Software Testing

Selenium Automation Testing

Selenium Webdriver

Selenium WebDriver

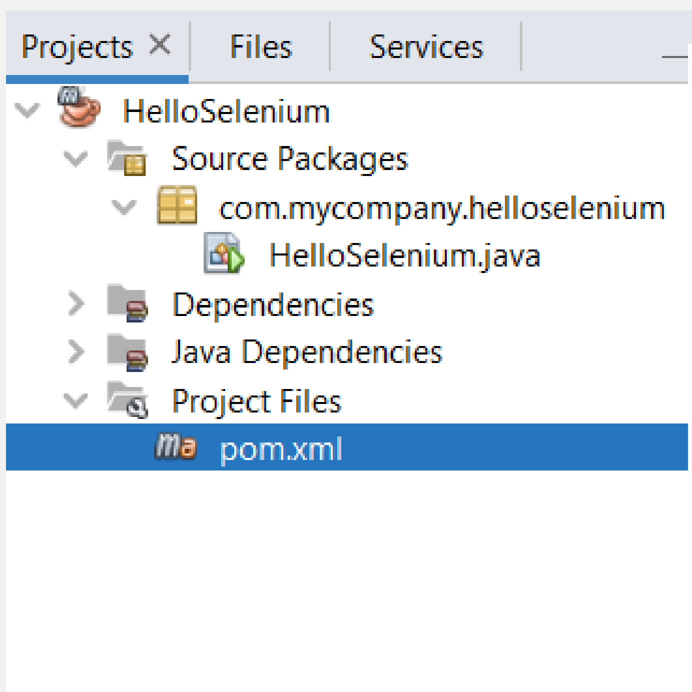
https://www.selenium.dev/documentation/webdriver/

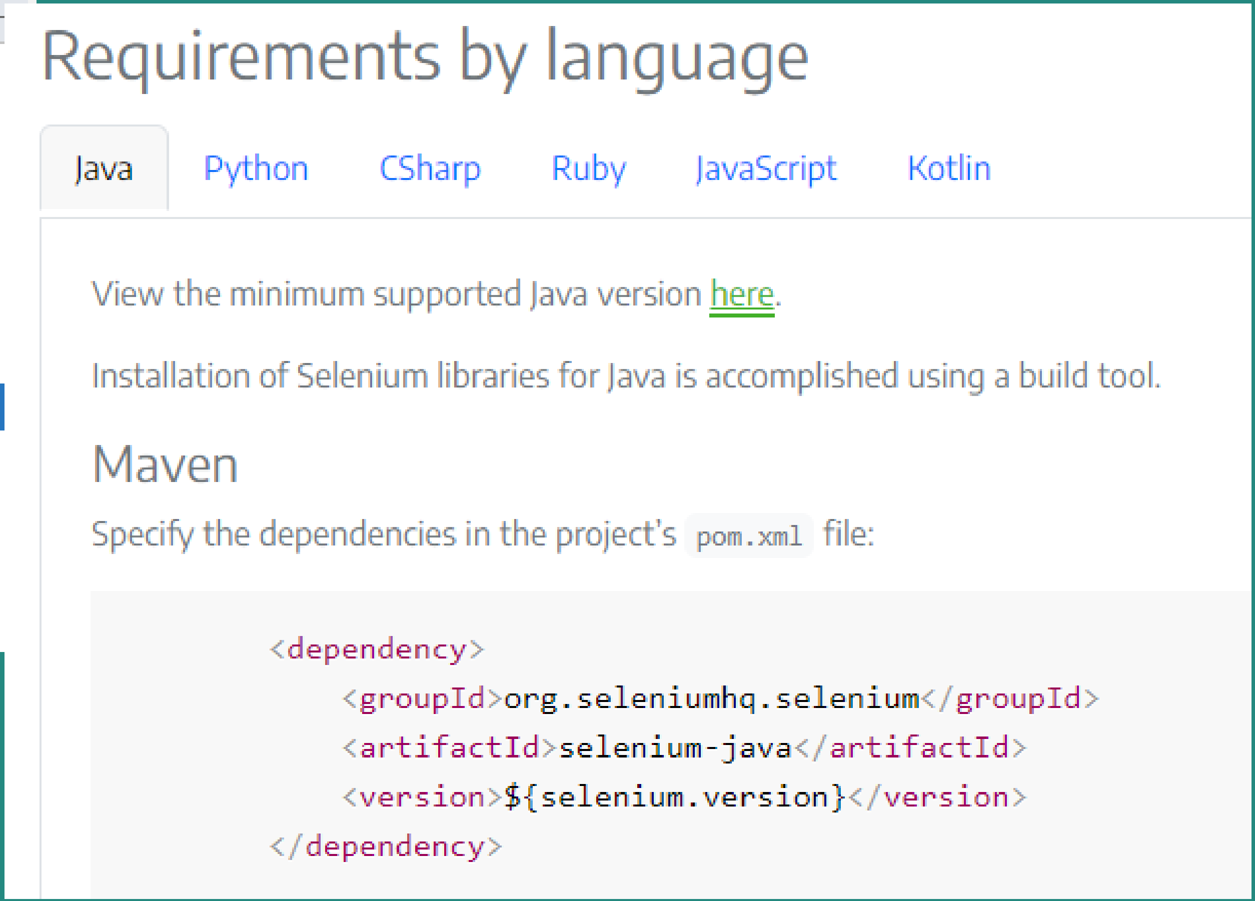
Download Apache NetBeans

- https://netbeans.apache.org/front/main/download/index.html

Install Selenium library

- https://www.selenium.dev/documentation/webdriver/getting_started/install_library/





Requirements by language

View the minimum supported Java version here.

Installation of Selenium libraries for Java is accomplished using a build tool.

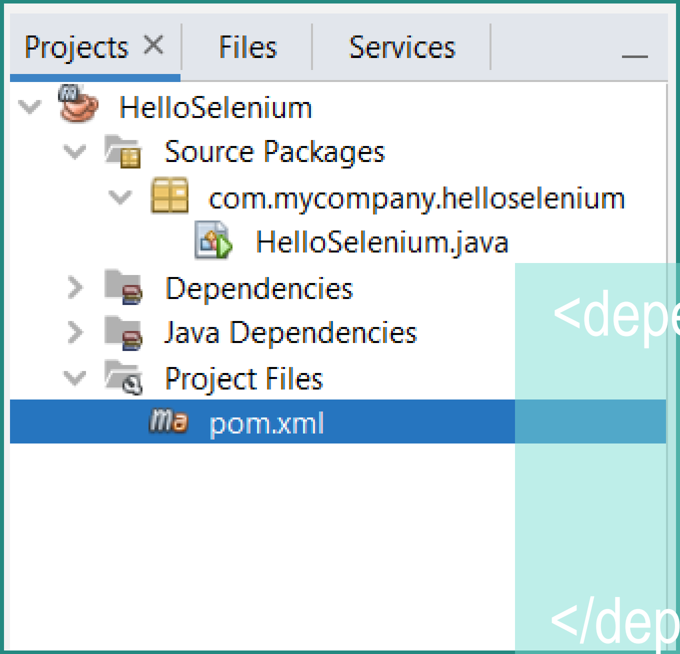
Maven

Specify the dependencies in the project's `pom.xml` file:

```
        <dependency>
            <groupId>org.seleniumhq.selenium</groupId>
            <artifactId>selenium-java</artifactId>
            <version>${selenium.version}</version>
        </dependency>
```

