Practical No. 10 TestNg Framework Introduction

Date: _____

Aim:

To study TestNg Framework.

Theory:

TestNG is an automation testing framework in which NG stands for "Next Generation". It was developed by Cedric Beust. TestNG is inspired by JUnit which uses the annotations (@). TestNG overcomes the disadvantages of JUnit and is designed to make end-to-end testing easy. Using TestNG, you can generate a proper report, and you can easily come to know how many test cases are passed, failed, and skipped. You can execute the failed test cases separately. Default Selenium tests do not generate a proper format for the test results. Using TestNG in Selenium, we can generate test results.

**Following are the key features of Selenium TestNG:**

- Generate the report in a proper format including a number of test cases runs, the number of test cases passed, the number of test cases failed, and the number of test cases skipped.
- Multiple test cases can be grouped more easily by converting them into testng.xml file. In which you can make priorities which test case should be executed first.
- The same test case can be executed multiple times without loops just by using keyword called 'invocation count.'
- Using testng, you can execute multiple test cases on multiple browsers, i.e., cross browser testing.
- The TestNG framework can be easily integrated with tools like TestNG Maven, Jenkins, etc.
- Annotations used in the testing are very easy to understand ex: @BeforeMethod, @AfterMethod, @BeforeTest, @AfterTest
- TestNG simplifies the way the tests are coded. There is no more need for a static main method in our tests. The sequence of actions is regulated by easy-to-understand annotations that do not require methods to be static.
- Uncaught exceptions are automatically handled by TestNG without terminating the test prematurely. These exceptions are reported as failed steps in the report.
- WebDriver has no native mechanism for generating reports. TestNG can generate the report in a readable format like the one shown below.
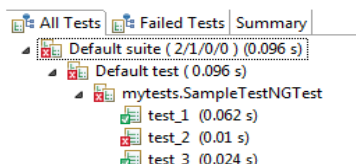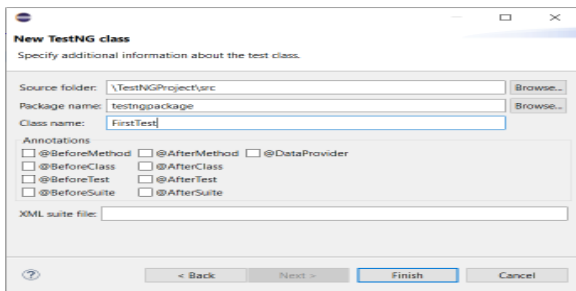
Creating your first TestNG class

Perform the following steps to create your first TestNG class:

1. Go to File | New | Other. This will open a new Add wizard window in Eclipse.

2. Select TestNG class from the Add wizard window and click on Next.
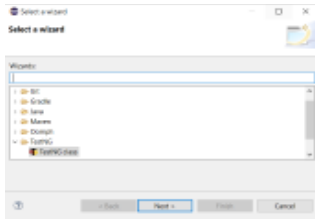
3. On the next window click on the Browse button and select the Java project here you need to add your class.

4. Enter the package name and the test class name and click on Finish.

This window also gives you an option to select different annotations while creating a new TestNG class.

If selected, the plugin will generate dummy methods for these annotations while generating the class.

This will add a new TestNG class to your project.

Write the following code to your newly created test class:

```java
package test;

import org.testng.annotations.Test;

public class FirstTest {
  @Test
  public void testMethod() {
    System.out.println("First TestNG test");
  }
}
```

Understanding testng.xml

testng.xml is a configuration file for TestNG.

It is used to define test suites and tests in TestNG. It is also used to pass parameters to test methods.

testng.xml provides different options to include packages, classes, and independent test methods in our test suite.

It also allows us to configure multiple tests in a single test suite and run them in a multithreaded environment.

