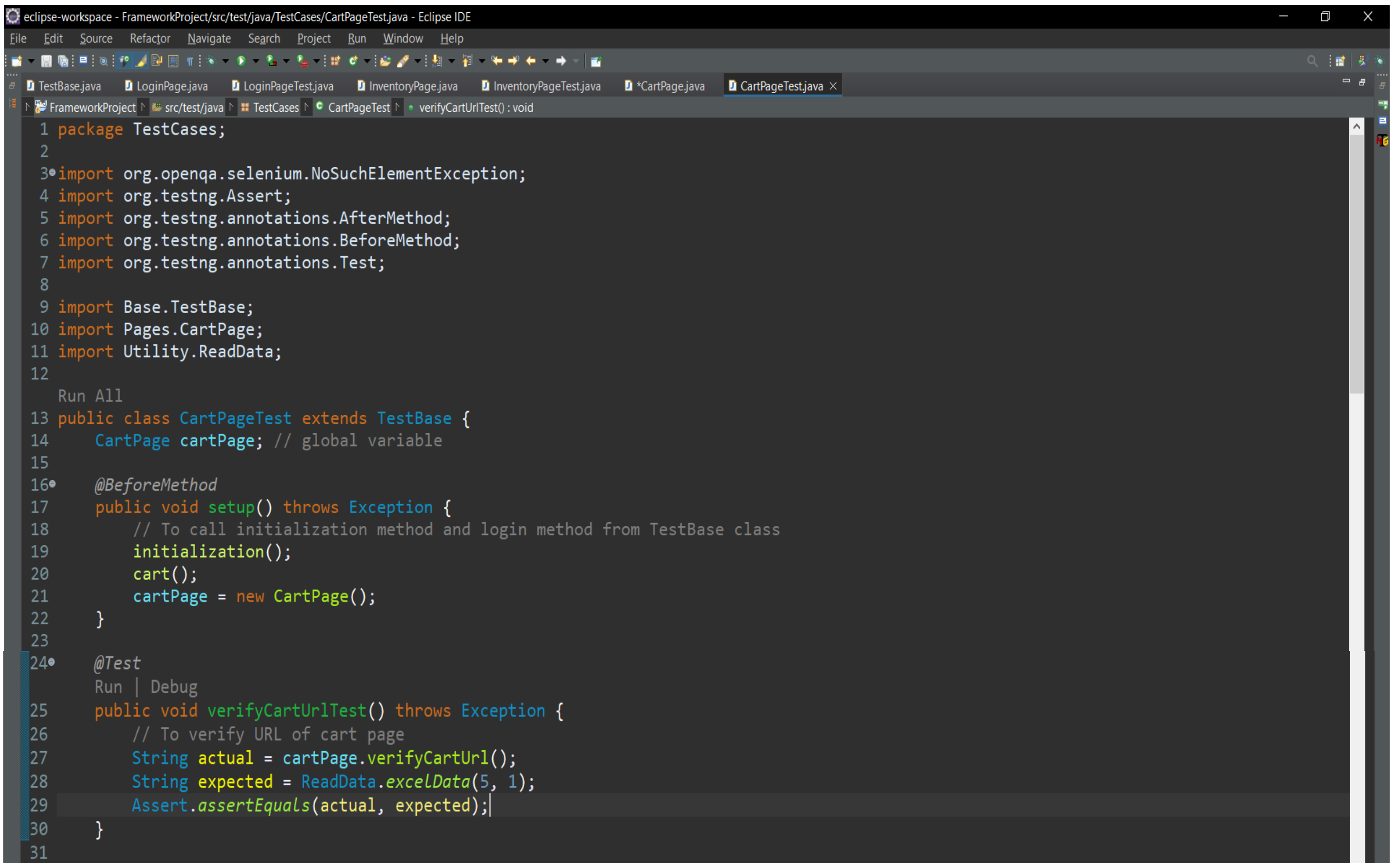
```java
package Pages;

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

import Base.TestBase;

public class CartPage extends TestBase {
    @FindBy(xpath = "//a[@class='shopping_cart_link']")
    public WebElement cart;
    @FindBy(xpath = "//span[@class='title']")
    private WebElement yourCartText;
    @FindBy(id = "continue-shopping")
    public WebElement continueShopingBtn;
    @FindBy(id = "add-to-cart-sauce-labs-backpack")
    private WebElement backPackAddToCartBtn;
    @FindBy(id = "add-to-cart-sauce-labs-fleece-jacket")
    private WebElement jacketAddTOCArtBtn;
    @FindBy(xpath = "//div[contains(text(),'Backpack')]")
    public WebElement backPack;
    @FindBy(xpath = "//div[contains(text(),'Jacket')]")
    public WebElement jacket;
    @FindBy(id = "remove-sauce-labs-backpack")
    public WebElement backPackRemoveToCartBtn;
    @FindBy(id = "checkout")
    private WebElement checkoutBtn;

    public CartPage() {
        // To call initialization method
        PageFactory.initElements(driver, this);
    }

    public String verifyCartUrl() {
        // To get current URL
        return driver.getCurrentUrl();
    }

    public String verifyYourCartText() {
        // To verify Your Cart Text spelling
        return yourCartText.getText();
    }
```

Situ



```
    public void addItemsInCart()  {
        // To add Items in the cart
        continueShopingBtn.click();
        backPackAddToCartBtn.click();
        jacketAddTOCArtBtn.click();
        cart.click();
    }

    public String verifyCountOfCart()  {
        continueShopingBtn.click();
        backPackAddToCartBtn.click();
        jacketAddTOCArtBtn.click();
        return cart.getText();
    }

    public WebElement getCheckoutBtn() {
        return checkoutBtn;
    }

    public WebElement getCart() {
        return cart;
    }

}
```

<u>CartPageTest Class:</u>

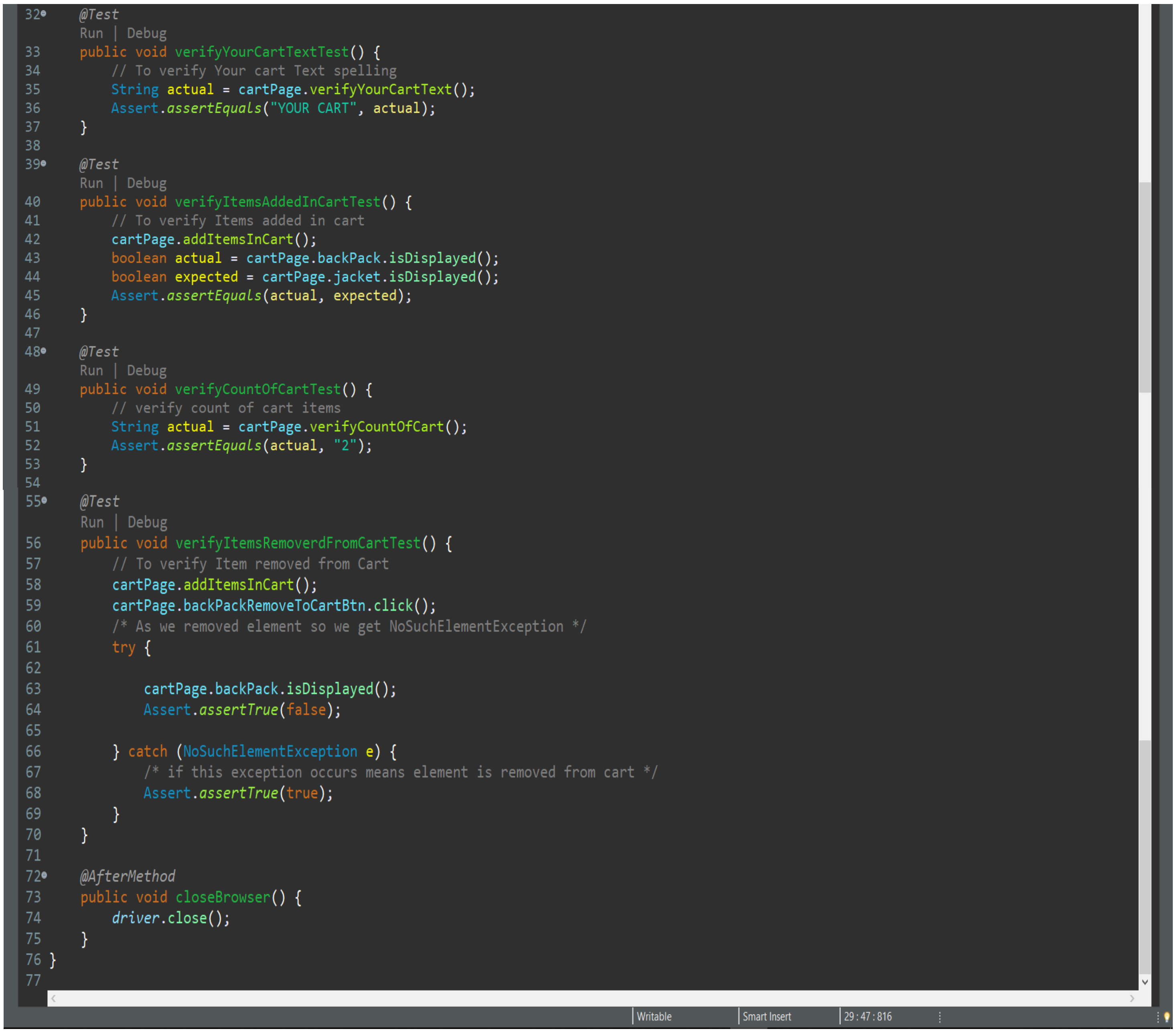


eclipse-workspace - FrameworkProject/src/test/java/TestCases/CartPageTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
TestBase.java
LoginPage.java
LoginPageTest.java
InventoryPage.java
InventoryPageTest.java
*CartPage.java
CartPageTest.java
FrameworkProject
src/test/java
TestCases
CartPageTest
verifyCartUrlTest() : void

```
package TestCases;

import org.openqa.selenium.NoSuchElementException;
import org.testng.Assert;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.CartPage;
import Utility.ReadData;

Run All
public class CartPageTest extends TestBase {
    CartPage cartPage; // global variable

    @BeforeMethod
    public void setup() throws Exception {
        // To call initialization method and login method from TestBase class
        initialization();
        cart();
        cartPage = new CartPage();
    }

    @Test
    Run | Debug
    public void verifyCartUrlTest() throws Exception {
        // To verify URL of cart page
        String actual = cartPage.verifyCartUrl();
        String expected = ReadData.excelData(5, 1);
        Assert.assertEquals(actual, expected);
    }

