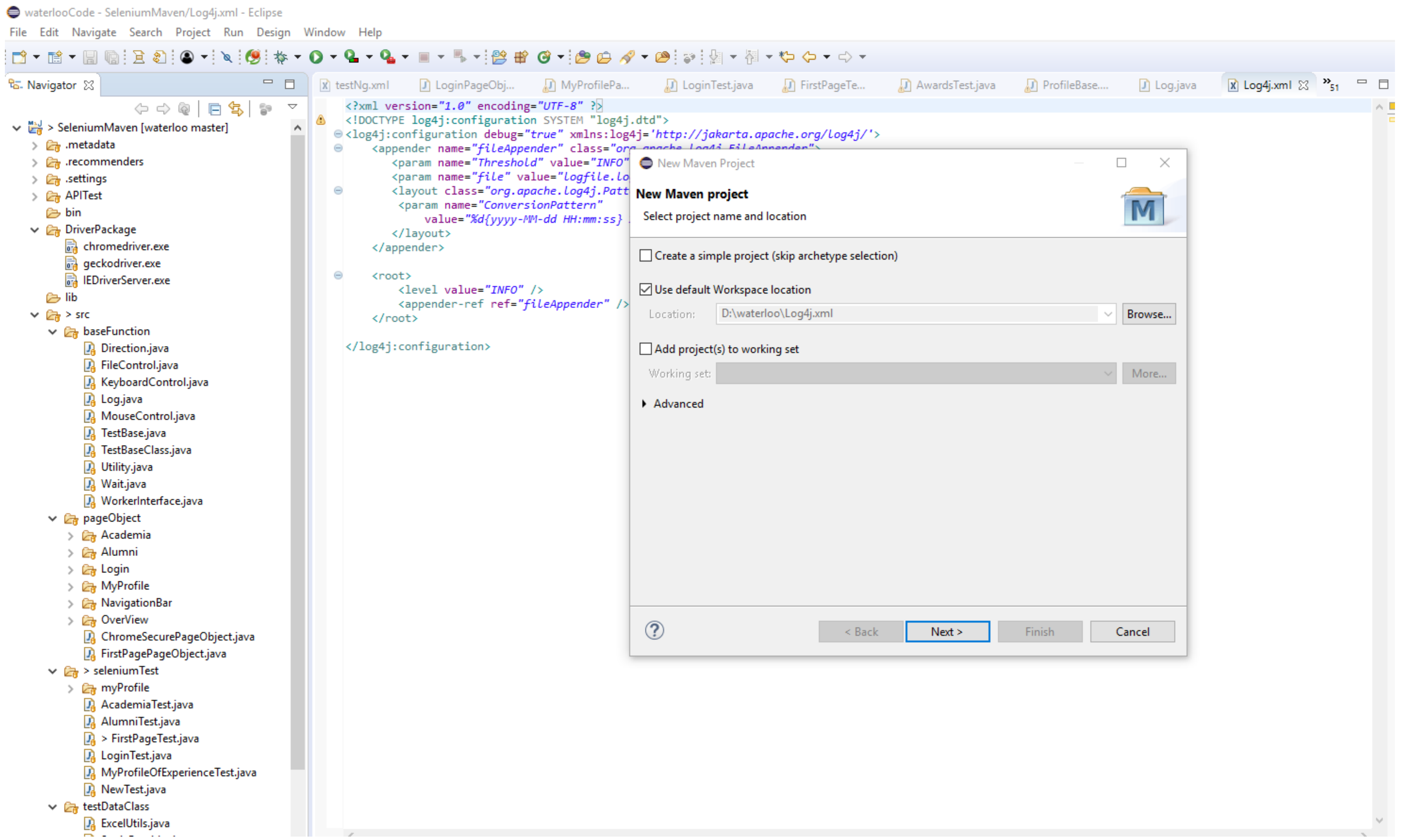
Test Framework

David LI

- Java(1.8 or above)
- Eclipse
- TestNG
  - To control test through testing.xml
- Maven
  - To introduce all needed package in POM.xml

Step 1 File-> New->Maven project



waterlooCode - SeleniumMaven/Log4j.xml - Eclipse

File Edit Navigate Search Project Run Design Window Help

Navigator

```
> SeleniumMaven [waterloo master]
    .metadata
    .recommenders
    .settings
    APITest
    bin
    DriverPackage
        chromedriver.exe
        geckodriver.exe
        IEDriverServer.exe
    lib
    > src
        baseFunction
            Direction.java
            FileControl.java
            KeyboardControl.java
            Log.java
            MouseControl.java
            TestBase.java
            TestBaseClass.java
            Utility.java
            Wait.java
            WorkerInterface.java
        pageObject
            Academia
            Alumni
            Login
            MyProfile
            NavigationBar
            OverView
            ChromeSecurePageObject.java
            FirstPagePageObject.java
        > seleniumTest
            myProfile
            AcademiaTest.java
            AlumniTest.java
            > FirstPageTest.java
            LoginTest.java
            MyProfileOfExperienceTest.java
            NewTest.java
        testDataClass
            ExcelUtils.java
```

testNg.xml | LoginPageObj... | MyProfilePa... | LoginTest.java | FirstPageTe... | AwardsTest.java | ProfileBase.... | Log.java | Log4j.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration debug="true" xmlns:log4j='http://jakarta.apache.org/log4j/'>
    <appender name="fileAppender" class="org.apache.log4j.FileAppender">
       <param name="Threshold" value="INFO"
       <param name="file" value="logfile.lo
       <layout class="org.apache.log4j.Patt
        <param name="ConversionPattern"
            value="%d{yyyy-MM-dd HH:mm:ss}
       </layout>
    </appender>

    <root>
        <level value="INFO" />
        <appender-ref ref="fileAppender" />
    </root>

</log4j:configuration>
```

New Maven Project

New Maven project

Select project name and location

[ ] Create a simple project (skip archetype selection)

[✓] Use default Workspace location

Location: D:\waterloo\Log4j.xml Browse...

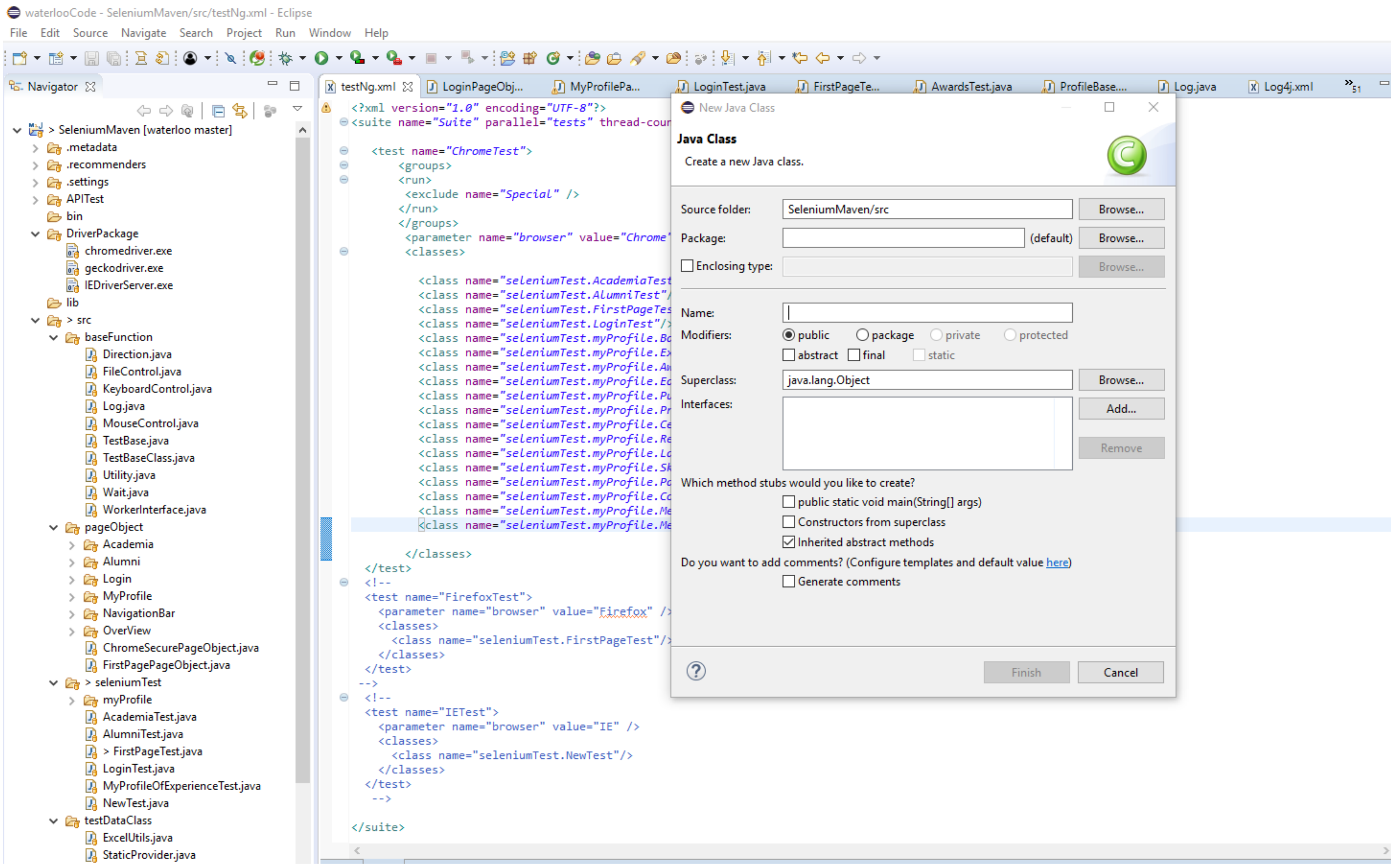
[ ] Add project(s) to working set

Working set: More...

Advanced

< Back | Next > | Finish | Cancel

Step 2 Create test PageObject