Install Selenium library

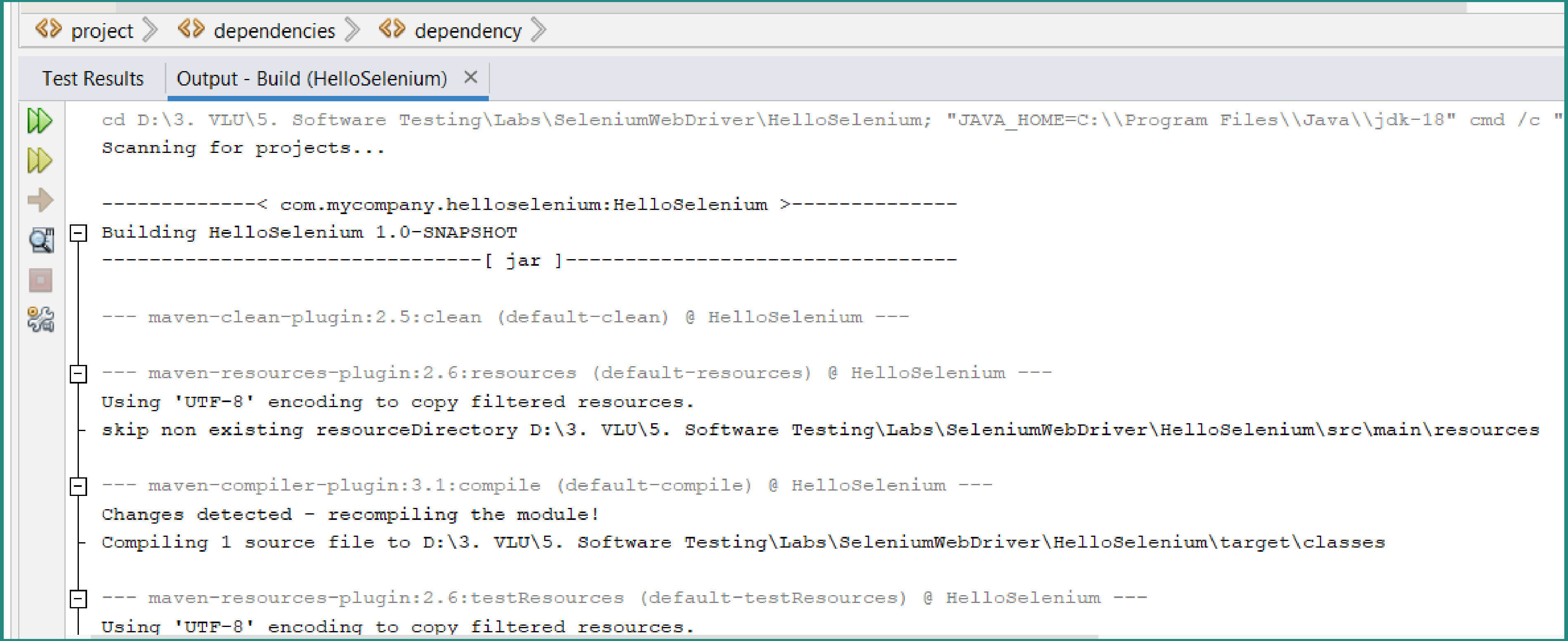
- Create Java *Maven* Project
- Specify the dependencies in the project's *pom.xml* file



```
<dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>4.18.1</version>
</dependency>
```