```

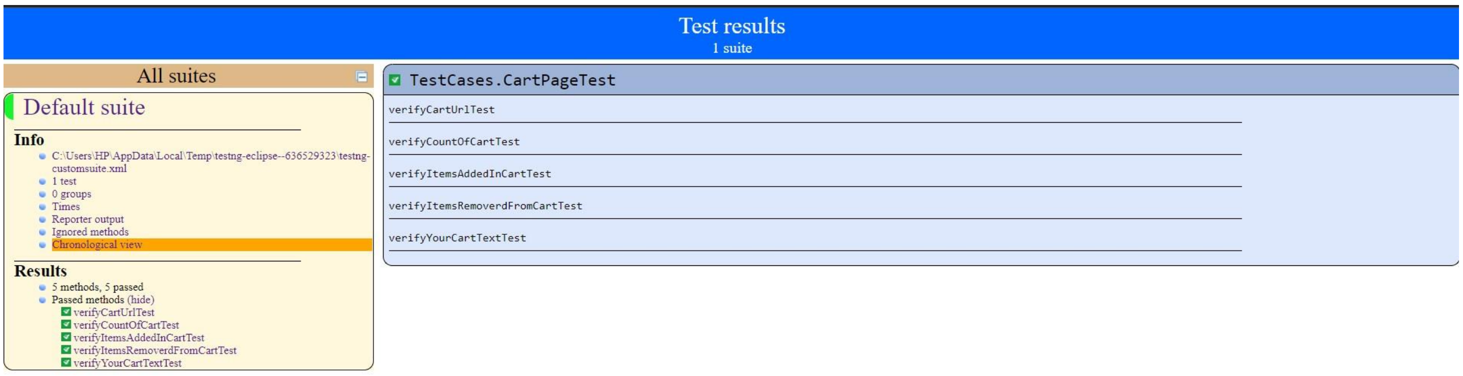
Testers don't make software; they make it better.

Situ



```
    @Test
    Run | Debug
    public void verifyYourCartTextTest() {
        // To verify Your cart Text spelling
        String actual = cartPage.verifyYourCartText();
        Assert.assertEquals("YOUR CART", actual);
    }

    @Test
    Run | Debug
    public void verifyItemsAddedInCartTest() {
        // To verify Items added in cart
        cartPage.addItemsInCart();
        boolean actual = cartPage.backPack.isDisplayed();
        boolean expected = cartPage.jacket.isDisplayed();
        Assert.assertEquals(actual, expected);
    }

    @Test
    Run | Debug
    public void verifyCountOfCartTest() {
        // verify count of cart items
        String actual = cartPage.verifyCountOfCart();
        Assert.assertEquals(actual, "2");
    }

    @Test
    Run | Debug
    public void verifyItemsRemoverdFromCartTest() {
        // To verify Item removed from Cart
        cartPage.addItemsInCart();
        cartPage.backPackRemoveToCartBtn.click();
        /* As we removed element so we get NoSuchElementException */
        try {

            cartPage.backPack.isDisplayed();
            Assert.assertTrue(false);

        } catch (NoSuchElementException e) {
            /* if this exception occurs means element is removed from cart */
            Assert.assertTrue(true);
        }
    }

    @AfterMethod
    public void closeBrowser() {
        driver.close();
    }
}
```

<u>Cart Report:</u>



Test results
1 suite

All suites

Default suite

Info
- C:\Users\HP\AppData\Local\Temp\testng-eclipse--636529323\testng-customsuite.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

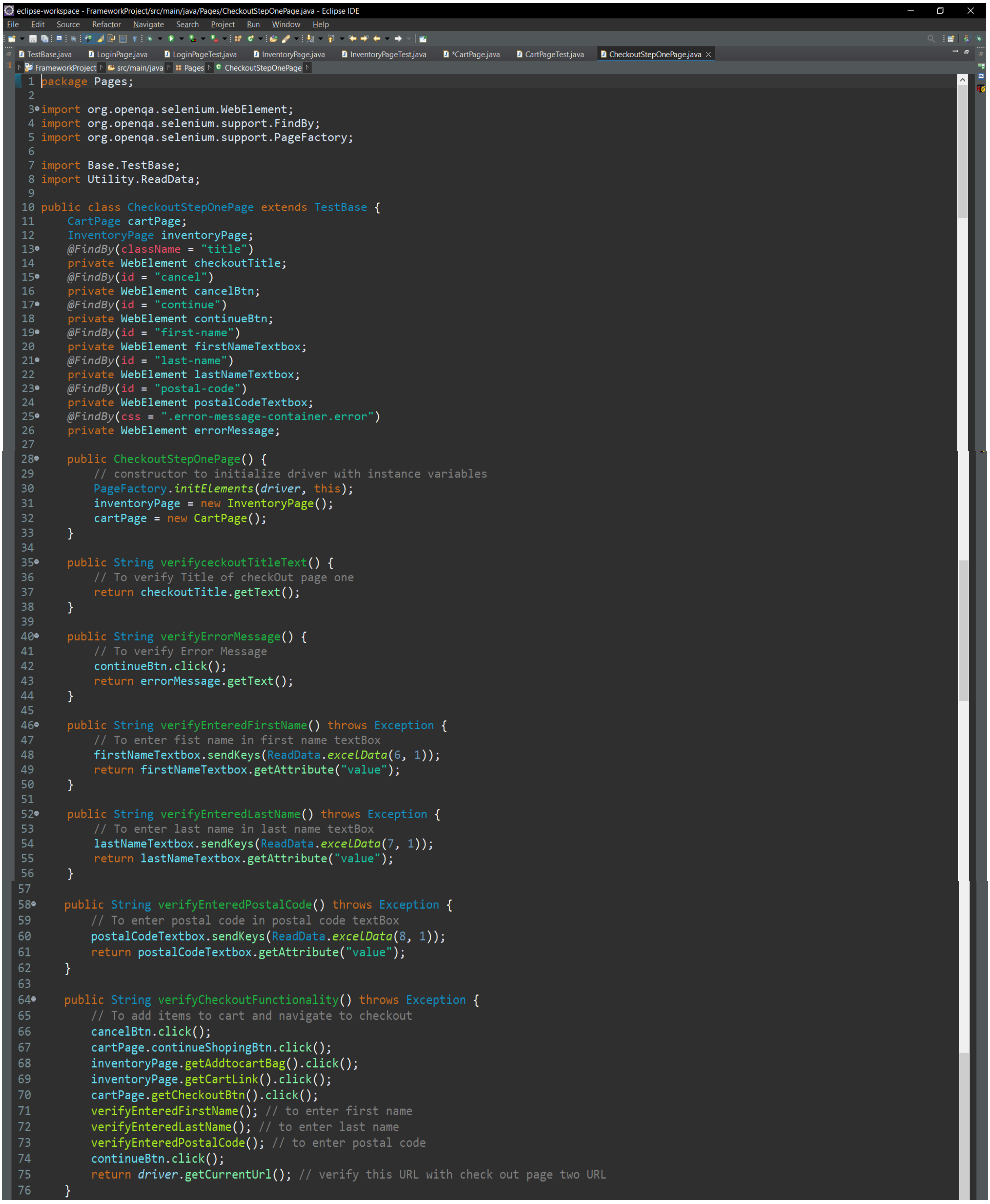
Results
- 5 methods, 5 passed
- Passed methods (hide)
  - ✓ verifyCartUrlTest
  - ✓ verifyCountOfCartTest
  - ✓ verifyItemsAddedInCartTest
  - ✓ verifyItemsRemoverdFromCartTest
  - ✓ verifyYourCartTextTest