```
> SeleniumMaven [waterloo master]
  .metadata
  .recommenders
  .settings
  APITest
  bin
  DriverPackage
    chromedriver.exe
    geckodriver.exe
    IEDriverServer.exe
  lib
  > src
    baseFunction
      Direction.java
      FileControl.java
      KeyboardControl.java
      Log.java
      MouseControl.java
      TestBase.java
      TestBaseClass.java
      Utility.java
      Wait.java
      WorkerInterface.java
    pageObject
      Academia
      Alumni
      Login
      MyProfile
      NavigationBar
      OverView
      ChromeSecurePageObject.java
      FirstPagePageObject.java
    > seleniumTest
      myProfile
      AcademiaTest.java
      AlumniTest.java
      > FirstPageTest.java
      LoginTest.java
      MyProfileOfExperienceTest.java
      NewTest.java
    testDataClass
      ExcelUtils.java
      StaticProvider.java
```

```
<?xml version="1.0" encoding="UTF-8"?>
<suite name="Suite" parallel="tests" thread-cour

   <test name="ChromeTest">
       <groups>
       <run>
        <exclude name="Special" />
       </run>
       </groups>
        <parameter name="browser" value="Chrome'
        <classes>

          <class name="seleniumTest.AcademiaTest
          <class name="seleniumTest.AlumniTest"/
          <class name="seleniumTest.FirstPageTes
          <class name="seleniumTest.LoginTest"/>
          <class name="seleniumTest.myProfile.Bo
          <class name="seleniumTest.myProfile.Ex
          <class name="seleniumTest.myProfile.Aw
          <class name="seleniumTest.myProfile.Ed
          <class name="seleniumTest.myProfile.Pu
          <class name="seleniumTest.myProfile.Pr
          <class name="seleniumTest.myProfile.Ce
          <class name="seleniumTest.myProfile.Re
          <class name="seleniumTest.myProfile.Lo
          <class name="seleniumTest.myProfile.Sk
          <class name="seleniumTest.myProfile.Po
          <class name="seleniumTest.myProfile.Co
          <class name="seleniumTest.myProfile.Me
          <class name="seleniumTest.myProfile.Me

        </classes>
  </test>
  <!--
  <test name="FirefoxTest">
    <parameter name="browser" value="Firefox" />
    <classes>
      <class name="seleniumTest.FirstPageTest"/>
    </classes>
  </test>
 -->
  <!--
  <test name="IETest">
    <parameter name="browser" value="IE" />
    <classes>
      <class name="seleniumTest.NewTest"/>
    </classes>
  </test>
   -->

</suite>
```

New Java Class

Java Class

Create a new Java class.

Source folder: SeleniumMaven/src [Browse...]

Package: (default) [Browse...]

☐ Enclosing type: [Browse...]

Name:

Modifiers: [✓ public] package private protected

☐ abstract ☐ final ☐ static

Superclass: java.lang.Object [Browse...]

Interfaces: [Add...] [Remove]

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

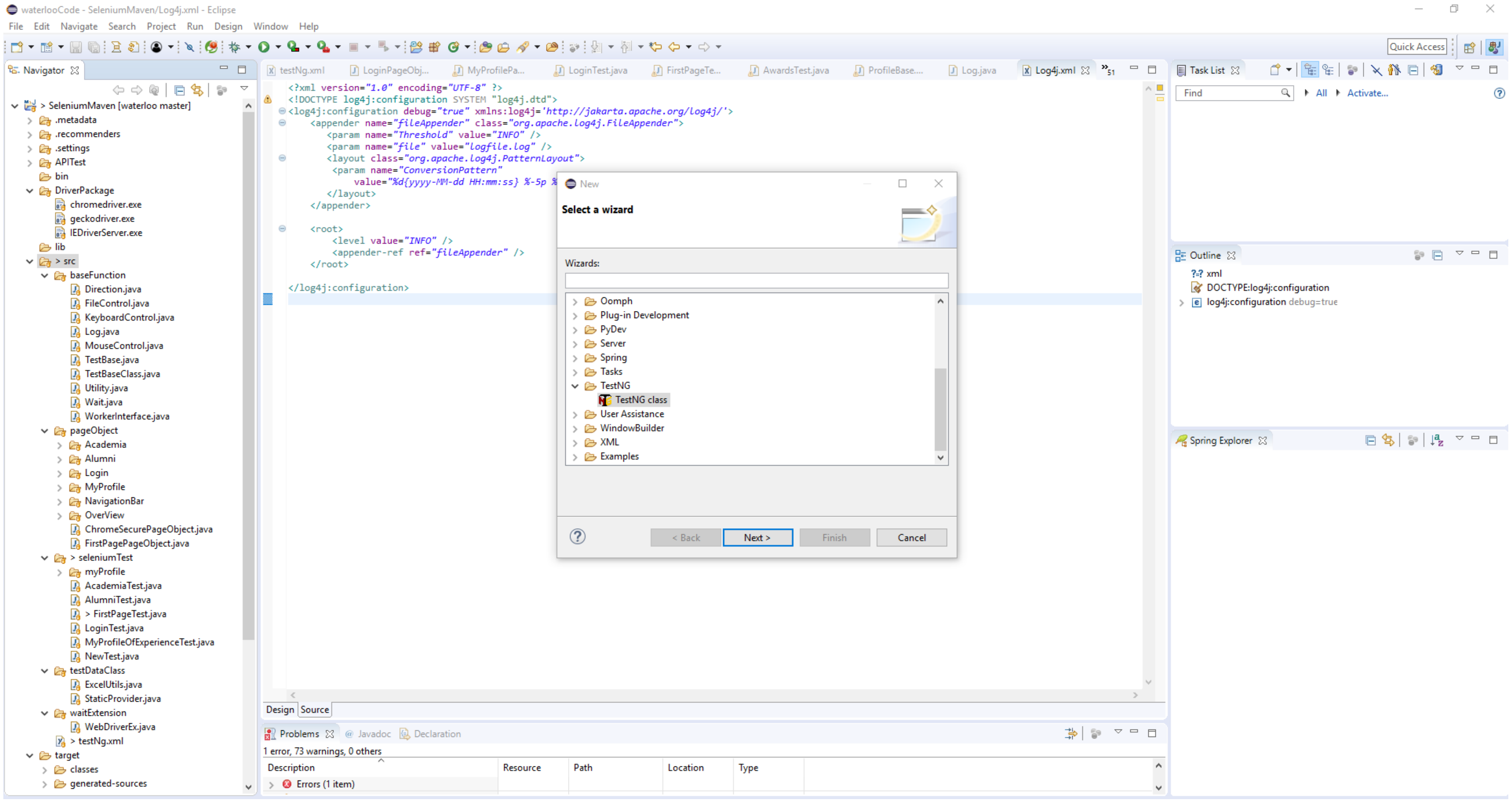
☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

☐ Generate comments

[Finish] [Cancel]

Step 3 Create testing class



waterlooCode - SeleniumMaven/Log4j.xml - Eclipse

File Edit Navigate Search Project Run Design Window Help

Navigator

```
> SeleniumMaven [waterloo master]
  .metadata
  .recommenders
  .settings
  APITest
  bin
  DriverPackage
    chromedriver.exe
    geckodriver.exe
    IEDriverServer.exe
  lib
  > src
    baseFunction
      Direction.java
      FileControl.java
      KeyboardControl.java
      Log.java
      MouseControl.java
      TestBase.java
      TestBaseClass.java
      Utility.java
      Wait.java
      WorkerInterface.java
    pageObject
      Academia
      Alumni
      Login
      MyProfile
      NavigationBar
      OverView
      ChromeSecurePageObject.java
      FirstPagePageObject.java
    > seleniumTest
      myProfile
      AcademiaTest.java
      AlumniTest.java
      > FirstPageTest.java
      LoginTest.java
      MyProfileOfExperienceTest.java
      NewTest.java
    testDataClass
      ExcelUtils.java
      StaticProvider.java
    waitExtension
      WebDriverEx.java
    > testNg.xml
  target
    classes
    generated-sources
```

testNg.xml | LoginPageObj... | MyProfilePa... | LoginTest.java | FirstPageTe... | AwardsTest.java | ProfileBase... | Log.java | Log4j.xml

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration debug="true" xmlns:log4j='http://jakarta.apache.org/log4j/'>
    <appender name="fileAppender" class="org.apache.log4j.FileAppender">
       <param name="Threshold" value="INFO" />
       <param name="file" value="logfile.log" />
       <layout class="org.apache.log4j.PatternLayout">
        <param name="ConversionPattern"
            value="%d{yyyy-MM-dd HH:mm:ss} %-5p %
       </layout>
    </appender>

    <root>
        <level value="INFO" />
        <appender-ref ref="fileAppender" />
    </root>

</log4j:configuration>
```

Design Source

New

Select a wizard

Wizards:

```
> Oomph
> Plug-in Development
> PyDev
> Server
> Spring
> Tasks
v TestNG
    TestNG class
> User Assistance
> WindowBuilder
> XML
> Examples
```

< Back | Next > | Finish | Cancel

Task List — Find — All — Activate...