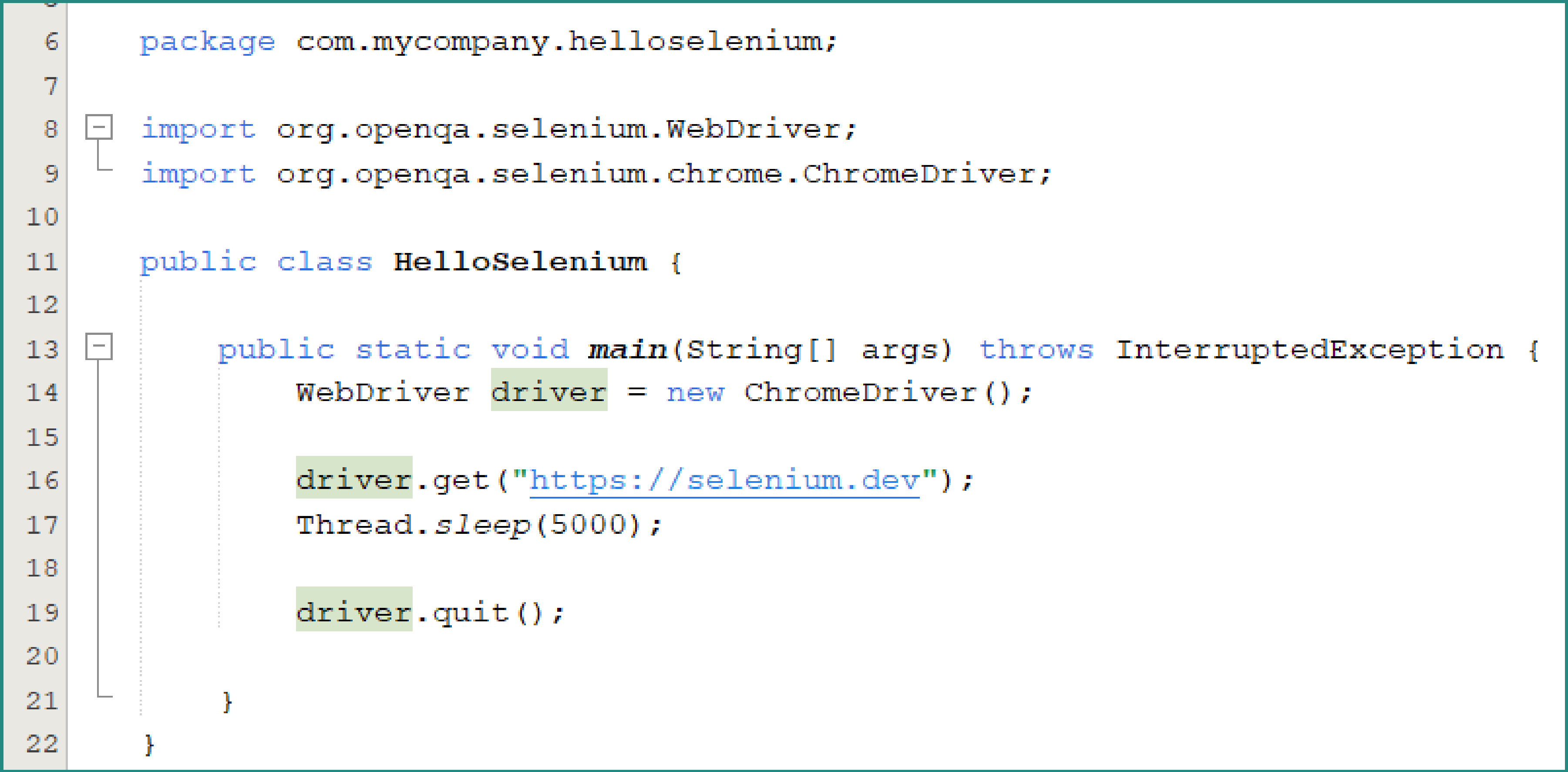
Build Project

- maven-install-plugin



```
cd D:\3. VLU\5. Software Testing\Labs\SeleniumWebDriver\HelloSelenium; "JAVA_HOME=C:\\Program Files\\Java\\jdk-18" cmd /c "
Scanning for projects...

-------------< com.mycompany.helloselenium:HelloSelenium >--------------
Building HelloSelenium 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- maven-clean-plugin:2.5:clean (default-clean) @ HelloSelenium ---

--- maven-resources-plugin:2.6:resources (default-resources) @ HelloSelenium ---
Using 'UTF-8' encoding to copy filtered resources.
skip non existing resourceDirectory D:\3. VLU\5. Software Testing\Labs\SeleniumWebDriver\HelloSelenium\src\main\resources

--- maven-compiler-plugin:3.1:compile (default-compile) @ HelloSelenium ---
Changes detected - recompiling the module!
Compiling 1 source file to D:\3. VLU\5. Software Testing\Labs\SeleniumWebDriver\HelloSelenium\target\classes

--- maven-resources-plugin:2.6:testResources (default-testResources) @ HelloSelenium ---
Using 'UTF-8' encoding to copy filtered resources.
```

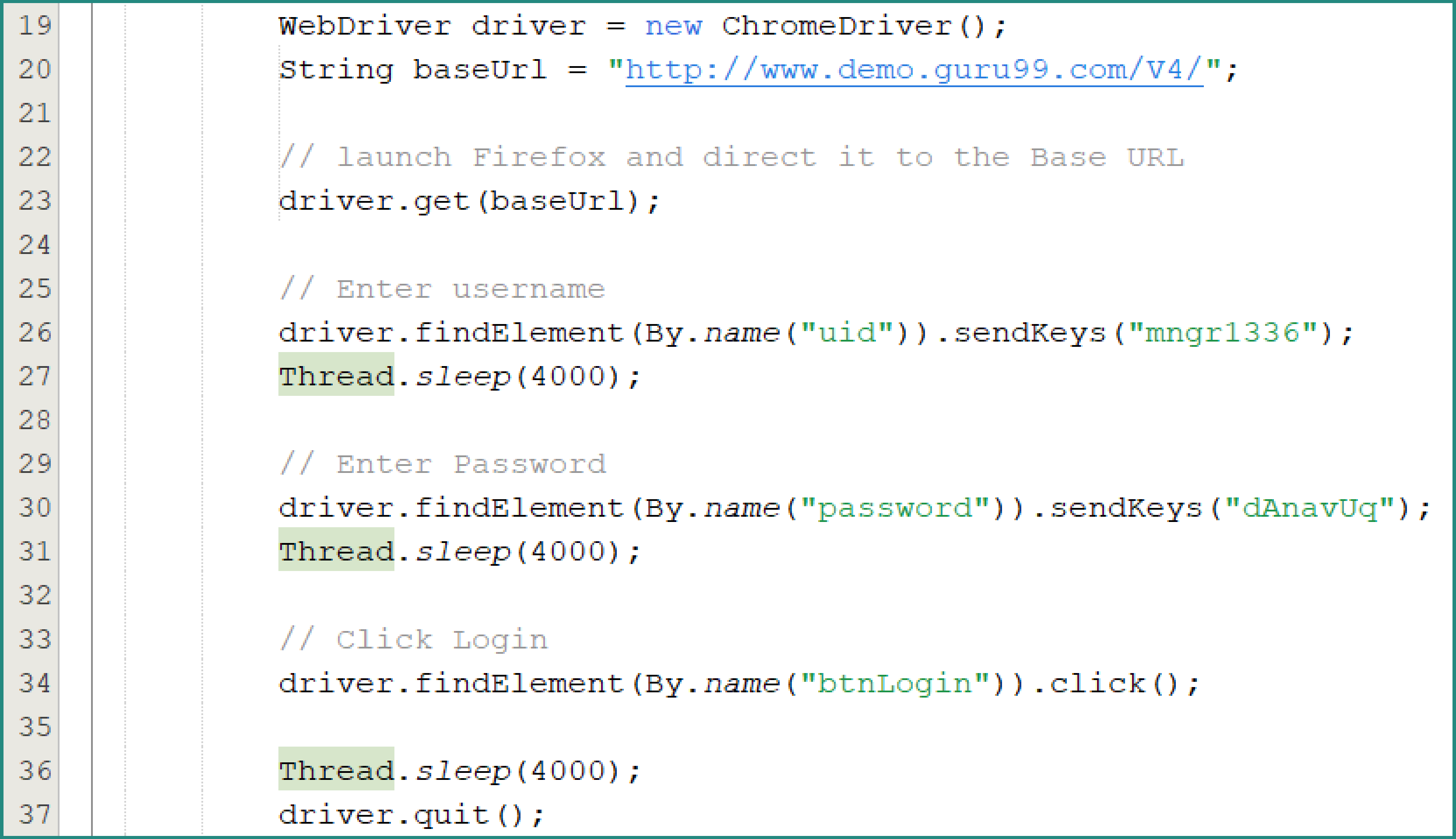
Hello Selenium



```java
package com.mycompany.helloselenium;

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

public class HelloSelenium {

    public static void main(String[] args) throws InterruptedException {
        WebDriver driver = new ChromeDriver();

        driver.get("https://selenium.dev");
        Thread.sleep(5000);

        driver.quit();

    }
}
```