✓ TestCases.CartPageTest

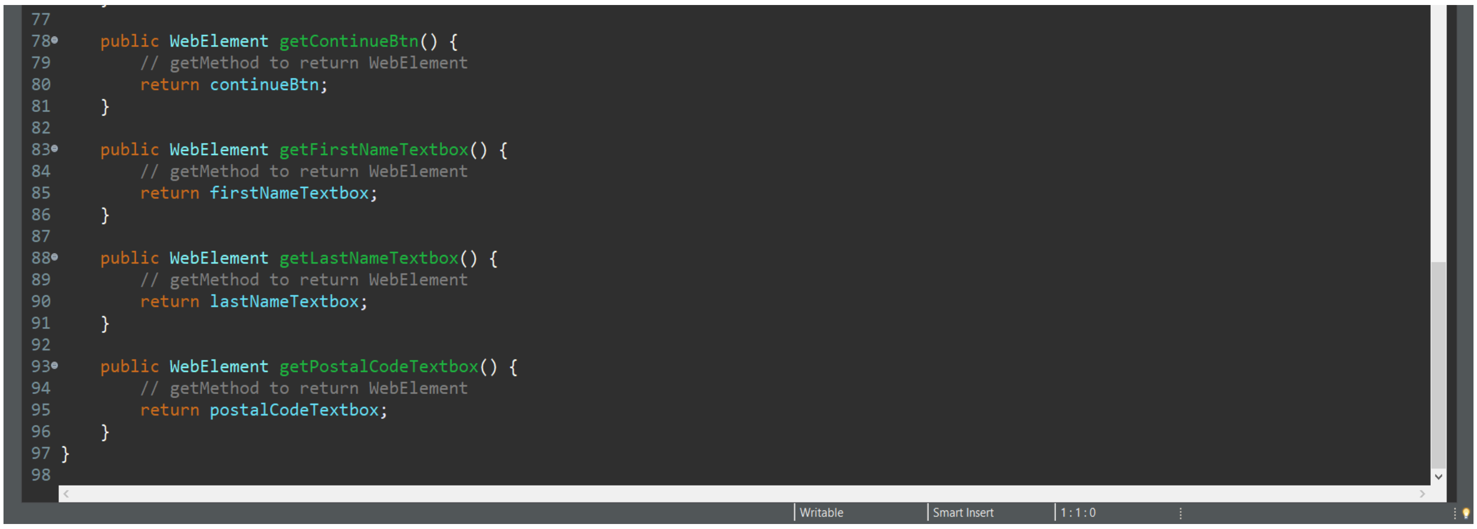
```
verifyCartUrlTest
verifyCountOfCartTest
verifyItemsAddedInCartTest
verifyItemsRemoverdFromCartTest
verifyYourCartTextTest
```

Testers don't make software; they make it better.

CheckOutStepOnePage Class:


eclipse-workspace - FrameworkProject/src/main/java/Pages/CheckoutStepOnePage.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
TestBase.java LoginPage.java LoginPageTest.java InventoryPage.java InventoryPageTest.java *CartPage.java CartPageTest.java CheckoutStepOnePage.java
FrameworkProject src/main/java Pages CheckoutStepOnePage

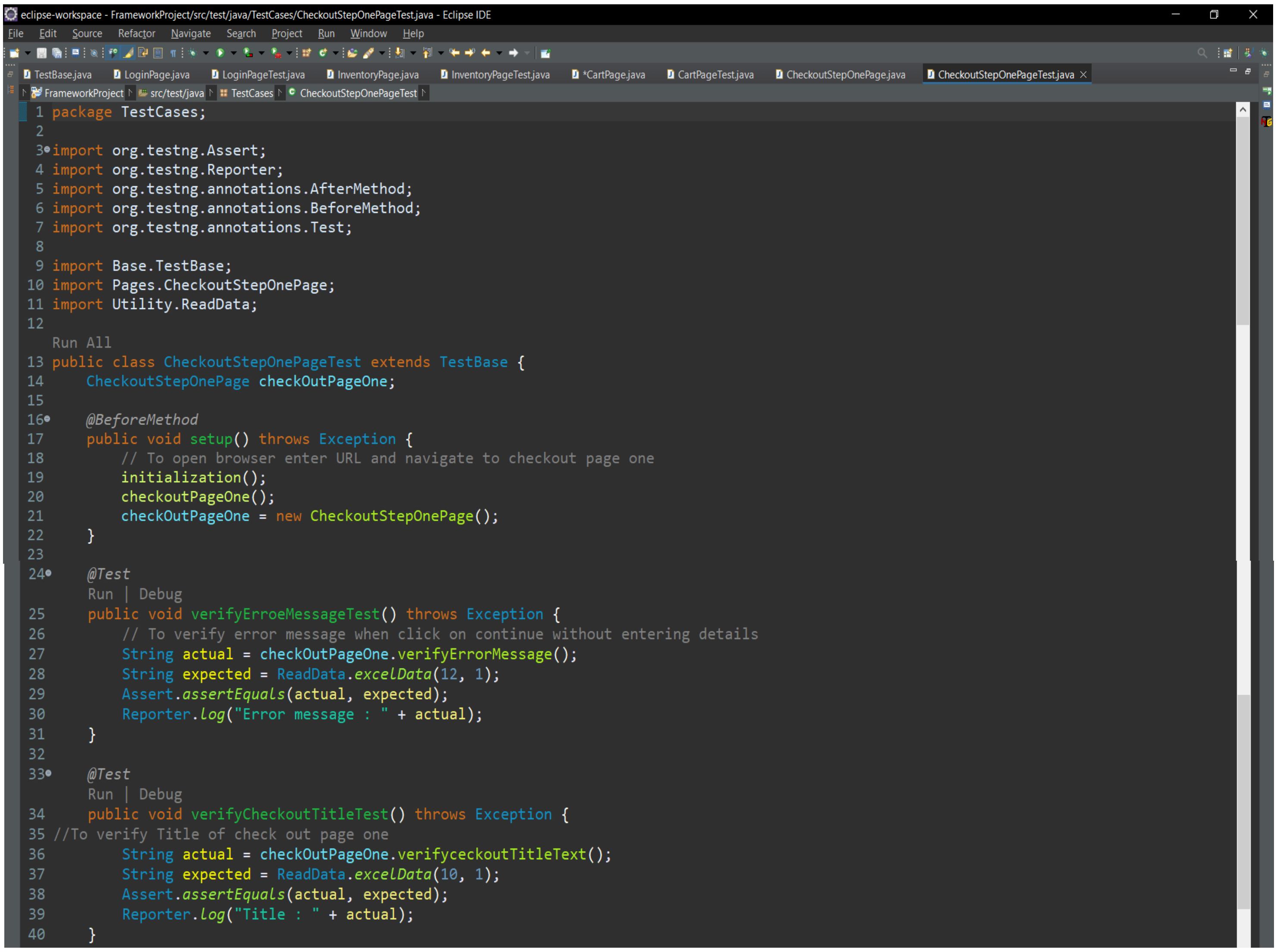
```java
package Pages;

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

import Base.TestBase;
import Utility.ReadData;

public class CheckoutStepOnePage extends TestBase {
    CartPage cartPage;
    InventoryPage inventoryPage;
    @FindBy(className = "title")
    private WebElement checkoutTitle;
    @FindBy(id = "cancel")
    private WebElement cancelBtn;
    @FindBy(id = "continue")
    private WebElement continueBtn;
    @FindBy(id = "first-name")
    private WebElement firstNameTextbox;
    @FindBy(id = "last-name")
    private WebElement lastNameTextbox;
    @FindBy(id = "postal-code")
    private WebElement postalCodeTextbox;
    @FindBy(css = ".error-message-container.error")
    private WebElement errorMessage;

    public CheckoutStepOnePage() {
        // constructor to initialize driver with instance variables
        PageFactory.initElements(driver, this);
        inventoryPage = new InventoryPage();
        cartPage = new CartPage();
    }

    public String verifyceckoutTitleText() {
        // To verify Title of checkOut page one
        return checkoutTitle.getText();
    }

    public String verifyErrorMessage() {
        // To verify Error Message
        continueBtn.click();
        return errorMessage.getText();
    }

    public String verifyEnteredFirstName() throws Exception {
        // To enter fist name in first name textBox
        firstNameTextbox.sendKeys(ReadData.excelData(6, 1));
        return firstNameTextbox.getAttribute("value");
    }

    public String verifyEnteredLastName() throws Exception {
        // To enter last name in last name textBox
        lastNameTextbox.sendKeys(ReadData.excelData(7, 1));
        return lastNameTextbox.getAttribute("value");
    }

    public String verifyEnteredPostalCode() throws Exception {
        // To enter postal code in postal code textBox
        postalCodeTextbox.sendKeys(ReadData.excelData(8, 1));
        return postalCodeTextbox.getAttribute("value");
    }

    public String verifyCheckoutFunctionality() throws Exception {
        // To add items to cart and navigate to checkout
        cancelBtn.click();
        cartPage.continueShopingBtn.click();
        inventoryPage.getAddtocartBag().click();
        inventoryPage.getCartLink().click();
        cartPage.getCheckoutBtn().click();
        verifyEnteredFirstName(); // to enter first name
        verifyEnteredLastName(); // to enter last name
        verifyEnteredPostalCode(); // to enter postal code
        continueBtn.click();
        return driver.getCurrentUrl(); // verify this URL with check out page two URL
    }
```



```
    public WebElement getContinueBtn() {
        // getMethod to return WebElement
        return continueBtn;
    }

    public WebElement getFirstNameTextbox() {
        // getMethod to return WebElement
        return firstNameTextbox;
    }

    public WebElement getLastNameTextbox() {
        // getMethod to return WebElement
        return lastNameTextbox;
    }

    public WebElement getPostalCodeTextbox() {
        // getMethod to return WebElement
        return postalCodeTextbox;
    }
}
```

