Module-7 Selenium-testNG

Problem Statement 1: Custom Annotation for Test Methods Create a Java program that utilizes custom annotations for test methods. Define annotations such as @SmokeTest and @RegressionTest and use them to categorize and execute specific groups of test methods.

```
package testNG;

import java.lang.annotation.Retention;
import java.lang.annotation.RetentionPolicy;
import java.lang.reflect.Method;

public class CustomAnnotationsTest{

	@Retention(RetentionPolicy.RUNTIME)
	@interface SmokeTest {}

	@Retention(RetentionPolicy.RUNTIME)
	@interface RegressionTest {}

	@SmokeTest
    public void smokeTest() {System.out.println("Executing Smoke Test");}

    @RegressionTest
    public void regressionTest() {System.out.println("Executing Regression
Test");}

    @SmokeTest
    @RegressionTest
    public void smokeAndRegressionTest() {System.out.println("Executing Both
Smoke and Regression Test");}


	/*
	 * If you're using custom annotations that TestNG doesn't recognize
	 * (e.g., @SmokeTest or @RegressionTest),
	 * you'll need to handle the execution manually using a main method.
	 */

    public static void main(String[] args) {
      CustomAnnotationsTest test = new CustomAnnotationsTest();


        Method[] methods = test.getClass().getMethods();
        for (Method method : methods) {
            if (method.isAnnotationPresent(SmokeTest.class)) {
                System.out.println("Running Smoke Test: " + method.getName());
                try {
                    method.invoke(test);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
            if (method.isAnnotationPresent(RegressionTest.class)) {
                System.out.println("Running Regression Test: "
+method.getName());
                try {
                    method.invoke(test);
                } catch (Exception e) {
```

```
                    e.printStackTrace();
                }
            }
        }
    }

}
```

Problem Statement 2: Running Tests in Parallel - XML Configuration Extend the program to include XML configuration for parallel execution. Create an XML file that defines test suites and configurations for running tests in parallel. Execute the tests using the XML configuration.

```
package testNG;

import org.testng.annotations.Test;


public class TestNGexample {


	@Test

	void testMethod1() {
		System.out.println("method-1");
	}
	@Test

	void appleMethod() {
		System.out.println("method-apple");
	}

	@Test

	void manMethod() {
		System.out.println("method-man");
	}

	@Test
	void bigMethod() {
		System.out.println("method-big");
	}
}
```

```java
package testNG;

import org.testng.annotations.Test;


public class TestNGparallelExecution {

	@Test(priority=2)

	void priority2Method() {
		System.out.println("parallel execution");
	}
}
```

testNG.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite" thread-count="2" parallel="tests">
  <test name="series">
    <classes>
      <!--<class name="testNG.TestNGexample"/> -->
      <class name="testNG.TestNGparallelExecution"/>
    </classes>
  </test> <!-- Test -->
   <test name="parallel">
    <classes>
      <class name="testNG.TestNGexample"/>
      <!--<class name="testNG.TestNGparallelExecution"/>-->
    </classes>
  </test> <!-- Test -->
</suite> <!-- Suite -->
```