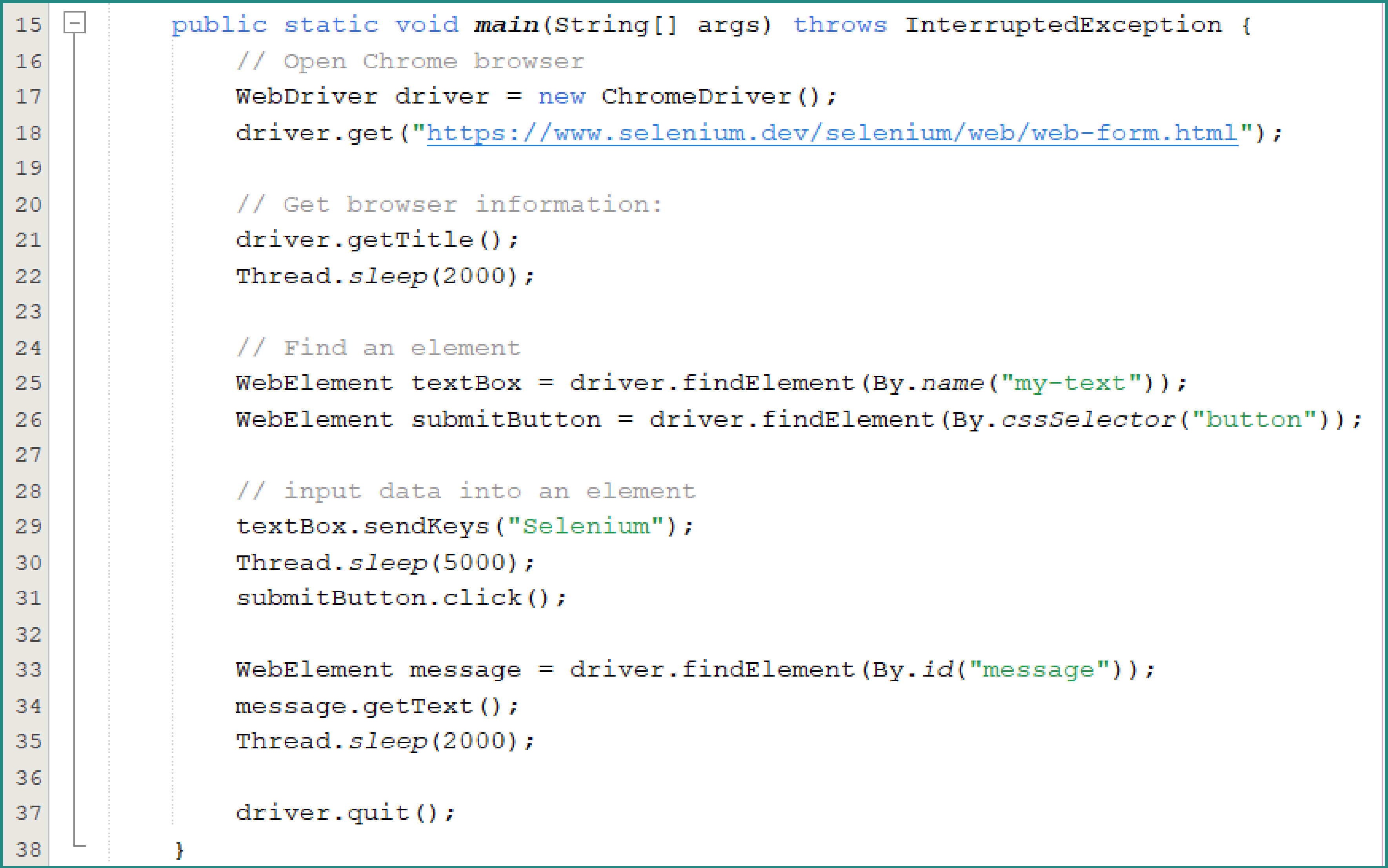
First script

- https://www.selenium.dev/documentation/webdriver/getting_started/first_script/



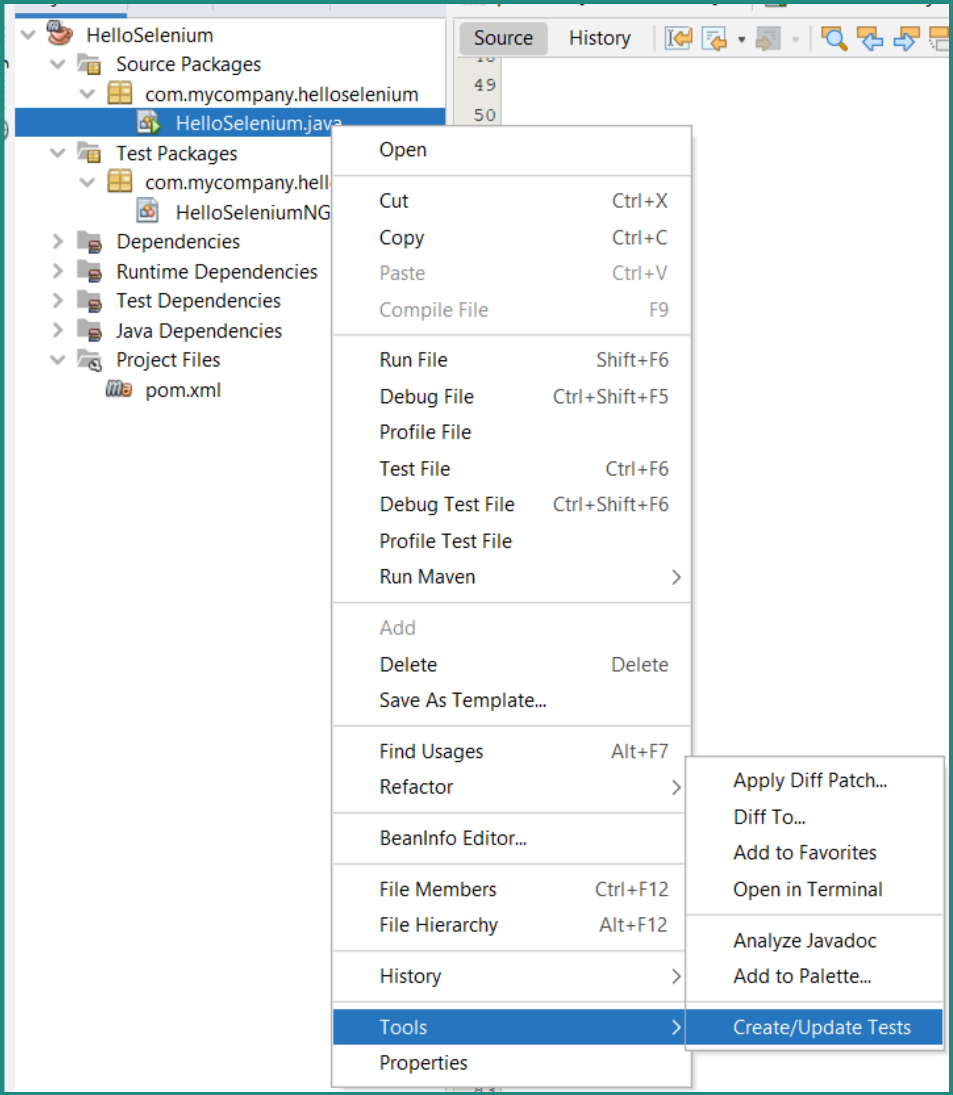
```
WebDriver driver = new ChromeDriver();
String baseUrl = "http://www.demo.guru99.com/V4/";

// launch Firefox and direct it to the Base URL
driver.get(baseUrl);

// Enter username
driver.findElement(By.name("uid")).sendKeys("mngr1336");
Thread.sleep(4000);

// Enter Password
driver.findElement(By.name("password")).sendKeys("dAnavUq");
Thread.sleep(4000);

// Click Login
driver.findElement(By.name("btnLogin")).click();

Thread.sleep(4000);
driver.quit();
```