Writable | Smart Insert | 1 : 1 : 0

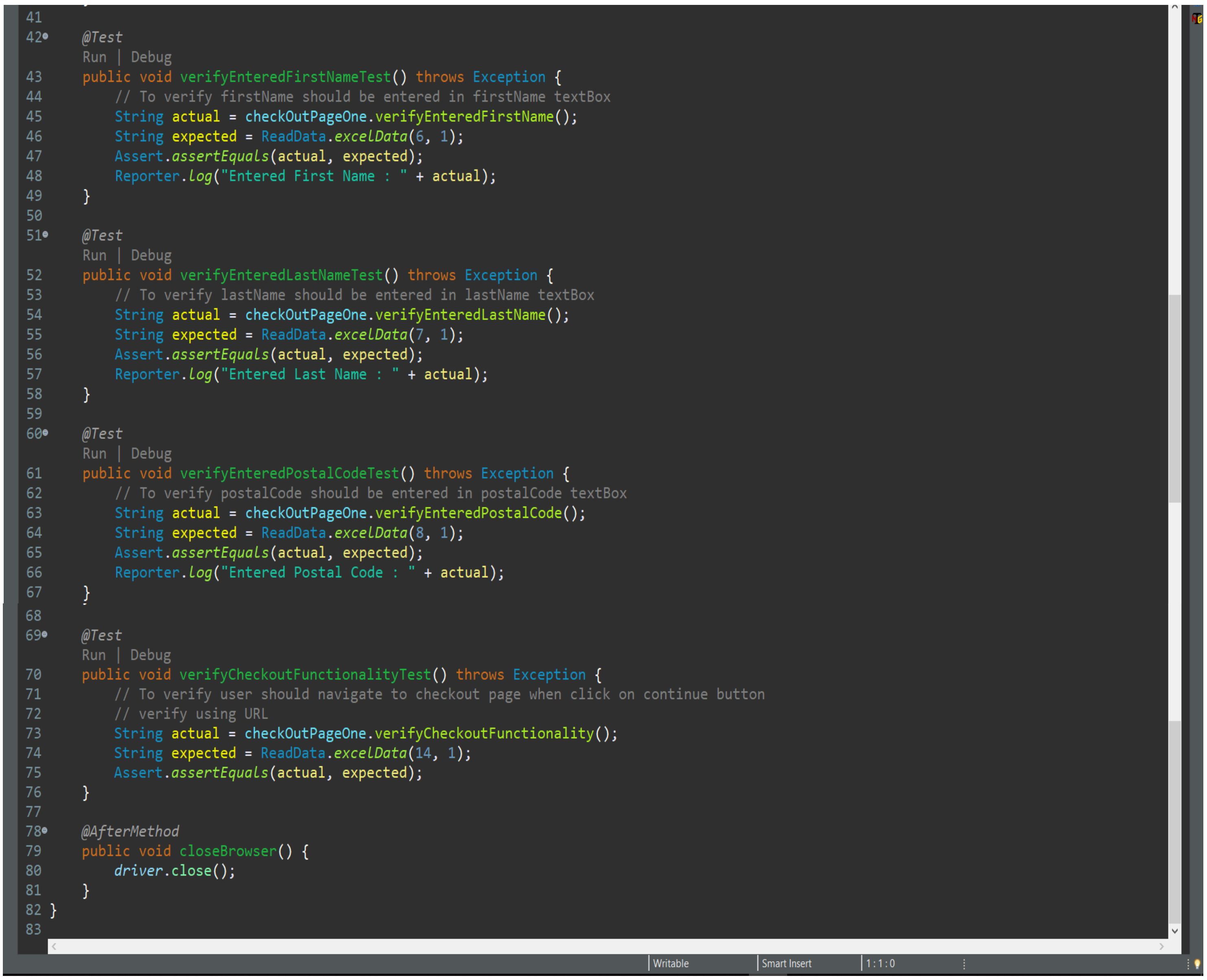
<u>CheckoutStepOnePageTest Class:</u>



eclipse-workspace - FrameworkProject/src/test/java/TestCases/CheckoutStepOnePageTest.java - Eclipse IDE

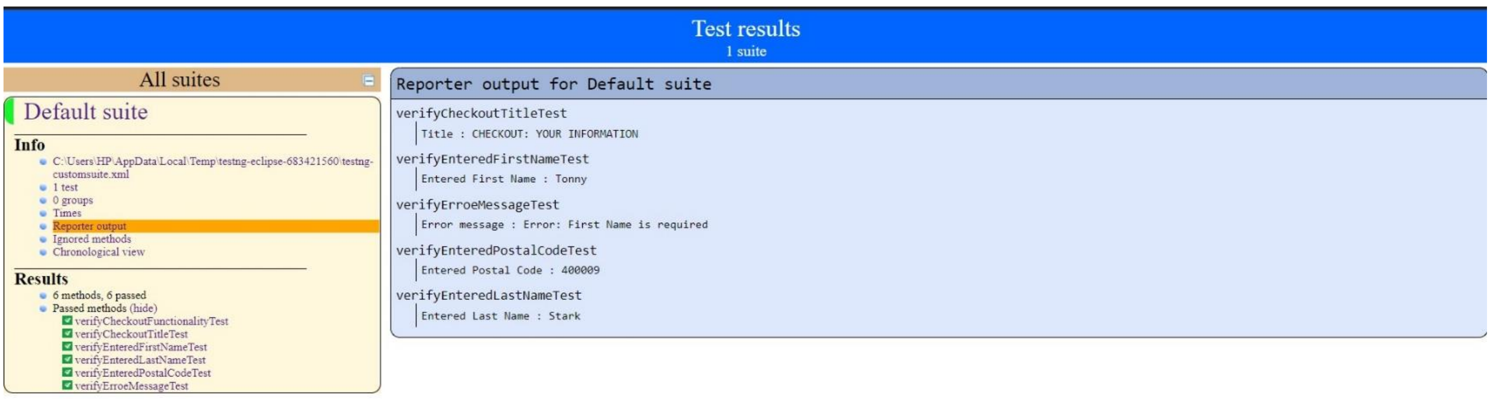
```
package TestCases;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.CheckoutStepOnePage;
import Utility.ReadData;

Run All
public class CheckoutStepOnePageTest extends TestBase {
    CheckoutStepOnePage checkOutPageOne;

    @BeforeMethod
    public void setup() throws Exception {
        // To open browser enter URL and navigate to checkout page one
        initialization();
        checkoutPageOne();
        checkOutPageOne = new CheckoutStepOnePage();
    }

    @Test
    Run | Debug
    public void verifyErroeMessageTest() throws Exception {
        // To verify error message when click on continue without entering details
        String actual = checkOutPageOne.verifyErrorMessage();
        String expected = ReadData.excelData(12, 1);
        Assert.assertEquals(actual, expected);
        Reporter.log("Error message : " + actual);
    }

    @Test
    Run | Debug
    public void verifyCheckoutTitleTest() throws Exception {
//To verify Title of check out page one
        String actual = checkOutPageOne.verifyceckoutTitleText();
        String expected = ReadData.excelData(10, 1);
        Assert.assertEquals(actual, expected);
        Reporter.log("Title : " + actual);
    }
```

Situ



```
    @Test
    Run | Debug
    public void verifyEnteredFirstNameTest() throws Exception {
        // To verify firstName should be entered in firstName textBox
        String actual = checkOutPageOne.verifyEnteredFirstName();
        String expected = ReadData.excelData(6, 1);
        Assert.assertEquals(actual, expected);
        Reporter.log("Entered First Name : " + actual);
    }

    @Test
    Run | Debug
    public void verifyEnteredLastNameTest() throws Exception {
        // To verify lastName should be entered in lastName textBox
        String actual = checkOutPageOne.verifyEnteredLastName();
        String expected = ReadData.excelData(7, 1);
        Assert.assertEquals(actual, expected);
        Reporter.log("Entered Last Name : " + actual);
    }

    @Test
    Run | Debug
    public void verifyEnteredPostalCodeTest() throws Exception {
        // To verify postalCode should be entered in postalCode textBox
        String actual = checkOutPageOne.verifyEnteredPostalCode();
        String expected = ReadData.excelData(8, 1);
        Assert.assertEquals(actual, expected);
        Reporter.log("Entered Postal Code : " + actual);
    }

    @Test
    Run | Debug
    public void verifyCheckoutFunctionalityTest() throws Exception {
        // To verify user should navigate to checkout page when click on continue button
        // verify using URL
        String actual = checkOutPageOne.verifyCheckoutFunctionality();
        String expected = ReadData.excelData(14, 1);
        Assert.assertEquals(actual, expected);
    }

    @AfterMethod
    public void closeBrowser() {
        driver.close();
    }
}
```

<u>Checkout Page One Report:</u>



Test results
1 suite

All suites

Default suite

Info

- C:\Users\HP\AppData\Local\Temp\testng-eclipse-683421560\testng-customsuite.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 6 methods, 6 passed
- Passed methods (hide)
  - ✓ verifyCheckoutFunctionalityTest
  - ✓ verifyCheckoutTitleTest
  - ✓ verifyEnteredFirstNameTest
  - ✓ verifyEnteredLastNameTest
  - ✓ verifyEnteredPostalCodeTest
  - ✓ verifyErroeMessageTest