Outline

```
?-? xml
DOCTYPE:log4j:configuration
> log4j:configuration debug=true
```

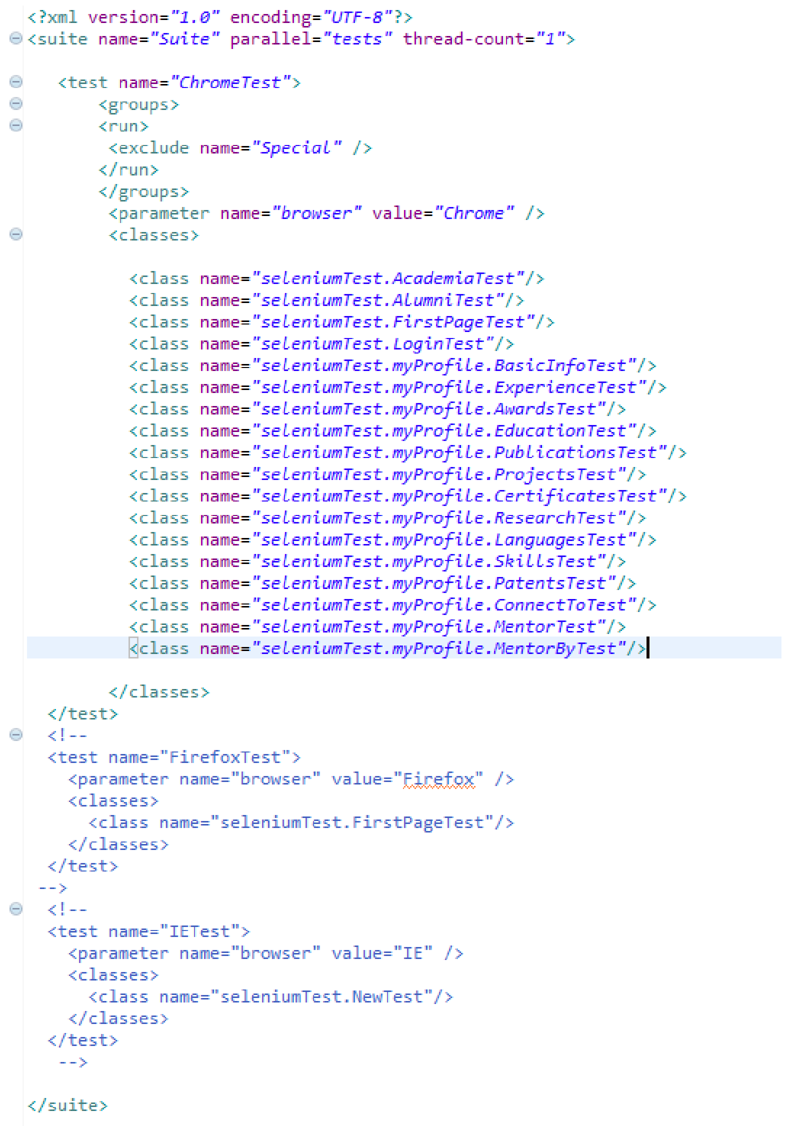
Spring Explorer

Problems | Javadoc | Declaration

1 error, 73 warnings, 0 others

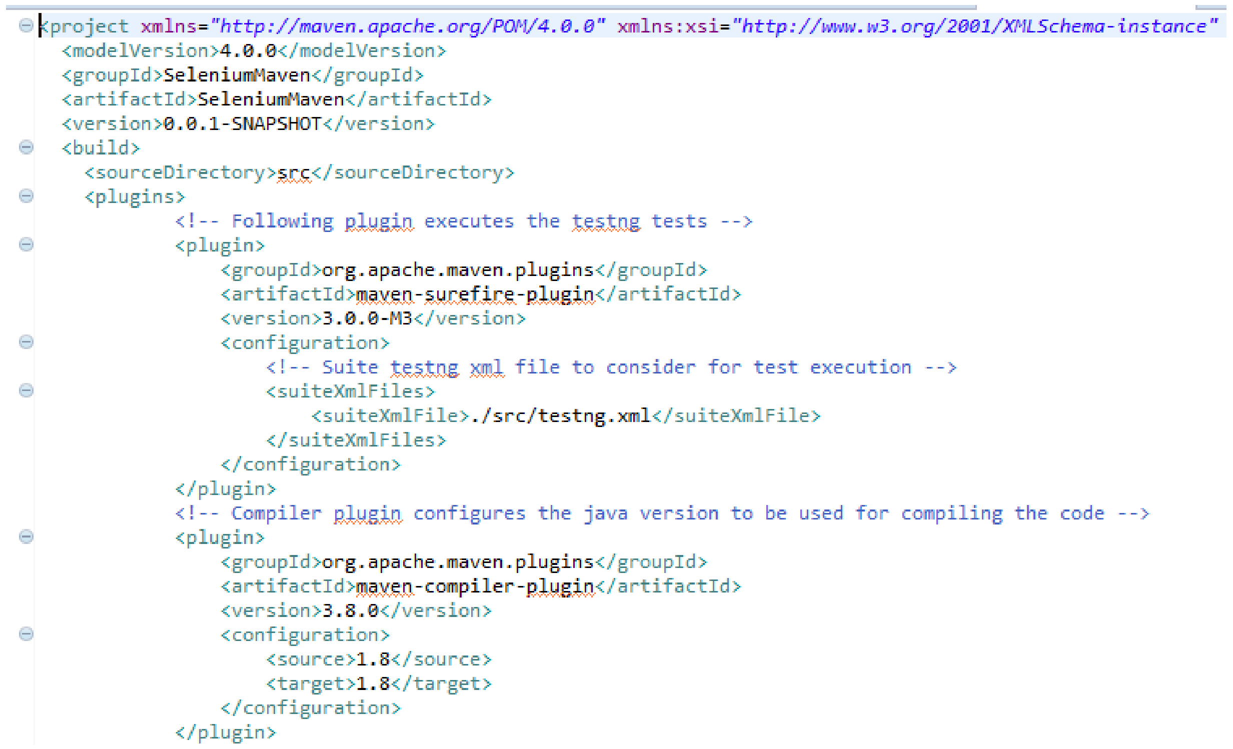
Description	Resource	Path	Location	Type
Errors (1 item)				

Step 4 Configure testing.xml

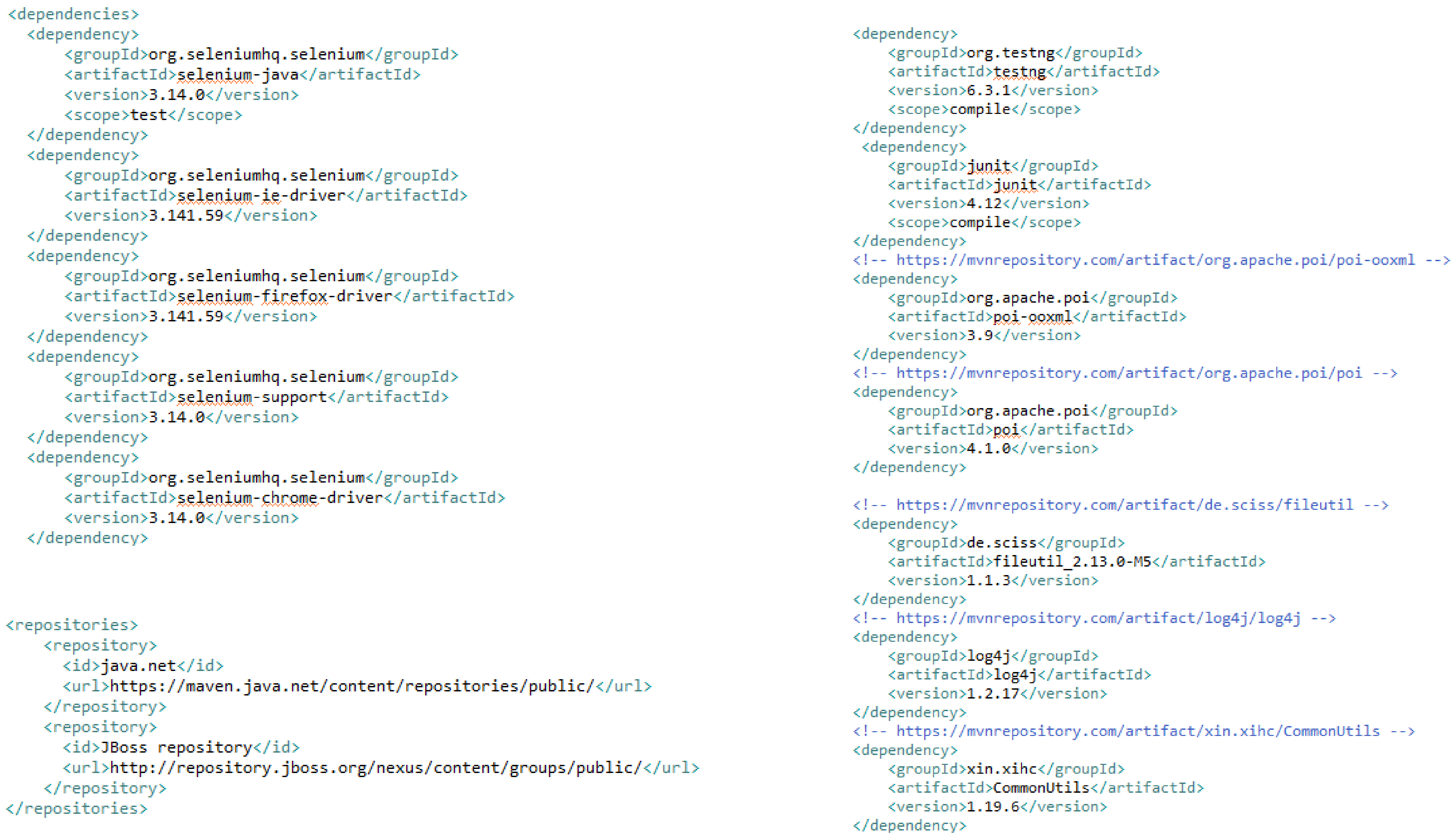


```xml
<?xml version="1.0" encoding="UTF-8"?>
<suite name="Suite" parallel="tests" thread-count="1">

   <test name="ChromeTest">
       <groups>
       <run>
        <exclude name="Special" />
       </run>
       </groups>
        <parameter name="browser" value="Chrome" />
        <classes>

          <class name="seleniumTest.AcademiaTest"/>
          <class name="seleniumTest.AlumniTest"/>
          <class name="seleniumTest.FirstPageTest"/>
          <class name="seleniumTest.LoginTest"/>
          <class name="seleniumTest.myProfile.BasicInfoTest"/>
          <class name="seleniumTest.myProfile.ExperienceTest"/>
          <class name="seleniumTest.myProfile.AwardsTest"/>
          <class name="seleniumTest.myProfile.EducationTest"/>
          <class name="seleniumTest.myProfile.PublicationsTest"/>
          <class name="seleniumTest.myProfile.ProjectsTest"/>
          <class name="seleniumTest.myProfile.CertificatesTest"/>
          <class name="seleniumTest.myProfile.ResearchTest"/>
          <class name="seleniumTest.myProfile.LanguagesTest"/>
          <class name="seleniumTest.myProfile.SkillsTest"/>
          <class name="seleniumTest.myProfile.PatentsTest"/>
          <class name="seleniumTest.myProfile.ConnectToTest"/>
          <class name="seleniumTest.myProfile.MentorTest"/>
          <class name="seleniumTest.myProfile.MentorByTest"/>

        </classes>
  </test>
  <!--
  <test name="FirefoxTest">
    <parameter name="browser" value="Firefox" />
    <classes>
      <class name="seleniumTest.FirstPageTest"/>
    </classes>
  </test>
 -->
  <!--
  <test name="IETest">
    <parameter name="browser" value="IE" />
    <classes>
      <class name="seleniumTest.NewTest"/>
    </classes>
  </test>
   -->

</suite>
```