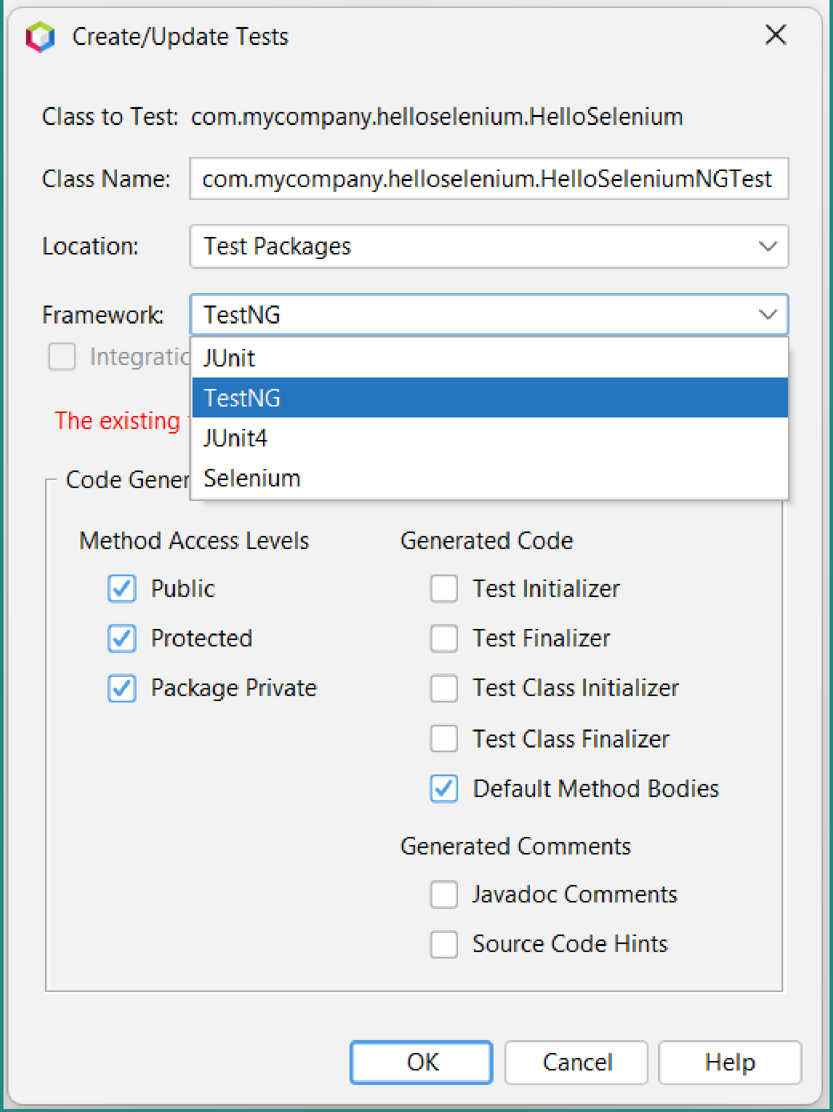
Second script



```java
    public static void main(String[] args) throws InterruptedException {
        // Open Chrome browser
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.selenium.dev/selenium/web/web-form.html");

        // Get browser information:
        driver.getTitle();
        Thread.sleep(2000);

        // Find an element
        WebElement textBox = driver.findElement(By.name("my-text"));
        WebElement submitButton = driver.findElement(By.cssSelector("button"));

        // input data into an element
        textBox.sendKeys("Selenium");
        Thread.sleep(5000);
        submitButton.click();

        WebElement message = driver.findElement(By.id("message"));
        message.getText();
        Thread.sleep(2000);

        driver.quit();
    }
```