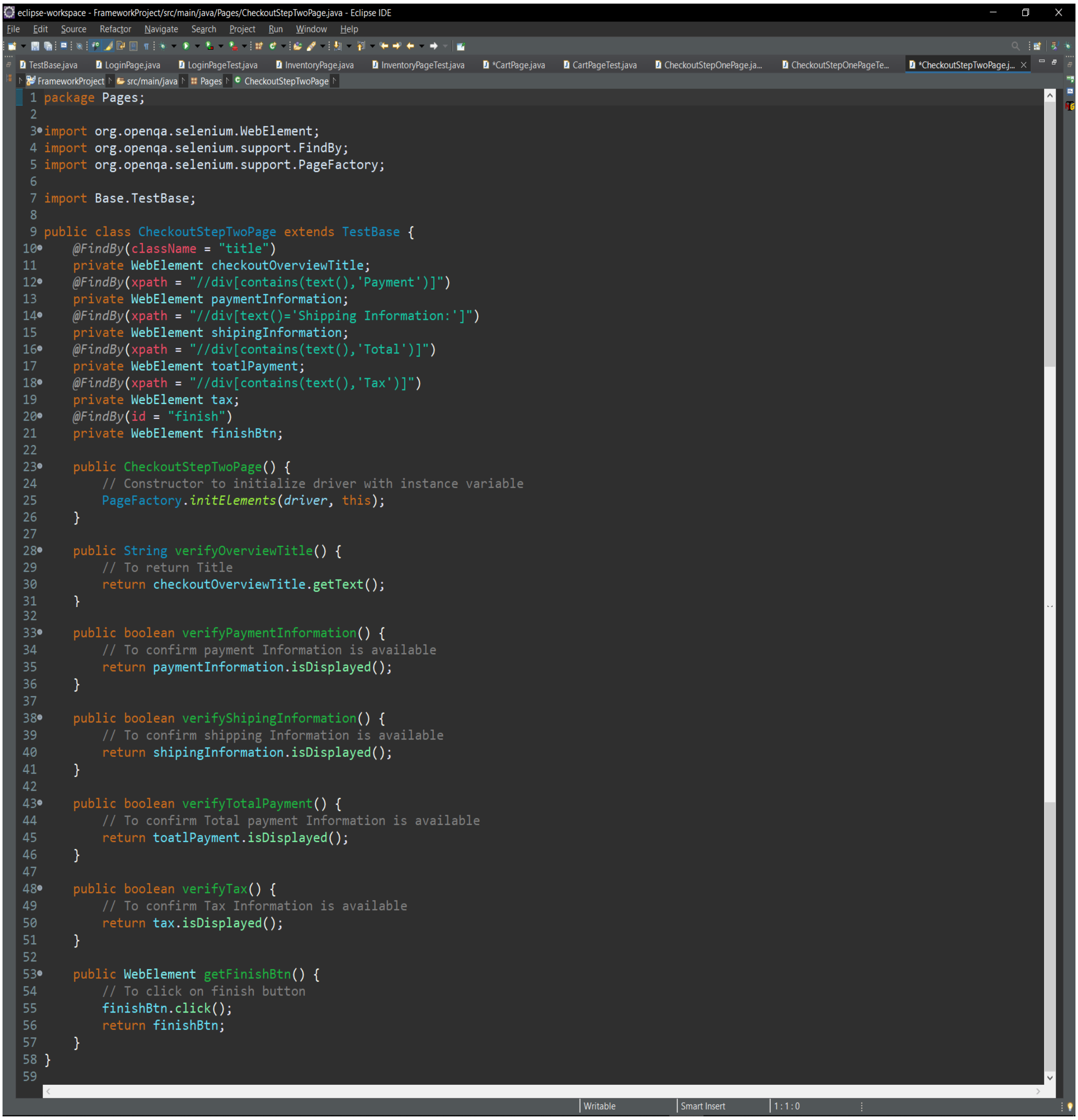
Reporter output for Default suite

```
verifyCheckoutTitleTest
    Title : CHECKOUT: YOUR INFORMATION
verifyEnteredFirstNameTest
    Entered First Name : Tonny
verifyErroeMessageTest
    Error message : Error: First Name is required
verifyEnteredPostalCodeTest
    Entered Postal Code : 400009
verifyEnteredLastNameTest
    Entered Last Name : Stark
```

Testers don't make software; they make it better.

Situ

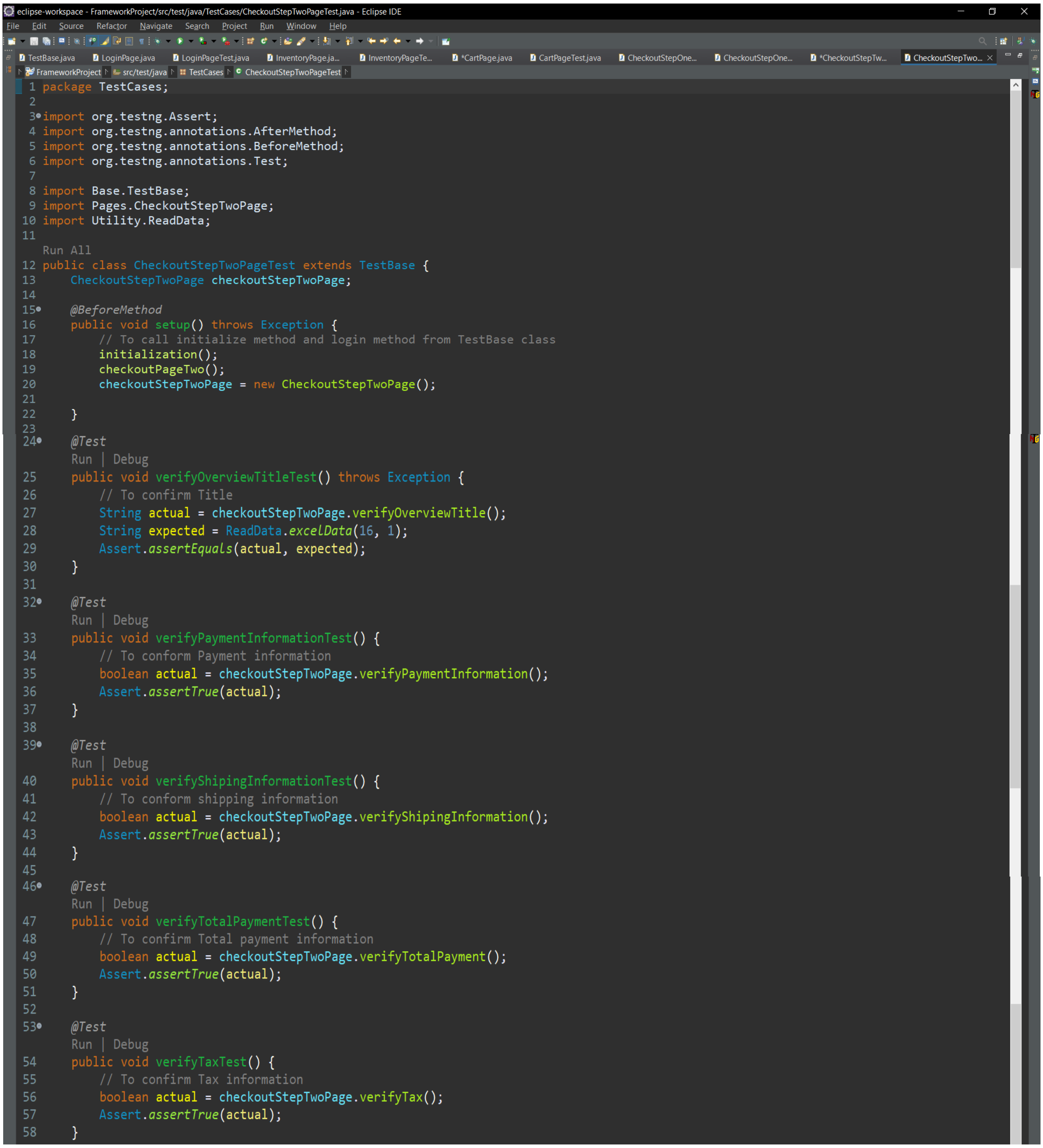
CheckoutStepTwoPage Class:


eclipse-workspace - FrameworkProject/src/main/java/Pages/CheckoutStepTwoPage.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
TestBase.java LoginPage.java LoginPageTest.java InventoryPage.java InventoryPageTest.java *CartPage.java CartPageTest.java CheckoutStepOnePage.ja... CheckoutStepOnePageTe... *CheckoutStepTwoPage.j...
FrameworkProject src/main/java Pages CheckoutStepTwoPage
Writable
Smart Insert
1 : 1 : 0