Step 5 Configure POM.xml



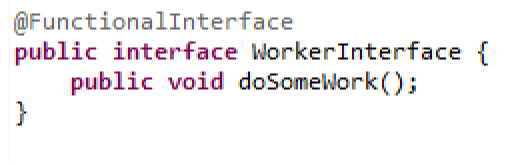
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  <modelVersion>4.0.0</modelVersion>
  <groupId>SeleniumMaven</groupId>
  <artifactId>SeleniumMaven</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <build>
    <sourceDirectory>src</sourceDirectory>
    <plugins>
            <!-- Following plugin executes the testng tests -->
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-surefire-plugin</artifactId>
                <version>3.0.0-M3</version>
                <configuration>
                    <!-- Suite testng xml file to consider for test execution -->
                    <suiteXmlFiles>
                        <suiteXmlFile>./src/testng.xml</suiteXmlFile>
                    </suiteXmlFiles>
                </configuration>
            </plugin>
            <!-- Compiler plugin configures the java version to be used for compiling the code -->
            <plugin>
                <groupId>org.apache.maven.plugins</groupId>
                <artifactId>maven-compiler-plugin</artifactId>
                <version>3.8.0</version>
                <configuration>
                    <source>1.8</source>
                    <target>1.8</target>
                </configuration>
            </plugin>
```



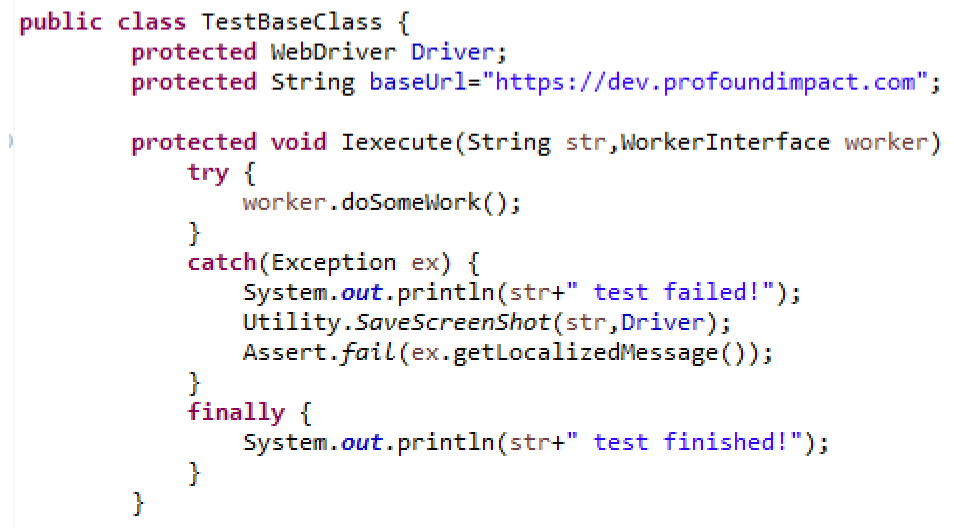
```xml
<dependencies>
  <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-java</artifactId>
      <version>3.14.0</version>
      <scope>test</scope>
  </dependency>
  <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-ie-driver</artifactId>
      <version>3.141.59</version>
  </dependency>
  <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-firefox-driver</artifactId>
      <version>3.141.59</version>
  </dependency>
  <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-support</artifactId>
      <version>3.14.0</version>
  </dependency>
  <dependency>
      <groupId>org.seleniumhq.selenium</groupId>
      <artifactId>selenium-chrome-driver</artifactId>
      <version>3.14.0</version>
  </dependency>



<repositories>
    <repository>
      <id>java.net</id>
      <url>https://maven.java.net/content/repositories/public/</url>
    </repository>
    <repository>
      <id>JBoss repository</id>
      <url>http://repository.jboss.org/nexus/content/groups/public/</url>
    </repository>
</repositories>
```

```xml
<dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>6.3.1</version>
    <scope>compile</scope>
</dependency>
 <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.12</version>
    <scope>compile</scope>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi-ooxml -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi-ooxml</artifactId>
    <version>3.9</version>
</dependency>
<!-- https://mvnrepository.com/artifact/org.apache.poi/poi -->
<dependency>
    <groupId>org.apache.poi</groupId>
    <artifactId>poi</artifactId>
    <version>4.1.0</version>
</dependency>