Create TestNG




Create/Update Tests

Class to Test: com.mycompany.helloselenium.HelloSelenium

Class Name: com.mycompany.helloselenium.HelloSeleniumNGTest

Location: Test Packages

Framework: TestNG

□ Integratic

The existing

Code Gener

Method Access Levels

☑ Public

☑ Protected

☑ Package Private

Generated Code

□ Test Initializer

□ Test Finalizer

□ Test Class Initializer

□ Test Class Finalizer

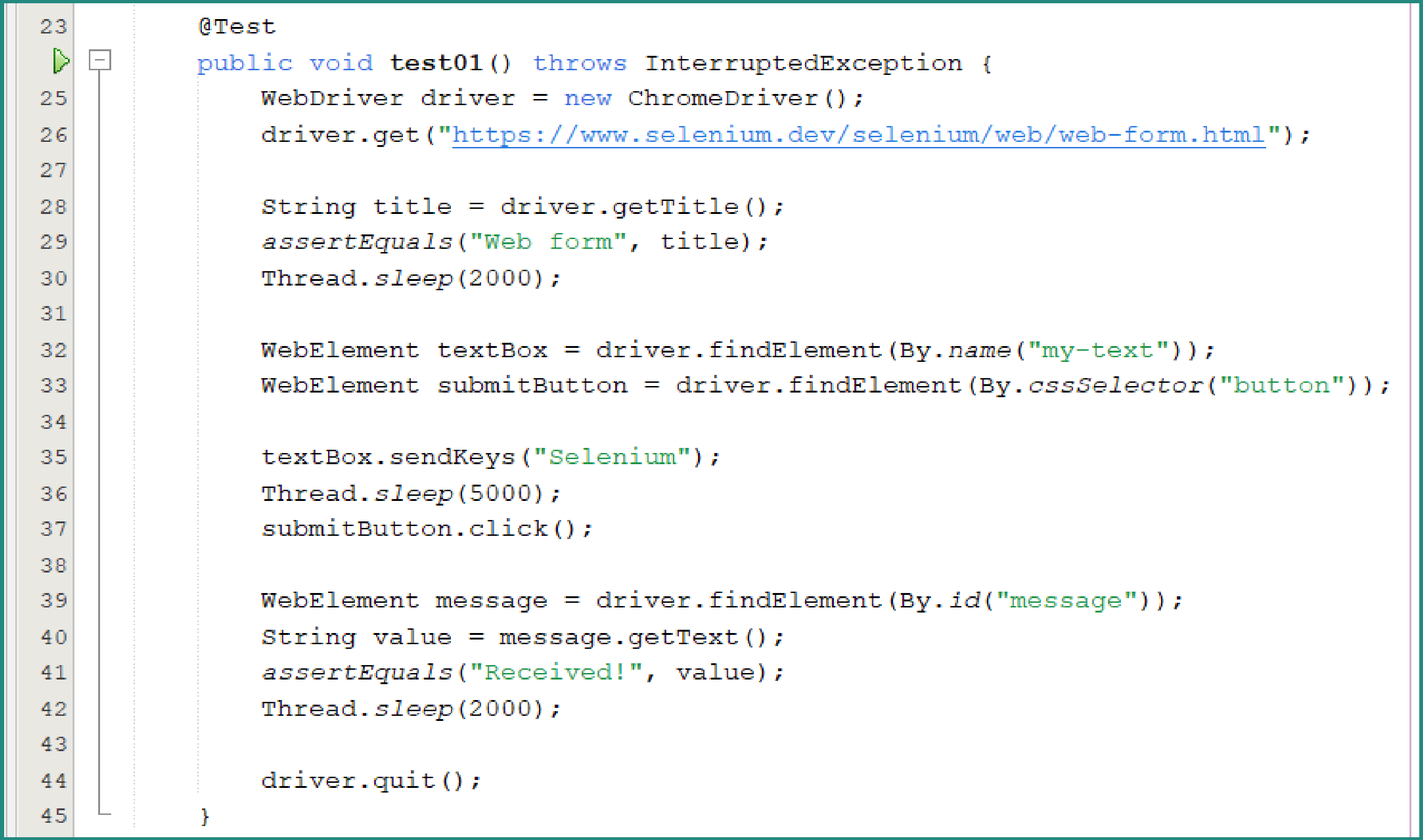
☑ Default Method Bodies

Generated Comments

□ Javadoc Comments

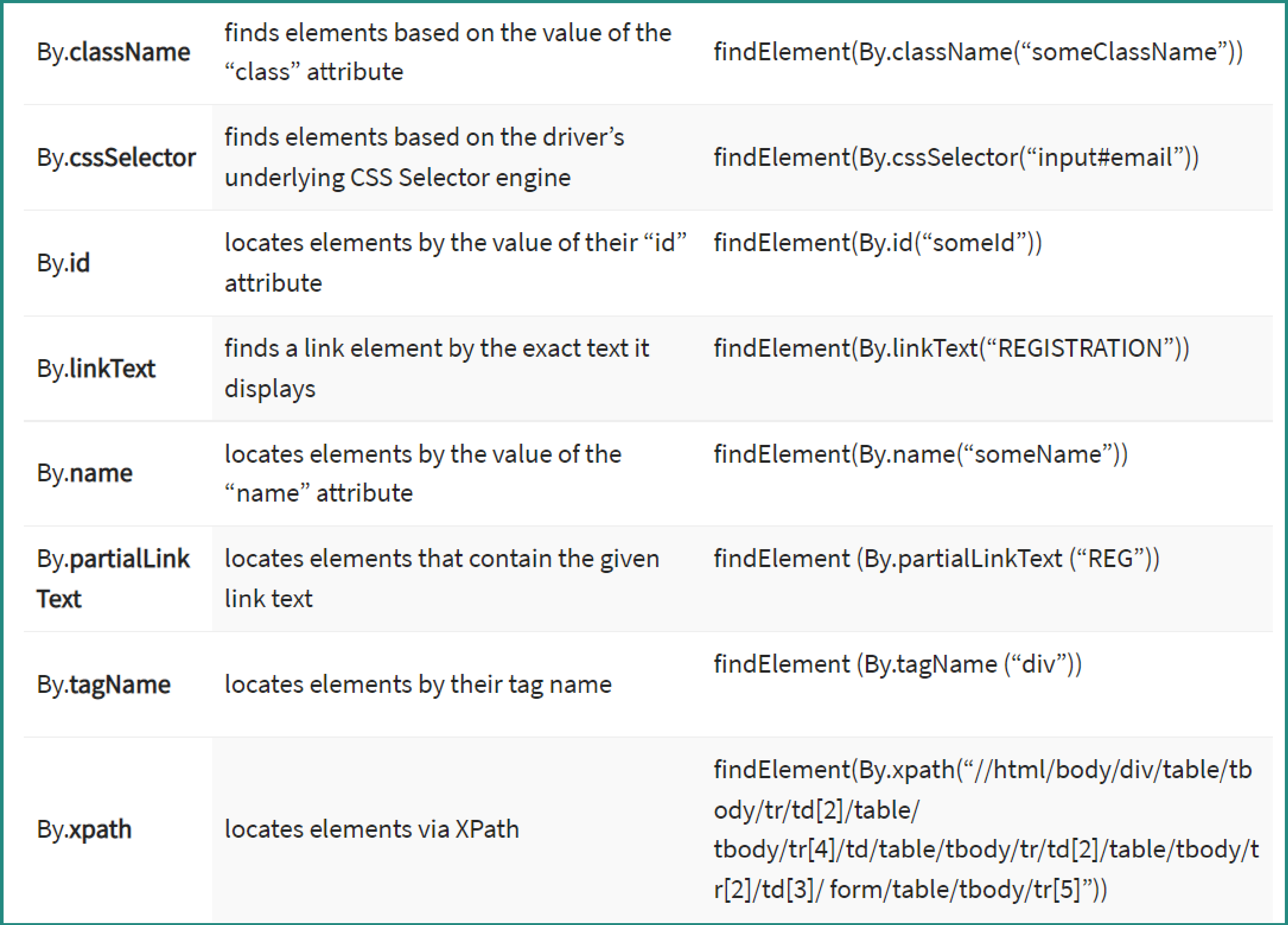
□ Source Code Hints

Selenium TestNG



```
    @Test
    public void test01() throws InterruptedException {
        WebDriver driver = new ChromeDriver();
        driver.get("https://www.selenium.dev/selenium/web/web-form.html");

        String title = driver.getTitle();
        assertEquals("Web form", title);
        Thread.sleep(2000);

        WebElement textBox = driver.findElement(By.name("my-text"));
        WebElement submitButton = driver.findElement(By.cssSelector("button"));

        textBox.sendKeys("Selenium");
        Thread.sleep(5000);
        submitButton.click();

        WebElement message = driver.findElement(By.id("message"));
        String value = message.getText();
        assertEquals("Received!", value);
        Thread.sleep(2000);

        driver.quit();
    }
```

Locating GUI Elements



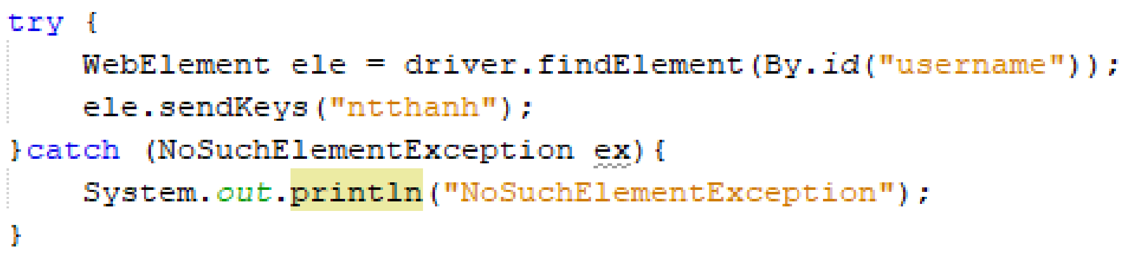
By.**className**	finds elements based on the value of the “class” attribute	findElement(By.className(“someClassName”))
By.**cssSelector**	finds elements based on the driver’s underlying CSS Selector engine	findElement(By.cssSelector(“input#email”))
By.**id**	locates elements by the value of their “id” attribute	findElement(By.id(“someId”))
By.**linkText**	finds a link element by the exact text it displays	findElement(By.linkText(“REGISTRATION”))
By.**name**	locates elements by the value of the “name” attribute	findElement(By.name(“someName”))
By.**partialLinkText**	locates elements that contain the given link text	findElement (By.partialLinkText (“REG”))
By.**tagName**	locates elements by their tag name	findElement (By.tagName (“div”))
By.**xpath**	locates elements via XPath	findElement(By.xpath(“//html/body/div/table/tbody/tr/td[2]/table/ tbody/tr[4]/td/table/tbody/tr/td[2]/table/tbody/tr[2]/td[3]/ form/table/tbody/tr[5]”))