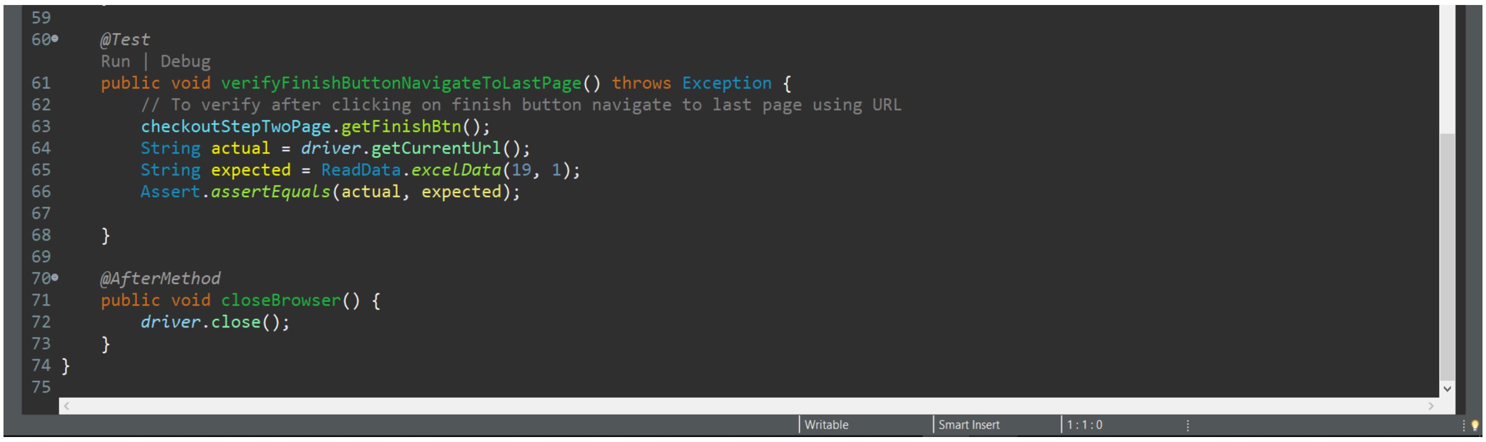
```java
package Pages;

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

import Base.TestBase;

public class CheckoutStepTwoPage extends TestBase {
    @FindBy(className = "title")
    private WebElement checkoutOverviewTitle;
    @FindBy(xpath = "//div[contains(text(),'Payment')]")
    private WebElement paymentInformation;
    @FindBy(xpath = "//div[text()='Shipping Information:']")
    private WebElement shipingInformation;
    @FindBy(xpath = "//div[contains(text(),'Total')]")
    private WebElement toatlPayment;
    @FindBy(xpath = "//div[contains(text(),'Tax')]")
    private WebElement tax;
    @FindBy(id = "finish")
    private WebElement finishBtn;

    public CheckoutStepTwoPage() {
        // Constructor to initialize driver with instance variable
        PageFactory.initElements(driver, this);
    }

    public String verifyOverviewTitle() {
        // To return Title
        return checkoutOverviewTitle.getText();
    }

    public boolean verifyPaymentInformation() {
        // To confirm payment Information is available
        return paymentInformation.isDisplayed();
    }

    public boolean verifyShipingInformation() {
        // To confirm shipping Information is available
        return shipingInformation.isDisplayed();
    }

    public boolean verifyTotalPayment() {
        // To confirm Total payment Information is available
        return toatlPayment.isDisplayed();
    }

    public boolean verifyTax() {
        // To confirm Tax Information is available
        return tax.isDisplayed();
    }

    public WebElement getFinishBtn() {
        // To click on finish button
        finishBtn.click();
        return finishBtn;
    }
}

```

Testers don't make software; they make it better.

CheckoutStepTwoPageTest Class:


eclipse-workspace - FrameworkProject/src/test/java/TestCases/CheckoutStepTwoPageTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
TestBase.java LoginPage.java LoginPageTest.java InventoryPage.ja... InventoryPageTe... *CartPage.java CartPageTest.java CheckoutStepOne... CheckoutStepOne... *CheckoutStepTw... CheckoutStepTwo...
FrameworkProject src/test/java TestCases CheckoutStepTwoPageTest

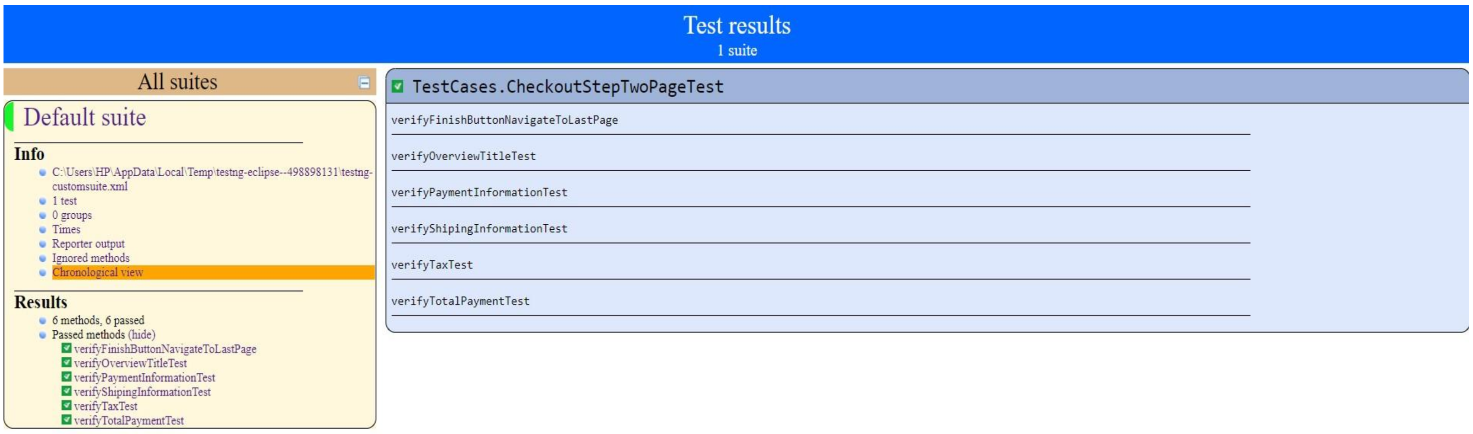
```java
package TestCases;

import org.testng.Assert;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.CheckoutStepTwoPage;
import Utility.ReadData;

Run All
public class CheckoutStepTwoPageTest extends TestBase {
    CheckoutStepTwoPage checkoutStepTwoPage;

    @BeforeMethod
    public void setup() throws Exception {
        // To call initialize method and login method from TestBase class
        initialization();
        checkoutPageTwo();
        checkoutStepTwoPage = new CheckoutStepTwoPage();

    }

    @Test
    Run | Debug
    public void verifyOverviewTitleTest() throws Exception {
        // To confirm Title
        String actual = checkoutStepTwoPage.verifyOverviewTitle();
        String expected = ReadData.excelData(16, 1);
        Assert.assertEquals(actual, expected);
    }

    @Test
    Run | Debug
    public void verifyPaymentInformationTest() {
        // To conform Payment information
        boolean actual = checkoutStepTwoPage.verifyPaymentInformation();
        Assert.assertTrue(actual);
    }

    @Test
    Run | Debug
    public void verifyShipingInformationTest() {
        // To conform shipping information
        boolean actual = checkoutStepTwoPage.verifyShipingInformation();
        Assert.assertTrue(actual);
    }

    @Test
    Run | Debug
    public void verifyTotalPaymentTest() {
        // To confirm Total payment information
        boolean actual = checkoutStepTwoPage.verifyTotalPayment();
        Assert.assertTrue(actual);
    }

    @Test
    Run | Debug
    public void verifyTaxTest() {
        // To confirm Tax information
        boolean actual = checkoutStepTwoPage.verifyTax();
        Assert.assertTrue(actual);
    }
```



```
    @Test
    Run | Debug
    public void verifyFinishButtonNavigateToLastPage() throws Exception {
        // To verify after clicking on finish button navigate to last page using URL
        checkoutStepTwoPage.getFinishBtn();
        String actual = driver.getCurrentUrl();
        String expected = ReadData.excelData(19, 1);
        Assert.assertEquals(actual, expected);

    }

    @AfterMethod
    public void closeBrowser() {
        driver.close();
    }
}
```

Writable | Smart Insert | 1 : 1 : 0

Checkout Step Two Report:



Test results
1 suite

All suites

Default suite

Info

- C:\Users\HP\AppData\Local\Temp\testng-eclipse--498898131\testng-customsuite.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 6 methods, 6 passed
- Passed methods (hide)
  - ✓ verifyFinishButtonNavigateToLastPage
  - ✓ verifyOverviewTitleTest
  - ✓ verifyPaymentInformationTest
  - ✓ verifyShipingInformationTest
  - ✓ verifyTaxTest
  - ✓ verifyTotalPaymentTest