TestNG allows you to do the following:

Create tests with packages

Create tests using classes

Create tests using test methods

Include/exclude a particular package, class, or test method

Use of regular expression while using the include/exclude feature

Store parameter values for passing to test methods at runtime

Configure multithreaded execution options

Creating a test suite

Let's now create our first TestNG test suite using testng.xml. We will create a simple test suite with only one test method.

Perform the following steps for creating a test suite:

1. Go to the Eclipse project that we created in the previous chapter.

2. Select the project and then right-click on it and select New | File.

3. Select the project in the File window

4. Enter text testng.xml in the File name section, and click on Finish.

5. Eclipse will add the new file to your project and will open the file in the editor.

Add the following snippet to the newly created tesntg.xml file and save it.

```
<suite name="First Suite" verbose="1" >
    <test name="First Test" >
        <classes>
            <class name="test.FirstTest" />
        </classes>
    </test>
</suite>
```

Running testng.xml

Perform the following steps for executing testng.xml using Eclipse:

1. Open Eclipse and go to the project where we have created the testng.xml file.

2. Select the testng.xml file, right-click on it, and select Run As | TestNG suite.

3. Eclipse will execute the XML file as TestNG suite and you can see the report in Eclipse
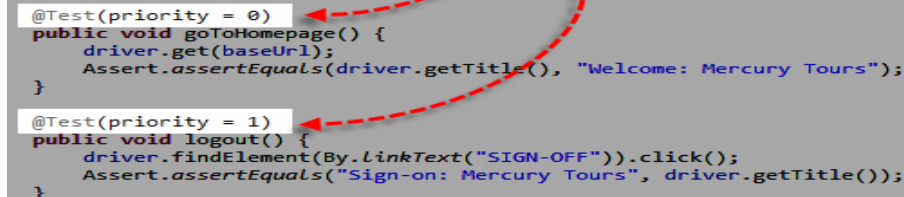
What is Annotation in TestNG?

Annotations in TestNG are lines of code that can control how the method below them will be executed.

They are always preceded by the @ symbol.

A very early and quick TestNG example is the one shown below.

These are 2 examples of annotations

```
@Test(priority = 0)
public void goToHomepage() {
    driver.get(baseUrl);
    Assert.assertEquals(driver.getTitle(), "Welcome: Mercury Tours");
}

@Test(priority = 1)
public void logout() {
    driver.findElement(By.LinkText("SIGN-OFF")).click();
    Assert.assertEquals("Sign-on: Mercury Tours", driver.getTitle());
}
```

The example above simply says that the method goToHomepage() should be executed first before logout() because it has a lower priority number

Implementation

1. Create two TestNg Classes and create and configure TestNG test suite(testng.xml) by adding few test classes to the suite and execute it.

   FirstTest.java

   ```java
   package TestNGProject;
   import org.testng.annotations.Test;
   public class FirstTest {
     @Test
     public void first() {
           System.out.println("I am first method ");
     }
     @Test
     public void second() {
           System.out.println("I am second method");
     }
   }
   ```

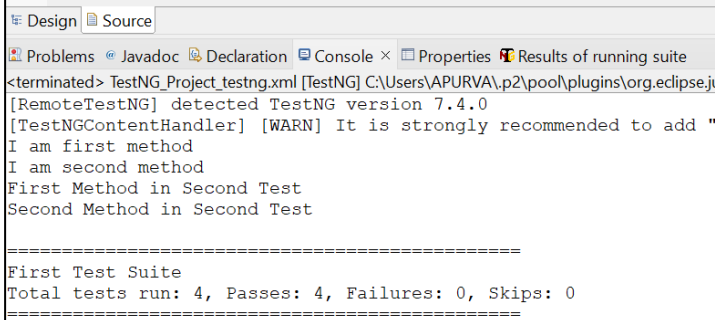   **SecondTest.java**
   ```java
   package TestNGProject;
   import org.testng.annotations.Test;
   public class SecondTest {
     @Test
     public void first() {
           System.out.println("First Method in Second Test");
   }
     @Test
   public void second() {
           System.out.println("Second Method in Second Test");
       }
   }
   ```