<!-- https://mvnrepository.com/artifact/de.sciss/fileutil -->
<dependency>
    <groupId>de.sciss</groupId>
    <artifactId>fileutil_2.13.0-M5</artifactId>
    <version>1.1.3</version>
</dependency>
<!-- https://mvnrepository.com/artifact/log4j/log4j -->
<dependency>
    <groupId>log4j</groupId>
    <artifactId>log4j</artifactId>
    <version>1.2.17</version>
</dependency>
<!-- https://mvnrepository.com/artifact/xin.xihc/CommonUtils -->
<dependency>
    <groupId>xin.xihc</groupId>
    <artifactId>CommonUtils</artifactId>
    <version>1.19.6</version>
</dependency>
```

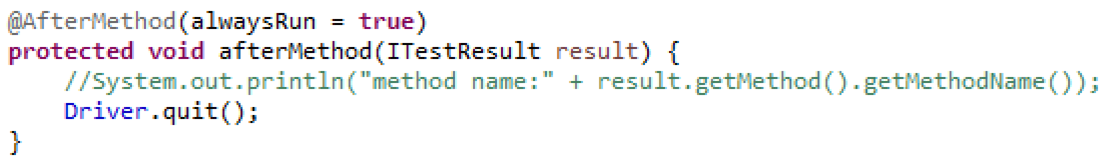
Base Class



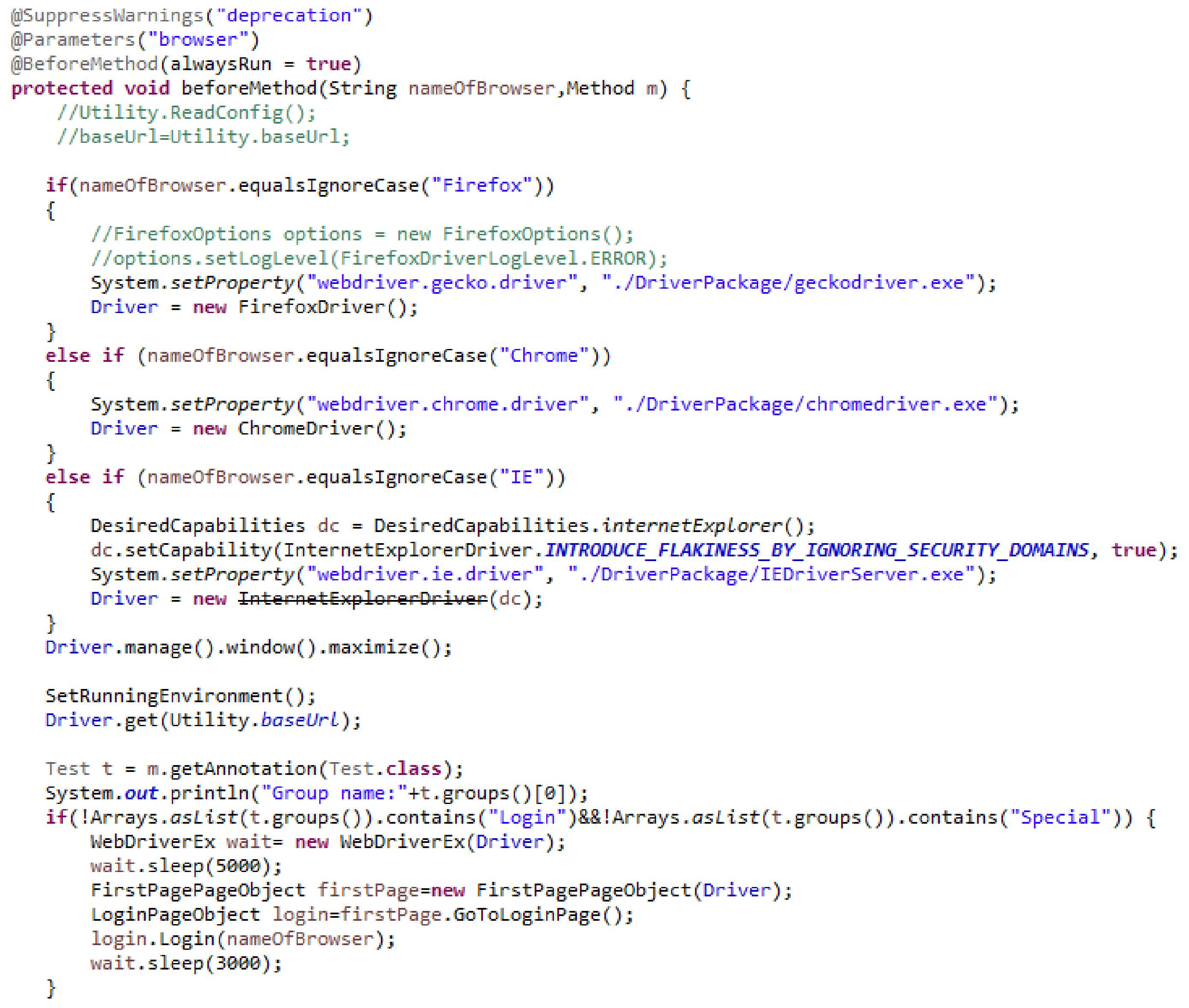
```java
@FunctionalInterface
public interface WorkerInterface {
    public void doSomeWork();
}
```



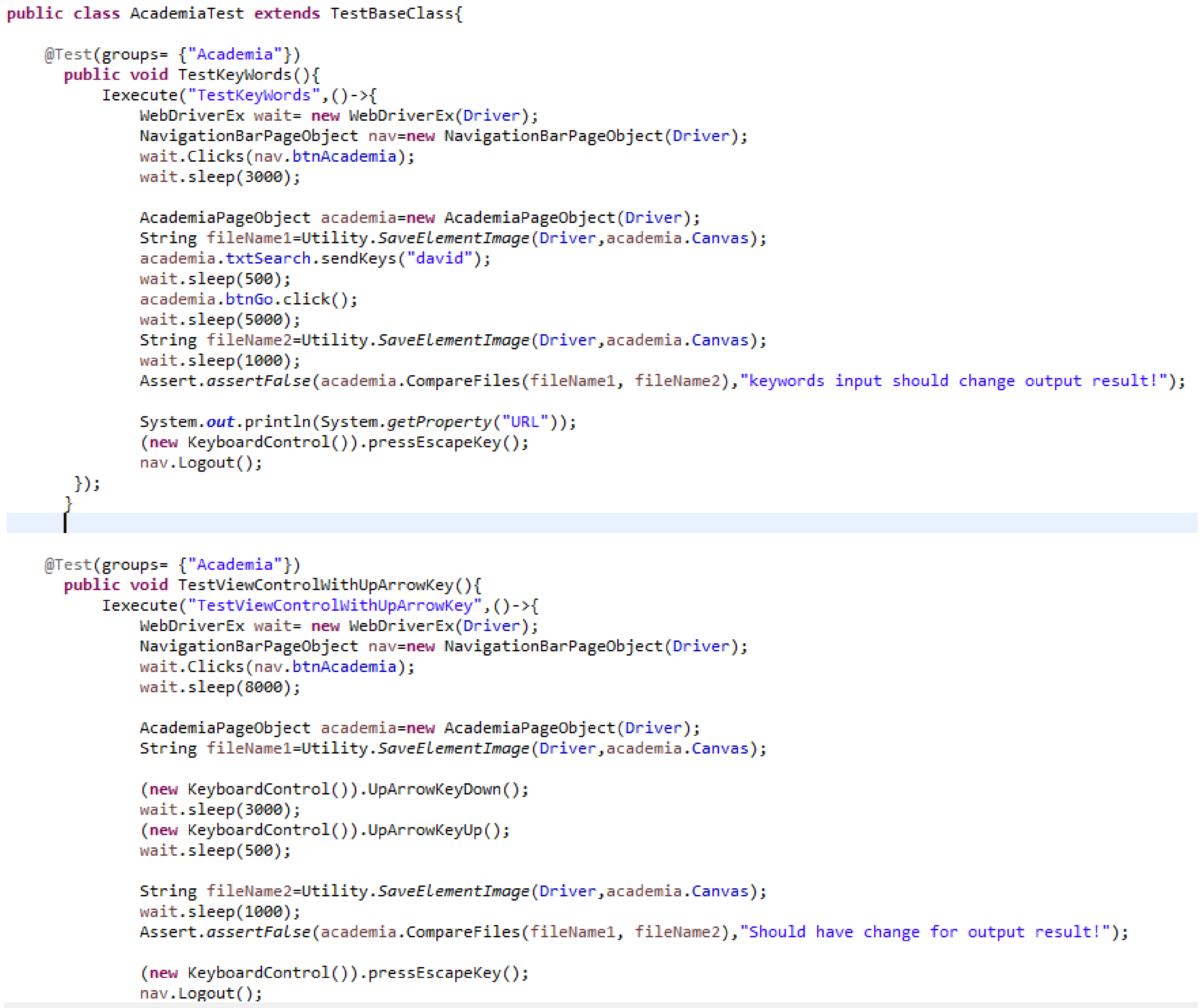
```java
public class TestBaseClass {
        protected WebDriver Driver;
        protected String baseUrl="https://dev.profoundimpact.com";

        protected void Iexecute(String str,WorkerInterface worker)
            try {
                worker.doSomeWork();
            }
            catch(Exception ex) {
                System.out.println(str+" test failed!");
                Utility.SaveScreenShot(str,Driver);
                Assert.fail(ex.getLocalizedMessage());
            }
            finally {
                System.out.println(str+" test finished!");
            }
        }
```



```java
@AfterMethod(alwaysRun = true)
protected void afterMethod(ITestResult result) {
    //System.out.println("method name:" + result.getMethod().getMethodName());
    Driver.quit();
}
```

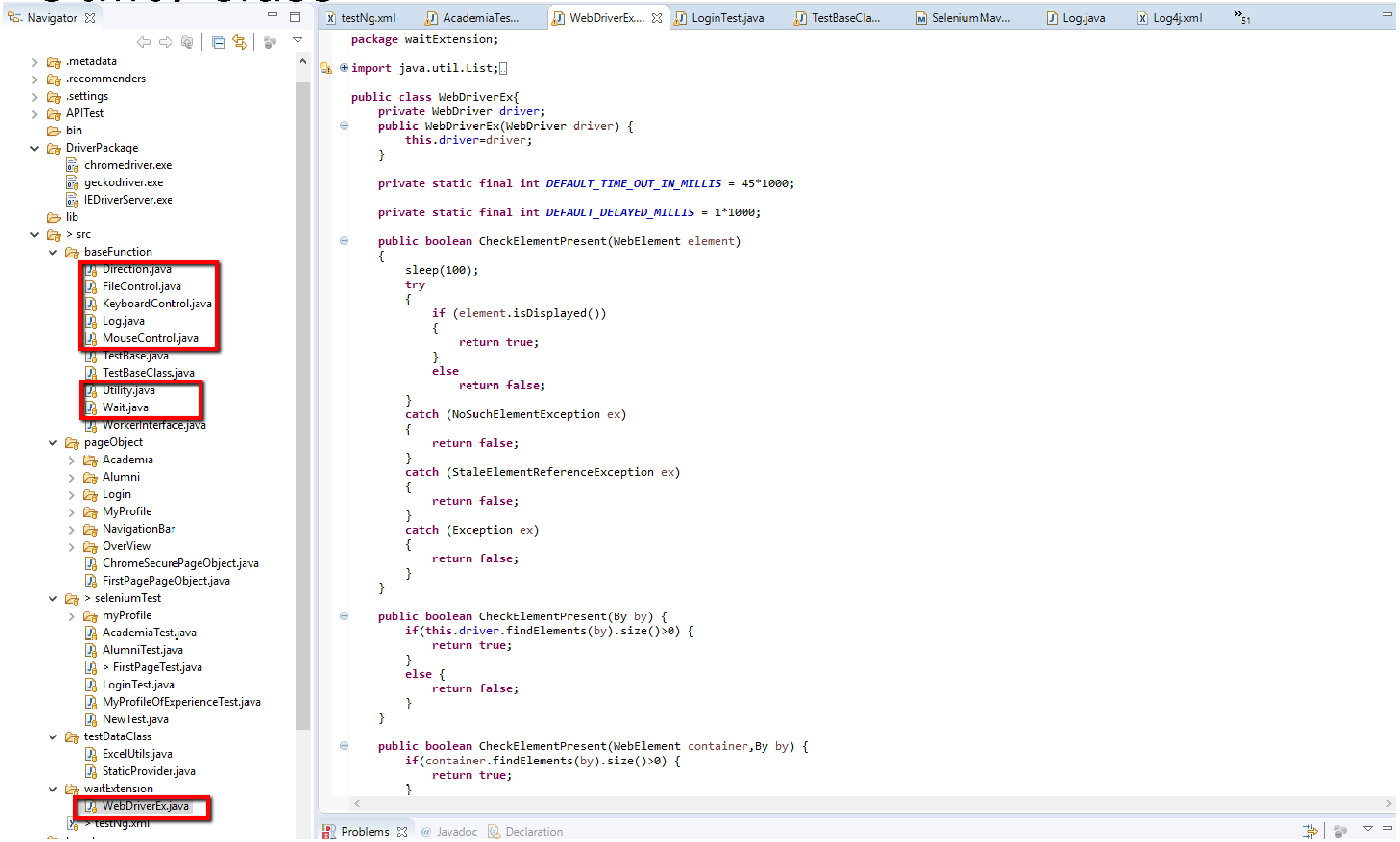