✓ TestCases.CheckoutStepTwoPageTest

verifyFinishButtonNavigateToLastPage

verifyOverviewTitleTest

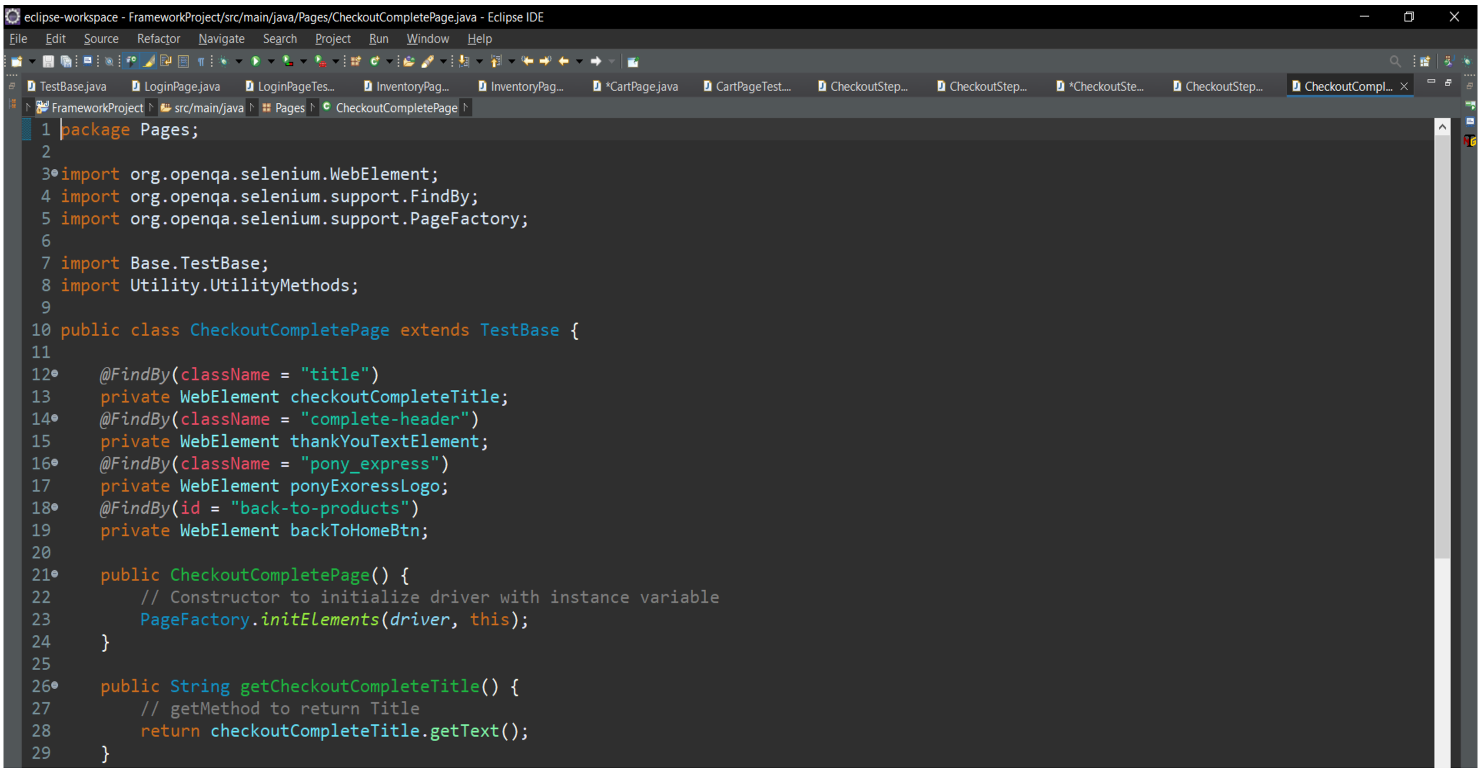
verifyPaymentInformationTest

verifyShipingInformationTest

verifyTaxTest

verifyTotalPaymentTest

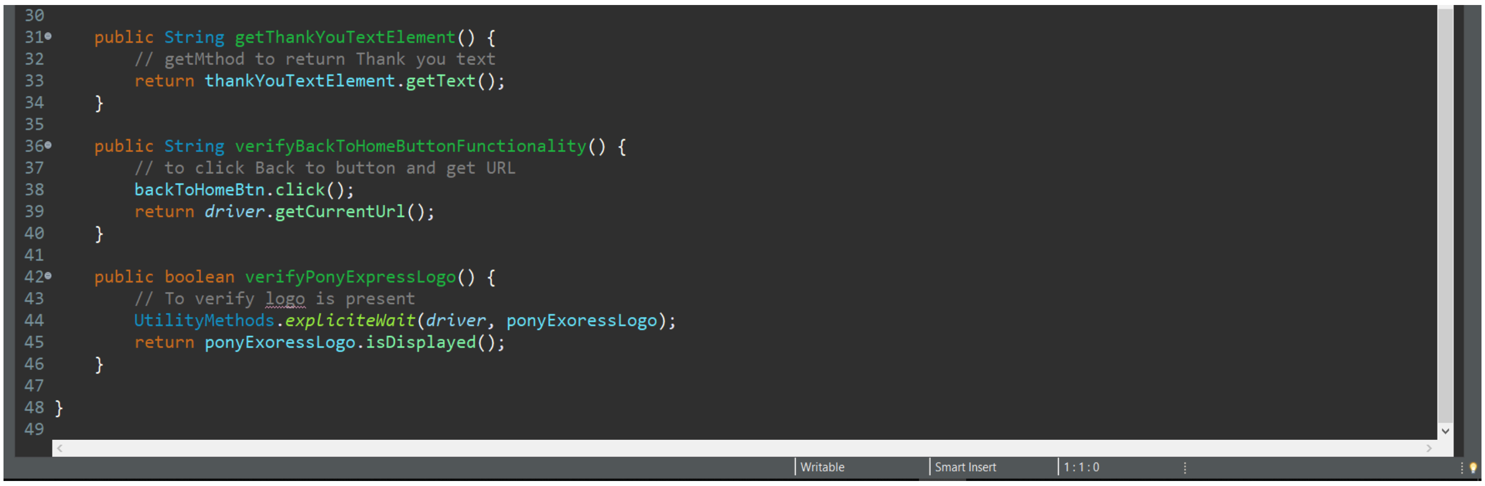
CheckoutStepCompletePage Class:



eclipse-workspace - FrameworkProject/src/main/java/Pages/CheckoutCompletePage.java - Eclipse IDE

```
package Pages;

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

import Base.TestBase;
import Utility.UtilityMethods;

public class CheckoutCompletePage extends TestBase {

    @FindBy(className = "title")
    private WebElement checkoutCompleteTitle;
    @FindBy(className = "complete-header")
    private WebElement thankYouTextElement;
    @FindBy(className = "pony_express")
    private WebElement ponyExoressLogo;
    @FindBy(id = "back-to-products")
    private WebElement backToHomeBtn;

    public CheckoutCompletePage() {
        // Constructor to initialize driver with instance variable
        PageFactory.initElements(driver, this);
    }

    public String getCheckoutCompleteTitle() {
        // getMethod to return Title
        return checkoutCompleteTitle.getText();
    }
```

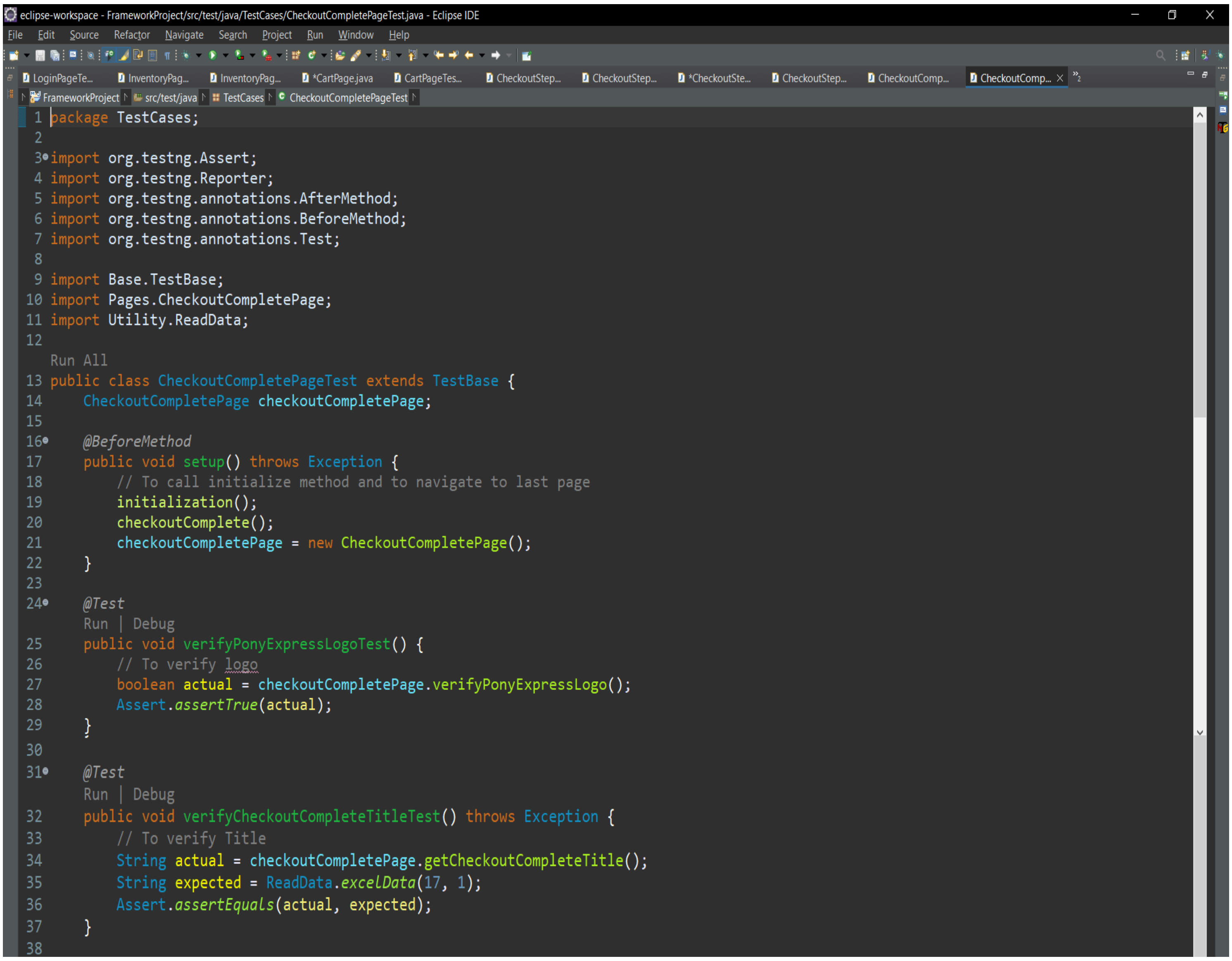
Situ



```
    public String getThankYouTextElement() {
        // getMthod to return Thank you text
        return thankYouTextElement.getText();
    }

    public String verifyBackToHomeButtonFunctionality() {
        // to click Back to button and get URL
        backToHomeBtn.click();
        return driver.getCurrentUrl();
    }

    public boolean verifyPonyExpressLogo() {
        // To verify logo is present
        UtilityMethods.expliciteWait(driver, ponyExoressLogo);
        return ponyExoressLogo.isDisplayed();
    }

}
```