Commands

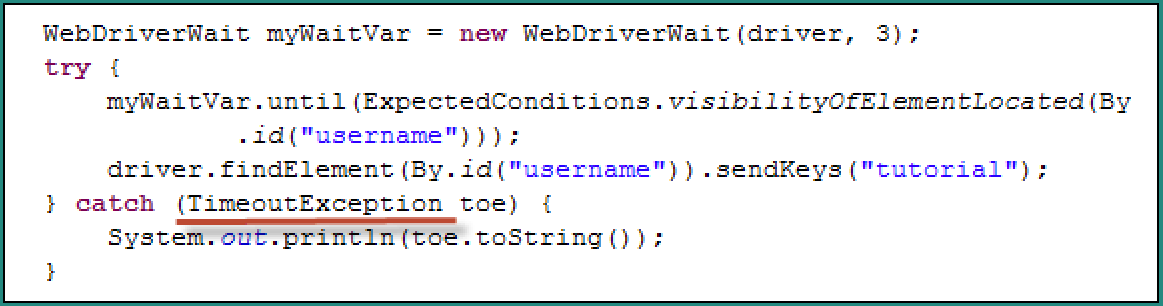
- **get(url)**
- **getTitle()**
- **getCurrentUrl()**
- **getText()**
- **navigate().forward()**
- **navigate().back()**
- **navigate().to()**
- **navigate().refresh()**
- **close()** close a single window;
- **quit()** close all windows
- **switchTo().frame()**
- **switchTo().alert()**
- **Implicit waits**
- **Explicit waits**

- isEnabled()
- isDisplayed()
- isSelected()

(if the element does not exists
WebDriver will throw
a *NoSuchElementException*)

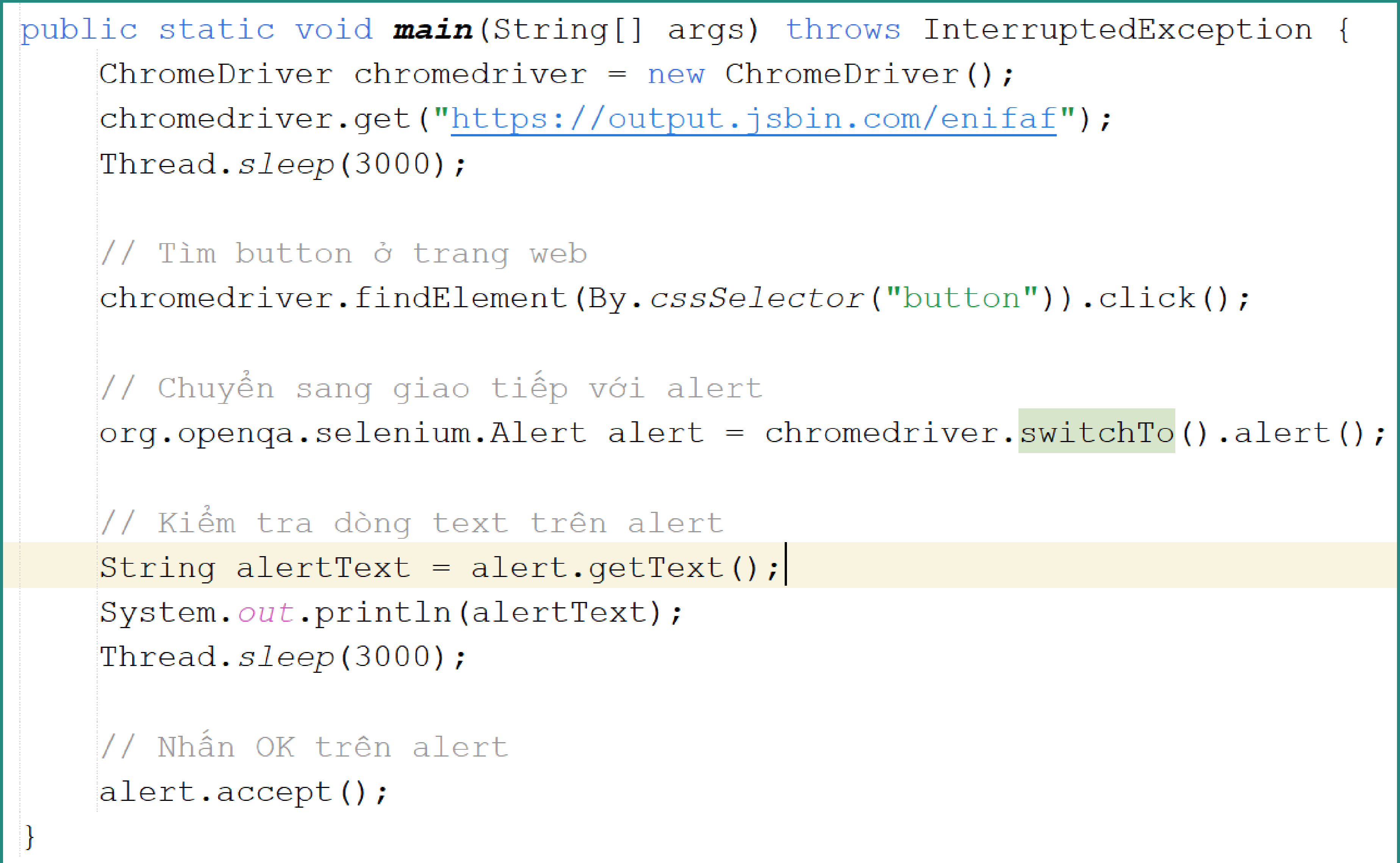


```
try {
    WebElement ele = driver.findElement(By.id("username"));
    ele.sendKeys("ntthanh");
}catch (NoSuchElementException ex){
    System.out.println("NoSuchElementException");
}
```



```
WebDriverWait myWaitVar = new WebDriverWait(driver, 3);
try {
    myWaitVar.until(ExpectedConditions.visibilityOfElementLocated(By
            .id("username")));
    driver.findElement(By.id("username")).sendKeys("tutorial");
} catch (TimeoutException toe) {
    System.out.println(toe.toString());
}
```

Alert



```java
public static void main(String[] args) throws InterruptedException {
    ChromeDriver chromedriver = new ChromeDriver();
    chromedriver.get("https://output.jsbin.com/enifaf");
    Thread.sleep(3000);

    // Tìm button ở trang web
    chromedriver.findElement(By.cssSelector("button")).click();

    // Chuyển sang giao tiếp với alert
    org.openqa.selenium.Alert alert = chromedriver.switchTo().alert();

    // Kiểm tra dòng text trên alert
    String alertText = alert.getText();
    System.out.println(alertText);
    Thread.sleep(3000);

    // Nhấn OK trên alert
    alert.accept();
}
```