```java
@SuppressWarnings("deprecation")
@Parameters("browser")
@BeforeMethod(alwaysRun = true)
protected void beforeMethod(String nameOfBrowser,Method m) {
    //Utility.ReadConfig();
    //baseUrl=Utility.baseUrl;

   if(nameOfBrowser.equalsIgnoreCase("Firefox"))
   {
       //FirefoxOptions options = new FirefoxOptions();
       //options.setLogLevel(FirefoxDriverLogLevel.ERROR);
       System.setProperty("webdriver.gecko.driver", "./DriverPackage/geckodriver.exe");
       Driver = new FirefoxDriver();
   }
   else if (nameOfBrowser.equalsIgnoreCase("Chrome"))
   {
       System.setProperty("webdriver.chrome.driver", "./DriverPackage/chromedriver.exe");
       Driver = new ChromeDriver();
   }
   else if (nameOfBrowser.equalsIgnoreCase("IE"))
   {
       DesiredCapabilities dc = DesiredCapabilities.internetExplorer();
       dc.setCapability(InternetExplorerDriver.INTRODUCE_FLAKINESS_BY_IGNORING_SECURITY_DOMAINS, true);
       System.setProperty("webdriver.ie.driver", "./DriverPackage/IEDriverServer.exe");
       Driver = new ~~InternetExplorerDriver~~(dc);
   }
   Driver.manage().window().maximize();

   SetRunningEnvironment();
   Driver.get(Utility.baseUrl);

   Test t = m.getAnnotation(Test.class);
   System.out.println("Group name:"+t.groups()[0]);
   if(!Arrays.asList(t.groups()).contains("Login")&&!Arrays.asList(t.groups()).contains("Special")) {
       WebDriverEx wait= new WebDriverEx(Driver);
       wait.sleep(5000);
       FirstPagePageObject firstPage=new FirstPagePageObject(Driver);
       LoginPageObject login=firstPage.GoToLoginPage();
       login.Login(nameOfBrowser);
       wait.sleep(3000);
   }
```



```java
public class AcademiaTest extends TestBaseClass{

    @Test(groups= {"Academia"})
      public void TestKeyWords(){
          Iexecute("TestKeyWords",()->{
              WebDriverEx wait= new WebDriverEx(Driver);
              NavigationBarPageObject nav=new NavigationBarPageObject(Driver);
              wait.Clicks(nav.btnAcademia);
              wait.sleep(3000);

              AcademiaPageObject academia=new AcademiaPageObject(Driver);
              String fileName1=Utility.SaveElementImage(Driver,academia.Canvas);
              academia.txtSearch.sendKeys("david");
              wait.sleep(500);
              academia.btnGo.click();
              wait.sleep(5000);
              String fileName2=Utility.SaveElementImage(Driver,academia.Canvas);
              wait.sleep(1000);
              Assert.assertFalse(academia.CompareFiles(fileName1, fileName2),"keywords input should change output result!");

              System.out.println(System.getProperty("URL"));
              (new KeyboardControl()).pressEscapeKey();
              nav.Logout();
       });
      }


    @Test(groups= {"Academia"})
      public void TestViewControlWithUpArrowKey(){
          Iexecute("TestViewControlWithUpArrowKey",()->{
              WebDriverEx wait= new WebDriverEx(Driver);
              NavigationBarPageObject nav=new NavigationBarPageObject(Driver);
              wait.Clicks(nav.btnAcademia);
              wait.sleep(8000);

              AcademiaPageObject academia=new AcademiaPageObject(Driver);
              String fileName1=Utility.SaveElementImage(Driver,academia.Canvas);

              (new KeyboardControl()).UpArrowKeyDown();
              wait.sleep(3000);
              (new KeyboardControl()).UpArrowKeyUp();
              wait.sleep(500);

              String fileName2=Utility.SaveElementImage(Driver,academia.Canvas);
              wait.sleep(1000);
              Assert.assertFalse(academia.CompareFiles(fileName1, fileName2),"Should have change for output result!");

              (new KeyboardControl()).pressEscapeKey();
              nav.Logout();
```

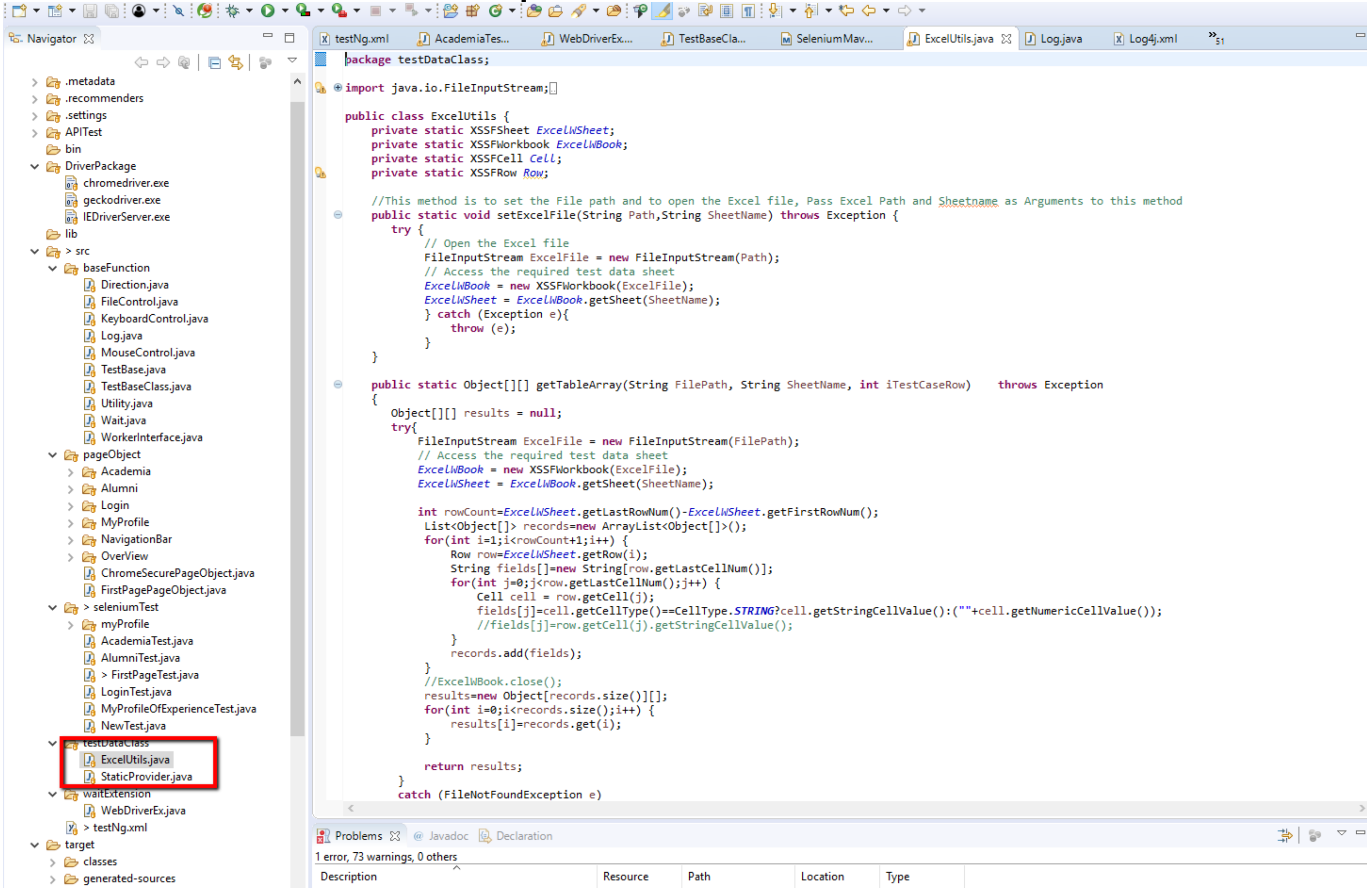
Utility class



```
.metadata
.recommenders
.settings
APITest
bin
DriverPackage
    chromedriver.exe
    geckodriver.exe
    IEDriverServer.exe
lib
> src
    baseFunction
        Direction.java
        FileControl.java
        KeyboardControl.java
        Log.java
        MouseControl.java
        TestBase.java
        TestBaseClass.java
        Utility.java
        Wait.java
        WorkerInterface.java
    pageObject
        Academia
        Alumni
        Login
        MyProfile
        NavigationBar
        OverView
        ChromeSecurePageObject.java
        FirstPagePageObject.java
    > seleniumTest
        myProfile
        AcademiaTest.java
        AlumniTest.java
        > FirstPageTest.java
        LoginTest.java
        MyProfileOfExperienceTest.java
        NewTest.java
    testDataClass
        ExcelUtils.java
        StaticProvider.java
    waitExtension
        WebDriverEx.java
    > testNg.xml
```

```java
package waitExtension;

import java.util.List;

public class WebDriverEx{
    private WebDriver driver;
    public WebDriverEx(WebDriver driver) {
        this.driver=driver;
    }

    private static final int DEFAULT_TIME_OUT_IN_MILLIS = 45*1000;

    private static final int DEFAULT_DELAYED_MILLIS = 1*1000;

    public boolean CheckElementPresent(WebElement element)
    {
        sleep(100);
        try
        {
            if (element.isDisplayed())
            {
                return true;
            }
            else
                return false;
        }
        catch (NoSuchElementException ex)
        {
            return false;
        }
        catch (StaleElementReferenceException ex)
        {
            return false;
        }
        catch (Exception ex)
        {
            return false;
        }
    }

    public boolean CheckElementPresent(By by) {
        if(this.driver.findElements(by).size()>0) {
            return true;
        }
        else {
            return false;
        }
    }