Writable | Smart Insert | 1 : 1 : 0

<u>CheckoutStepCompletePageTest Class:</u>

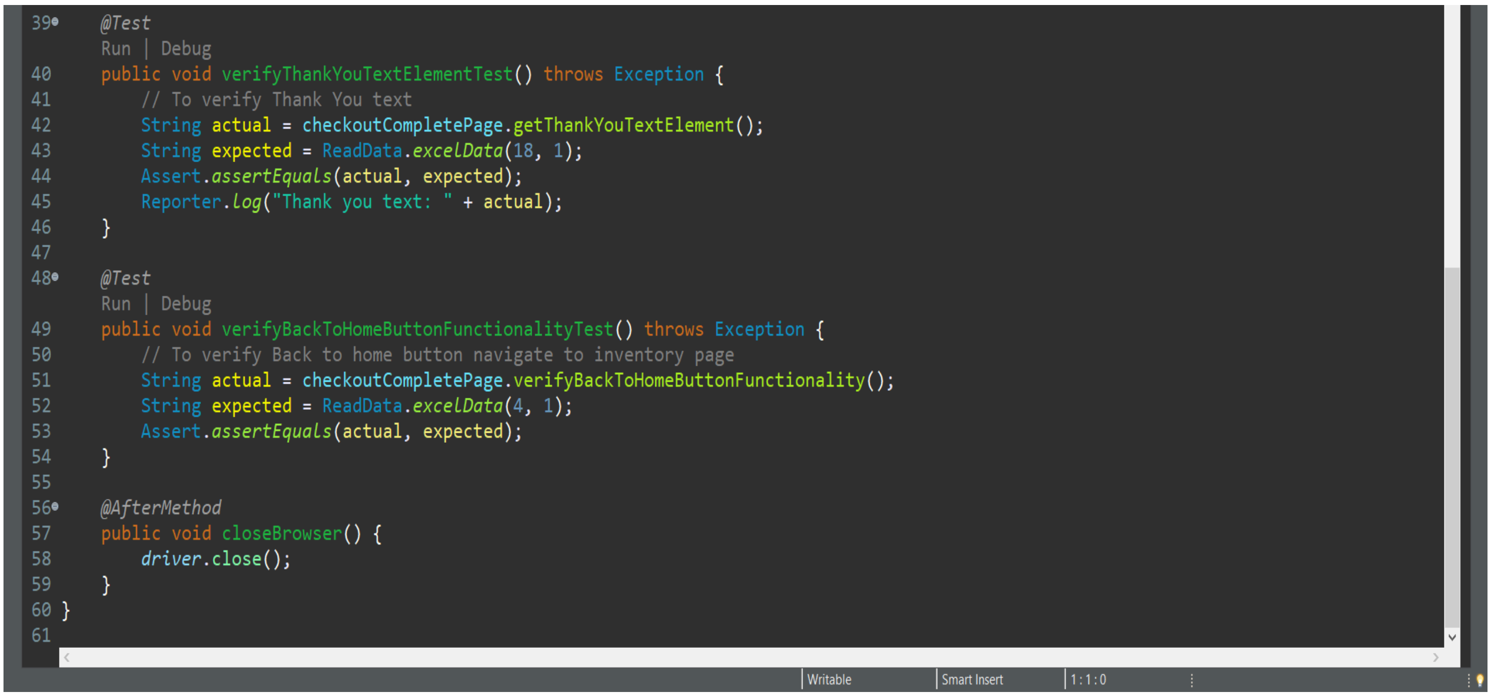


eclipse-workspace - FrameworkProject/src/test/java/TestCases/CheckoutCompletePageTest.java - Eclipse IDE

```
package TestCases;

import org.testng.Assert;
import org.testng.Reporter;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;

import Base.TestBase;
import Pages.CheckoutCompletePage;
import Utility.ReadData;

Run All
public class CheckoutCompletePageTest extends TestBase {
    CheckoutCompletePage checkoutCompletePage;

    @BeforeMethod
    public void setup() throws Exception {
        // To call initialize method and to navigate to last page
        initialization();
        checkoutComplete();
        checkoutCompletePage = new CheckoutCompletePage();
    }

    @Test
    Run | Debug
    public void verifyPonyExpressLogoTest() {
        // To verify logo
        boolean actual = checkoutCompletePage.verifyPonyExpressLogo();
        Assert.assertTrue(actual);
    }

    @Test
    Run | Debug
    public void verifyCheckoutCompleteTitleTest() throws Exception {
        // To verify Title
        String actual = checkoutCompletePage.getCheckoutCompleteTitle();
        String expected = ReadData.excelData(17, 1);
        Assert.assertEquals(actual, expected);
    }

```

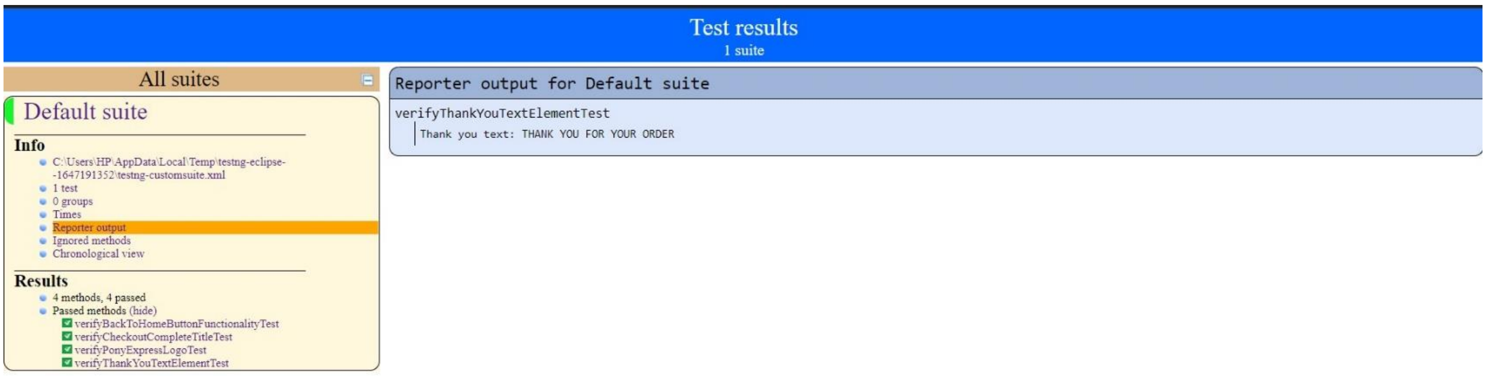
Testers don't make software; they make it better.

Situ



```
    @Test
    Run | Debug
    public void verifyThankYouTextElementTest() throws Exception {
        // To verify Thank You text
        String actual = checkoutCompletePage.getThankYouTextElement();
        String expected = ReadData.excelData(18, 1);
        Assert.assertEquals(actual, expected);
        Reporter.log("Thank you text: " + actual);
    }

    @Test
    Run | Debug
    public void verifyBackToHomeButtonFunctionalityTest() throws Exception {
        // To verify Back to home button navigate to inventory page
        String actual = checkoutCompletePage.verifyBackToHomeButtonFunctionality();
        String expected = ReadData.excelData(4, 1);
        Assert.assertEquals(actual, expected);
    }

    @AfterMethod
    public void closeBrowser() {
        driver.close();
    }
}

```

<u>Checkout Step Two Page Report:</u>



Test results
1 suite

All suites

Default suite

Info

- C:\Users\HP\AppData\Local\Temp\testng-eclipse--1647191352\testng-customsuite.xml
- 1 test
- 0 groups
- Times
- Reporter output
- Ignored methods
- Chronological view

Results

- 4 methods, 4 passed
- Passed methods (hide)
  - ✓ verifyBackToHomeButtonFunctionalityTest
  - ✓ verifyCheckoutCompleteTitleTest
  - ✓ verifyPonyExpressLogoTest
  - ✓ verifyThankYouTextElementTest

```
Reporter output for Default suite
verifyThankYouTextElementTest
    Thank you text: THANK YOU FOR YOUR ORDER
```