Testng.xml

```
<suite name="First Test Suite" verbose="1">
        <test name="First Test">
                <classes>
                        <class name="TestNGProject.FirstTest"/>
                </classes>
        </test>
        <test name="Second Test">
                <classes>
                        <class name="TestNGProject.SecondTest"></class>
                </classes>
        </test>
 </suite>
```

Output:

```
⊞ Design  Source

 Problems  @ Javadoc  Declaration  Console ×  Properties  Results of running suite
<terminated> TestNG_Project_testng.xml [TestNG] C:\Users\APURVA\.p2\pool\plugins\org.eclipse.ju
[RemoteTestNG] detected TestNG version 7.4.0
[TestNGContentHandler] [WARN] It is strongly recommended to add "
I am first method
I am second method
First Method in Second Test
Second Method in Second Test

=================================================
First Test Suite
Total tests run: 4, Passes: 4, Failures: 0, Skips: 0
=================================================
```

2. Create two packages in TestNg project. Each package contains two classes. Create and execute a TestNG test suite(testng.xml) to test a particular test package.

firstPackage

firstTestClass.java

```
package firstPackage;
import org.testng.annotations.Test;
public class firstTestClass {
 @Test
 public void first() {
        System.out.println("First Method in First Test Class in First Package");
 }
  @Test
 public void second() {
        System.out.println("Second Method in First Test Class in First Package");
 }
}
```

secondTestClass.java

```
package firstPackage;
import org.testng.annotations.Test;
public class secondTestClass {
 @Test
 public void first() {
        System.out.println("First Method in Second Class in First Package");
 }
```

```java
 @Test
 public void second() {
         System.out.println("Second Method in Second Class in First Package");
 }
}
```

secondPackage

firstTestClass.java

```java
package secondPackage;
import org.testng.annotations.Test;
public class firstTestClass {
        @Test
        public void first() {
                System.out.println("First Method in First Class in Second Package");
        }
         @Test
        public void second() {
                System.out.println("Second Method in First Class in Second Package");
        }
}
```

secondTestClass.java

```java
package secondPackage;
import org.testng.annotations.Test;
public class secondTestClass {
        @Test
        public void first() {
                System.out.println("First Method in Second Class in Second Package");
        }
         @Test
        public void second() {
                System.out.println("Second Method in Second Class in Second Package");
        }
}
```

Package-TestNG.xml

```xml
<suite name="package testing" verbose="1">
<test name="testing package">
        <packages>
                <package name="firstPackage"/>
        </packages>
</test>
</suite>
```

Output:

```
 Design  Source
 Problems  Javadoc  Declaration  Console ×  Properties  Results of running suite
<terminated> sampleTestNGProject_package-TestNG.xml [TestNG] C:\Users\APURVA\.p2\pool\plug
[RemoteTestNG] detected TestNG version 7.4.0
[TestNGContentHandler] [WARN] It is strongly recommended to add "
First Method in First Test Class in First Package
Second Method in First Test Class in First Package
First Method in Second Class in First Package
Second Method in Second Class in First Package

===============================================
package testing
Total tests run: 4, Passes: 4, Failures: 0, Skips: 0
===============================================
```

3. Create two packages inTestNg project, each package containing two classes. Each class will contain two test menthods. Create and execute a TestNG test suite(testng.xml) to execute only a particular method from a class.

method-TestNG.xml

```
<suite name="method testing" verbose="1">
        <test name="testing method">
                <classes>
                <class name="firstPackage.firstTestClass">
                        <methods><include name="second"/></methods>
                </class>
                </classes>
        </test>
</suite>
```

Output:

```
Design  Source

Problems  Javadoc  Declaration  Console ×  Properties  Results of running suite
<terminated> sampleTestNGProject_method-TestNG.xml [TestNG] C:\Users\APURVA\.p2\pool\plugins
[RemoteTestNG] detected TestNG version 7.4.0
[TestNGContentHandler] [WARN] It is strongly recommended to add "<
Second Method in First Test Class in First Package

===============================================
method testing
Total tests run: 1, Passes: 1, Failures: 0, Skips: 0
===============================================
```