    public boolean CheckElementPresent(WebElement container,By by) {
        if(container.findElements(by).size()>0) {
            return true;
        }
```

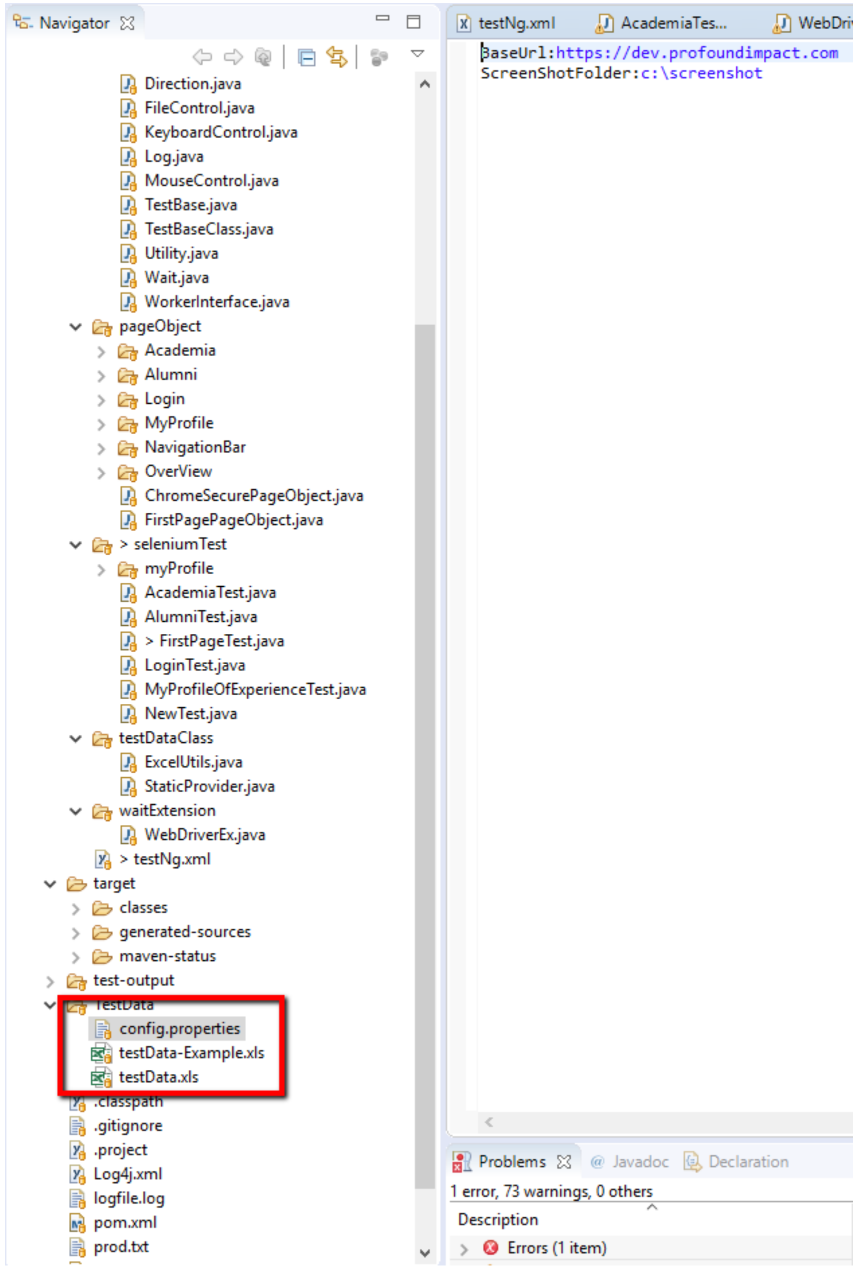
Test class for data provider



```java
package testDataClass;

import java.io.FileInputStream;

public class ExcelUtils {
    private static XSSFSheet ExcelWSheet;
    private static XSSFWorkbook ExcelWBook;
    private static XSSFCell Cell;
    private static XSSFRow Row;

    //This method is to set the File path and to open the Excel file, Pass Excel Path and Sheetname as Arguments to this method
    public static void setExcelFile(String Path,String SheetName) throws Exception {
       try {
            // Open the Excel file
            FileInputStream ExcelFile = new FileInputStream(Path);
            // Access the required test data sheet
            ExcelWBook = new XSSFWorkbook(ExcelFile);
            ExcelWSheet = ExcelWBook.getSheet(SheetName);
            } catch (Exception e){
                throw (e);
            }
    }

    public static Object[][] getTableArray(String FilePath, String SheetName, int iTestCaseRow)    throws Exception
    {
       Object[][] results = null;
       try{
           FileInputStream ExcelFile = new FileInputStream(FilePath);
           // Access the required test data sheet
           ExcelWBook = new XSSFWorkbook(ExcelFile);
           ExcelWSheet = ExcelWBook.getSheet(SheetName);

           int rowCount=ExcelWSheet.getLastRowNum()-ExcelWSheet.getFirstRowNum();
            List<Object[]> records=new ArrayList<Object[]>();
            for(int i=1;i<rowCount+1;i++) {
                Row row=ExcelWSheet.getRow(i);
                String fields[]=new String[row.getLastCellNum()];
                for(int j=0;j<row.getLastCellNum();j++) {
                    Cell cell = row.getCell(j);
                    fields[j]=cell.getCellType()==CellType.STRING?cell.getStringCellValue():(""+cell.getNumericCellValue());
                    //fields[j]=row.getCell(j).getStringCellValue();
                }
                records.add(fields);
            }
            //ExcelWBook.close();
            results=new Object[records.size()][];
            for(int i=0;i<records.size();i++) {
                results[i]=records.get(i);
            }

            return results;
        }
        catch (FileNotFoundException e)
```

```
1 error, 73 warnings, 0 others
```

Configure file and Test data



Navigator | testNg.xml | AcademiaTes... | WebDri