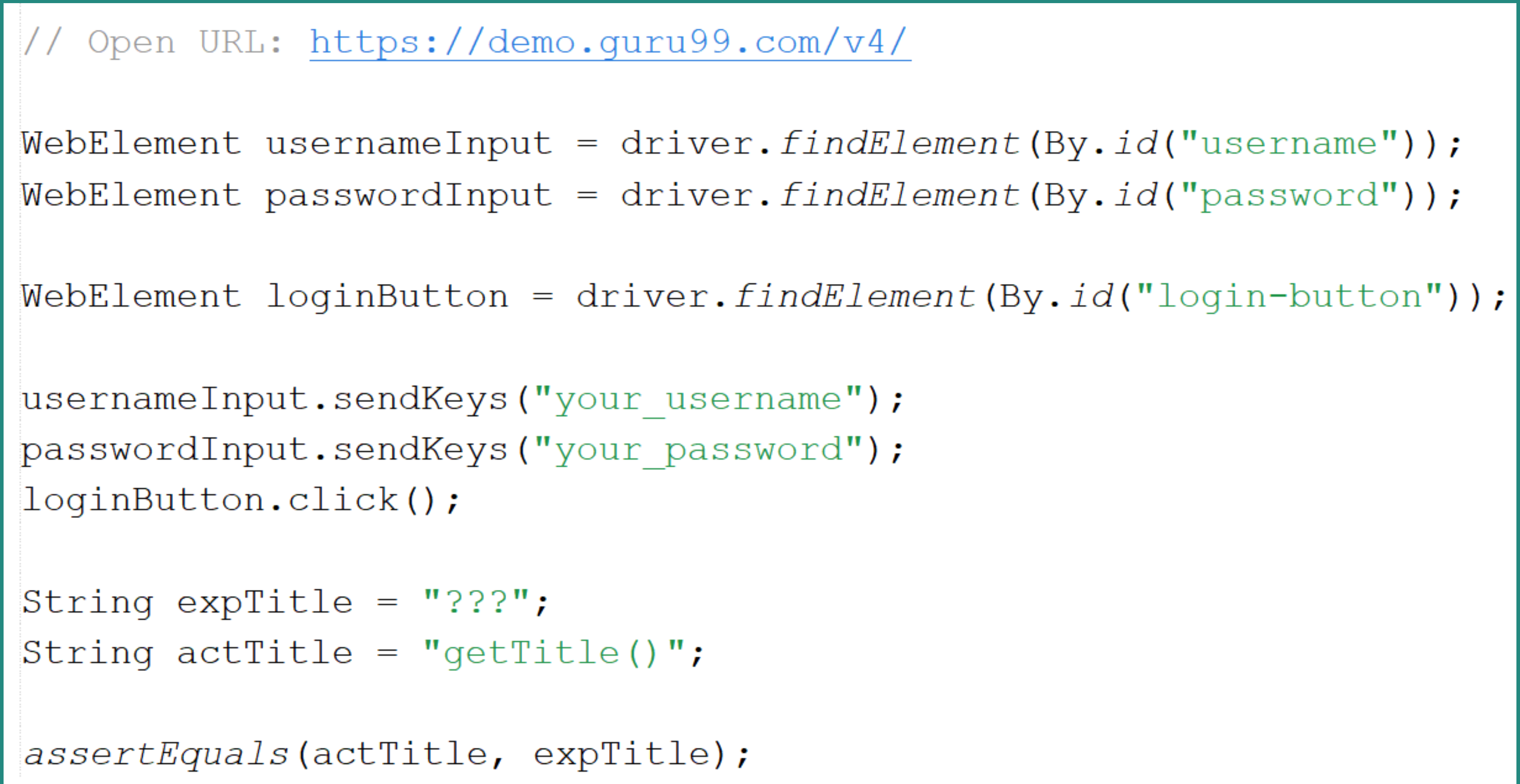
Interacting with web elements

- **click()**
- **sendKeys()**
  - types the keys into an element.
- **clear()**
  - resets the content of an element

Exercises

- Thực hành các thao tác click, send keys và clear
  - https://www.selenium.dev/selenium/web/inputs.html

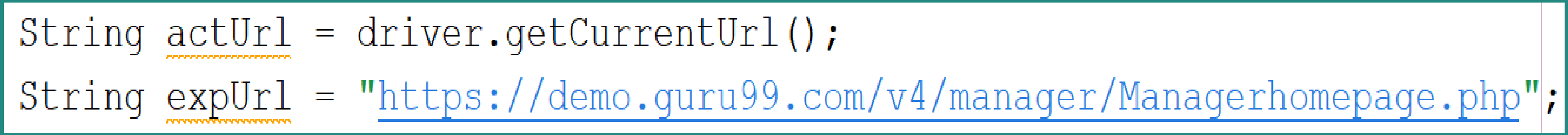
TC1. Verify the title



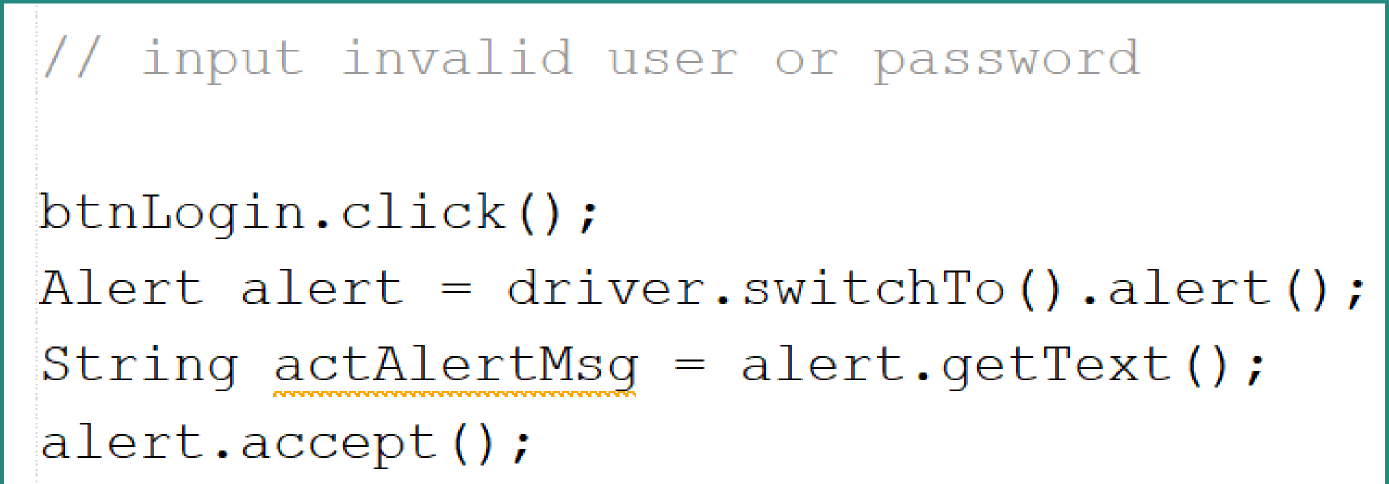
```
// Open URL: https://demo.guru99.com/v4/

WebElement usernameInput = driver.findElement(By.id("username"));
WebElement passwordInput = driver.findElement(By.id("password"));

WebElement loginButton = driver.findElement(By.id("login-button"));

usernameInput.sendKeys("your_username");
passwordInput.sendKeys("your_password");
loginButton.click();

String expTitle = "???";
String actTitle = "getTitle()";

assertEquals(actTitle, expTitle);
```

- **User: mngr563687**
- **Password: UvAnerE**

TC2. Verify Login successfully
TC3. Verify the message of alert



```
String actUrl = driver.getCurrentUrl();
String expUrl = "https://demo.guru99.com/v4/manager/Managerhomepage.php";
```



```
// input invalid user or password

btnLogin.click();
Alert alert = driver.switchTo().alert();
String actAlertMsg = alert.getText();
alert.accept();
```

Topic

- Tương tác với WebElement
  - TextBox, Button
  - CheckBox and Radio Button
  - Select Value from DropDown

  https://demo.guru99.com/test/newtours/reservation.php
- Thực hiện các thao tác
  - click
  - keyboard event (tab key, backspace key,…)
- Kiểm tra Alert và Popup

  https://demo.guru99.com/test/delete_customer.php