4. Create two packages inTestNg project, each package containing two classes. Each class will contain two test menthods. Create and execute a TestNG test suite(testng.xml) to include only a first method from each class.

method-TestNG.xml

```
<suite name="method testing" verbose="1">
<test name="testing method">
        <classes><class name="firstPackage.firstTestClass">
                        <methods><include name="first"/></methods>
                </class>
                <class name="firstPackage.secondTestClass">
                        <methods><include name="first"/></methods>
                </class>
                <class name="secondPackage.firstTestClass">
                        <methods><include name="first"/></methods>
                </class>
                <class name="secondPackage.secondTestClass">
                        <methods><include name="first"/></methods>
                </class>
        </classes>
</test>
</suite>
```
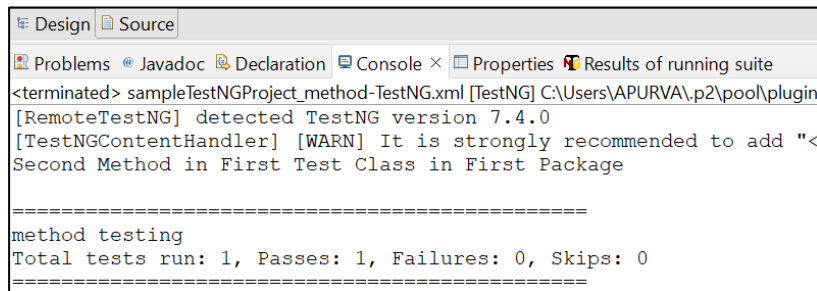
Output:

```
Design  Source

 Problems  @ Javadoc  Declaration  Console ×  Properties  Results of running suite
<terminated> sampleTestNGProject_method-TestNG.xml [TestNG] C:\Users\APURVA\.p2\pool\plugin
[RemoteTestNG] detected TestNG version 7.4.0
[TestNGContentHandler] [WARN] It is strongly recommended to add "<
First Method in First Test Class in First Package
First Method in Second Class in First Package
First Method in First Class in Second Package
First Method in Second Class in Second Package

===============================================
method testing
Total tests run: 4, Passes: 4, Failures: 0, Skips: 0
===============================================
```

5. Create three packages inTestNg project, each package containing two classes. Each class will contain two test menthods.Create and execute a TestNG test suite(testng.xml) to execute test methods from all the classes present under all the packages excluding those that belong to second package.

Combine-testing.xml

```
<suite name="Combine Testing" verbose="1">
<test name="testing method">
        <packages>
                <package name="firstPackage"/>
                <package name="thirdPackage"/>
        </packages>
        <classes>

        </classes>
</test>
</suite>
```

Output:

```
Design  Source

 Problems  @ Javadoc  Declaration  Console ×  Properties  Results of running suite
<terminated> sampleTestNGProject_combine-Testing.xml [TestNG] C:\Users\APURVA\.p2\pool\plugi
[RemoteTestNG] detected TestNG version 7.4.0
[TestNGContentHandler] [WARN] It is strongly recommended to add "
First Method in First Test Class in First Package
Second Method in First Test Class in First Package
I am First Method of First Test Class of Third Package
I am Second Method of First Test Class of Third Package
First Method in Second Class in First Package
Second Method in Second Class in First Package
I am First Method of Second Test Class of Third Package
I am Second Method of Second Test Class of Third Package

===============================================
Combine Testing
Total tests run: 8, Passes: 8, Failures: 0, Skips: 0
===============================================
```

Conclusion: Understood how to test application using TestNG frameworks.

After performing this Practical/lab, students are expected to answer following questions

Q.1 What is TestNG Framework?

Q.2 What is testing.xml file?