```
BaseUrl:https://dev.profoundimpact.com
ScreenShotFolder:c:\screenshot
```

```
        Direction.java
        FileControl.java
        KeyboardControl.java
        Log.java
        MouseControl.java
        TestBase.java
        TestBaseClass.java
        Utility.java
        Wait.java
        WorkerInterface.java
    pageObject
        Academia
        Alumni
        Login
        MyProfile
        NavigationBar
        OverView
        ChromeSecurePageObject.java
        FirstPagePageObject.java
    > seleniumTest
        myProfile
        AcademiaTest.java
        AlumniTest.java
        > FirstPageTest.java
        LoginTest.java
        MyProfileOfExperienceTest.java
        NewTest.java
    testDataClass
        ExcelUtils.java
        StaticProvider.java
    waitExtension
        WebDriverEx.java
    > testNg.xml
target
    classes
    generated-sources
    maven-status
test-output
TestData
    config.properties
    testData-Example.xls
    testData.xls
.classpath
.gitignore
.project
Log4j.xml
logfile.log
pom.xml
prod.txt
```

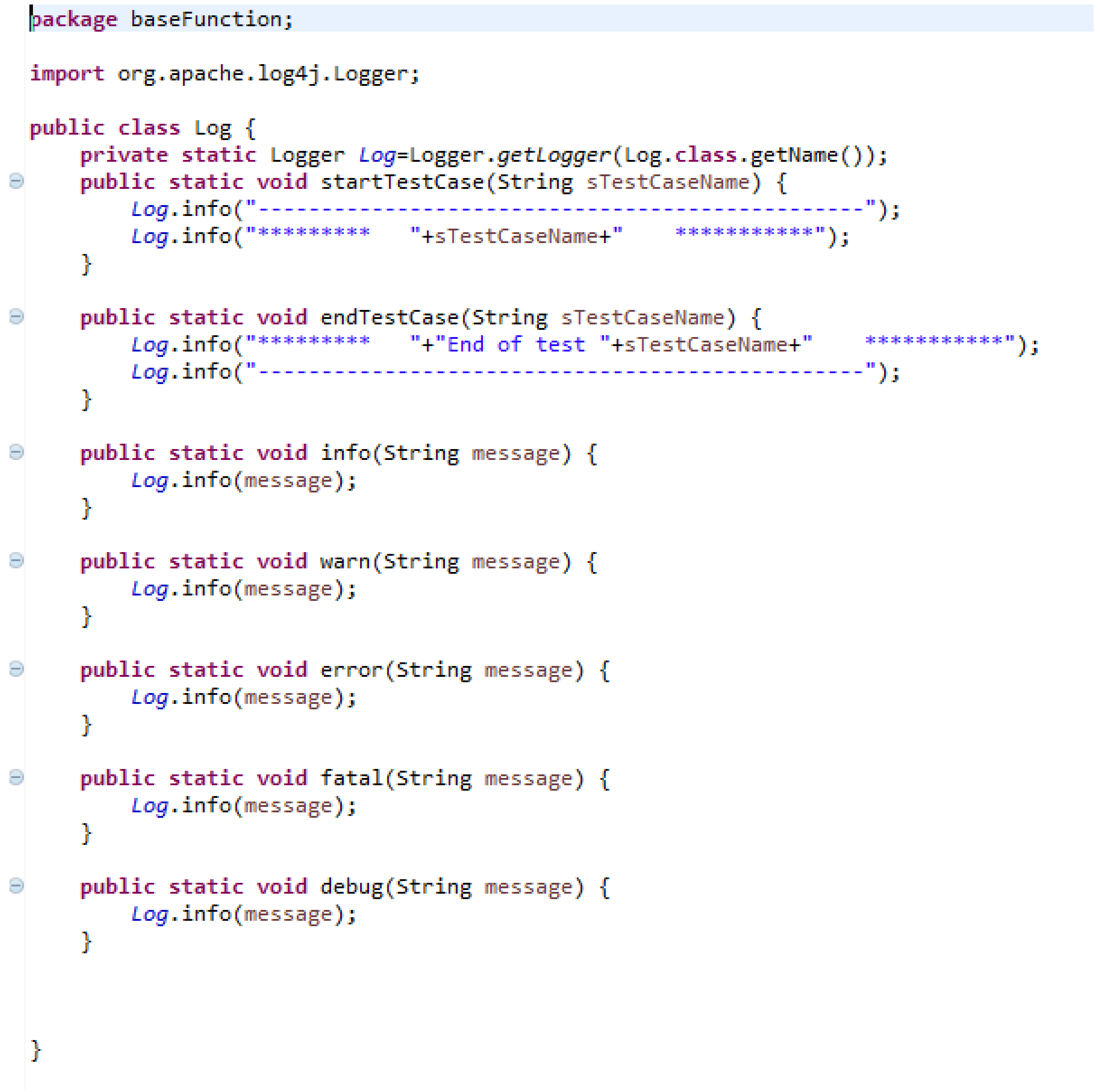
Problems | Javadoc | Declaration

1 error, 73 warnings, 0 others

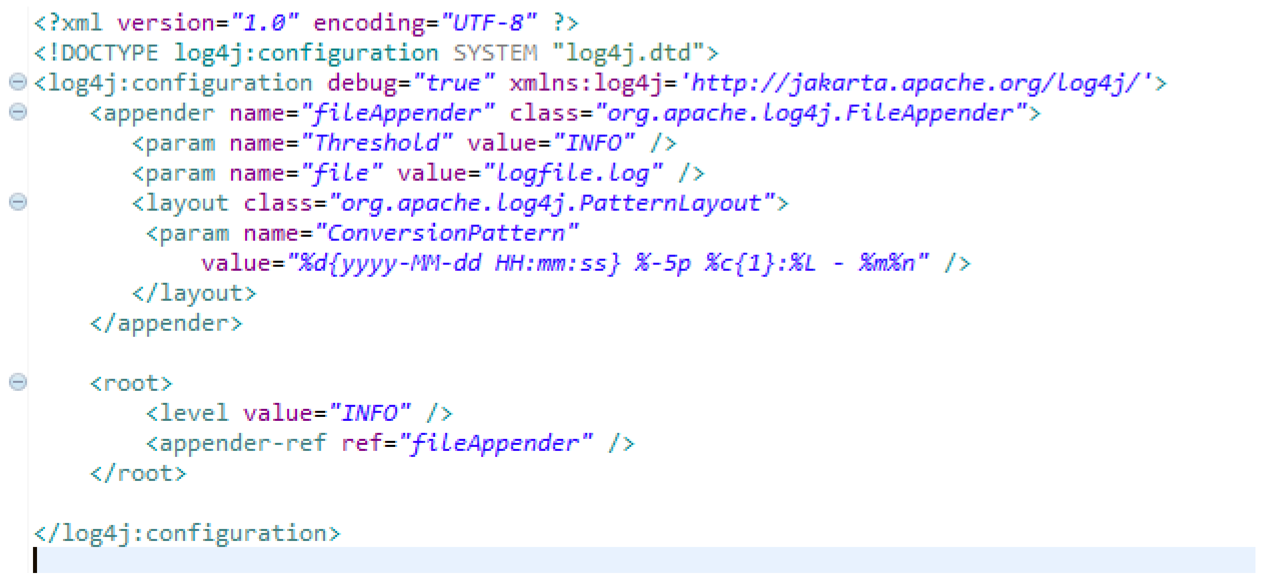
Description

Errors (1 item)

Log class

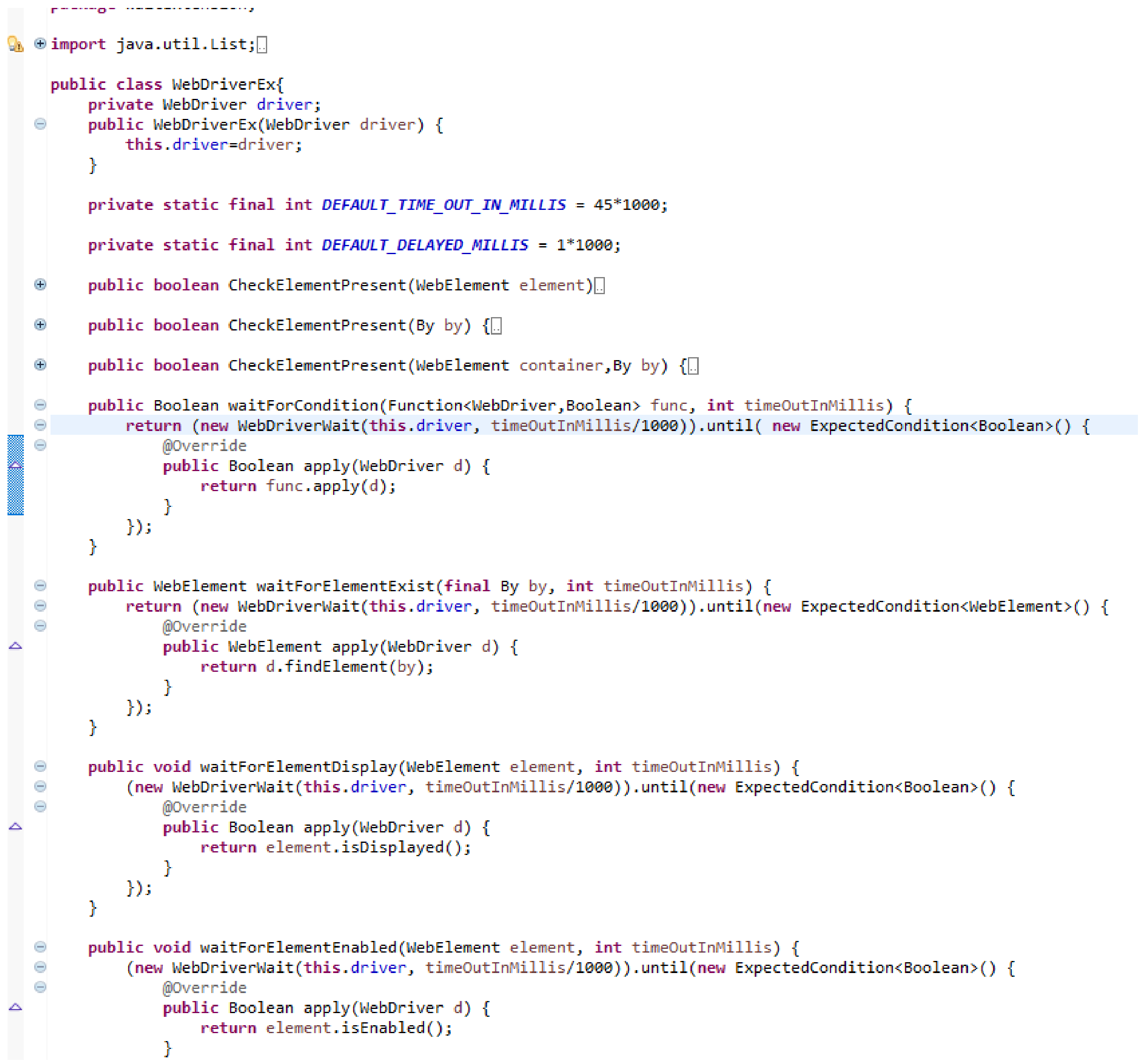


```java
package baseFunction;

import org.apache.log4j.Logger;

public class Log {
    private static Logger Log=Logger.getLogger(Log.class.getName());
    public static void startTestCase(String sTestCaseName) {
        Log.info("------------------------------------------------");
        Log.info("*********    "+sTestCaseName+"    ***********");
    }

    public static void endTestCase(String sTestCaseName) {
        Log.info("*********    "+"End of test "+sTestCaseName+"    ***********");
        Log.info("------------------------------------------------");
    }

    public static void info(String message) {
        Log.info(message);
    }

    public static void warn(String message) {
        Log.info(message);
    }

    public static void error(String message) {
        Log.info(message);
    }

    public static void fatal(String message) {
        Log.info(message);
    }

    public static void debug(String message) {
        Log.info(message);
    }


}
```



```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration debug="true" xmlns:log4j='http://jakarta.apache.org/log4j/'>
    <appender name="fileAppender" class="org.apache.log4j.FileAppender">
       <param name="Threshold" value="INFO" />
       <param name="file" value="logfile.log" />
       <layout class="org.apache.log4j.PatternLayout">
        <param name="ConversionPattern"
            value="%d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L - %m%n" />
       </layout>
    </appender>

    <root>
        <level value="INFO" />
        <appender-ref ref="fileAppender" />
    </root>

</log4j:configuration>
```

Explicit wait class



```java
import java.util.List;

public class WebDriverEx{
    private WebDriver driver;
    public WebDriverEx(WebDriver driver) {
        this.driver=driver;
    }

    private static final int DEFAULT_TIME_OUT_IN_MILLIS = 45*1000;

    private static final int DEFAULT_DELAYED_MILLIS = 1*1000;

    public boolean CheckElementPresent(WebElement element)

    public boolean CheckElementPresent(By by) {

    public boolean CheckElementPresent(WebElement container,By by) {

    public Boolean waitForCondition(Function<WebDriver,Boolean> func, int timeOutInMillis) {
        return (new WebDriverWait(this.driver, timeOutInMillis/1000)).until( new ExpectedCondition<Boolean>() {
            @Override
            public Boolean apply(WebDriver d) {
                return func.apply(d);
            }
        });
    }

    public WebElement waitForElementExist(final By by, int timeOutInMillis) {
        return (new WebDriverWait(this.driver, timeOutInMillis/1000)).until(new ExpectedCondition<WebElement>() {
            @Override
            public WebElement apply(WebDriver d) {
                return d.findElement(by);
            }
        });
    }

    public void waitForElementDisplay(WebElement element, int timeOutInMillis) {
        (new WebDriverWait(this.driver, timeOutInMillis/1000)).until(new ExpectedCondition<Boolean>() {
            @Override
            public Boolean apply(WebDriver d) {
                return element.isDisplayed();
            }
        });
    }

    public void waitForElementEnabled(WebElement element, int timeOutInMillis) {
        (new WebDriverWait(this.driver, timeOutInMillis/1000)).until(new ExpectedCondition<Boolean>() {
            @Override
            public Boolean apply(WebDriver d) {
                return element.isEnabled();
            }
```

Other encapsulated classes

- Keyboard control
- Mouse control
- Other common methods related to string/datetime

Final running

- MVN test

Any Question?