**TestNG Assignment**

**Name : SANJOG PATEL**

**Batch : 240**

**EmployeeId : 2607661**

**Problem Statement 1: TestNG Annotations**

**1. Create the Java Calculator Application and test the same with TestNG Framework.**

**Description of Calculator Application:**

**Create the Class names as Calculator**

a. Create four Methods:

i. int add (int n1, int n2);

ii. int sub (int n1, int n2);

iii. int mul (int n1, int n2);

iv. float div (int n1, int n2);

b. Create the main method

i. To get the input from the user and call the above said function**.**

package com.testng.TestNG\_Projects;

import java.util.Scanner;

public class Calculator {

int add(int n1,int n2) {

return n1+n2;

}

int sub(int n1,int n2) {

return n1-n2;

}

int mul(int n1,int n2) {

return n1\*n2;

}

float div(int n1,int n2) {

if(n2!=0) {

return n1/n2;

}

else {

throw new ArithmeticException();

}

}

public static void main(String args[]) {

Calculator cal = new Calculator();

Scanner sc = new Scanner(System.in);

System.out.println("Enter the 1st number : ");

int n1 = sc.nextInt();

sc.nextLine();

System.out.println("Enter the 2nd number : ");

int n2 = sc.nextInt();

sc.nextLine();

System.out.println("Choose which operation you want to perform ");

System.out.println("+");

System.out.println("-");

System.out.println("\*");

System.out.println("/");

System.out.println("---------------------");

System.out.println("---------------------");

char operation = sc.next().charAt(0);

int IntResult;

double DoubleResult;

switch(operation)

{

case '+':

IntResult = cal.add(n1, n2);

System.out.println("Sum of "+ n1 + " and " + n2 +" is " + IntResult);

break;

case '-':

IntResult = cal.sub(n1, n2);

System.out.println("Substraction of "+n1+ " and "+n2+" is "+IntResult);

break;

case '\*':

IntResult = cal.mul(n1, n2);

System.out.println("Multiplication of "+n1+ " and "+n2+" is "+IntResult);

break;

case '/':

DoubleResult = cal.div(n1, n2);

System.out.println("Division of "+n1+ " and "+n2+" is "+DoubleResult);

break;

}

}

}

**2. TestNG Basic Annotations and dependsOnMethods:**

**To understand the basic TestNG related annotations.**

**Test Procedure:**

a. Create a method calculatorObjectCreation () to create the object for the Calculator class.

Use appropriate annotation to invoke this method before the test execution starts.

b. Add the test methods, testAdd (), testMul (), testSub () and testDiv () to using test

annotation.

c. Test testSub () should execute if and only if the testAdd () Success.

d. Create a testComplete () Method to print a Message “Test cases are executed

Successfully” at the end of all the tests executed.

package com.testng.TestNG\_Projects;

import org.testng.Assert;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class CalculatorTest {

Calculator cal;

@BeforeClass

public void calculatorObjectCreation() {

cal = new Calculator();

}

@Test(priority = 1)

public void testAdd() {

int actual = cal.add(5, 5);

int expected = 10;

Assert.assertEquals(actual, expected,"testAdd Failed!");

}

@Test(priority = 2)

public void testSub() {

int actual = cal.sub(6, 2);

int expected = 4;

Assert.assertEquals(actual, expected,"testSub Failed!");

}

@Test

public void testMul() {

int actual = cal.mul(4, 5);

int expected = 20;

Assert.assertEquals(actual, expected,"testMul Failed!");

}

@Test

public void testDiv() {

double actual = cal.div(15, 5);

double expected = 3.0;

Assert.assertEquals(actual, expected,"testDiv Failed!");

}

@AfterClass

public void testComplete() {

System.out.println("Test cases are executed Successfully");

}

}

**3. TestNG Test Annotations with Priority:**

**To understand the basic TestNG related annotations.**

**Test Procedure:**

a. Add the test methods, testAdd (), testMul (), testSub () and testDiv () to using test

annotation.

b. Execute the test method in order like testAdd (), testSub (), testMul () and testDiv ()

c. Create a testComplete () Method to print a Message “Test cases are executed

Successfully” at the end of all the tests executed.

package com.testng.TestNG\_Projects;

import org.testng.Assert;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class CalculatorTest {

Calculator cal;

@BeforeClass

public void calculatorObjectCreation() {

cal = new Calculator();

}

@Test(priority = 1)

public void testAdd() {

int actual = cal.add(5, 5);

int expected = 10;

Assert.assertEquals(actual, expected,"testAdd Failed!");

}

@Test(priority = 2)

public void testSub() {

int actual = cal.sub(6, 2);

int expected = 4;

Assert.assertEquals(actual, expected,"testSub Failed!");

}

@Test(priority = 3)

public void testMul() {

int actual = cal.mul(4, 5);

int expected = 20;

Assert.assertEquals(actual, expected,"testMul Failed!");

}

@Test(priority = 4)

public void testDiv() {

double actual = cal.div(15, 5);

double expected = 3.0;

Assert.assertEquals(actual, expected,"testDiv Failed!");

}

@AfterClass

public void testComplete() {

System.out.println("Test cases are executed Successfully");

}

}

**4. TestNG Grouping:**

**To understand the basic TestNG related annotations.**

**Test Procedure:**

a. Add the test methods, testAdd (), testMul (), testSub () and testDiv () to using test

annotation.

b. Execute the only these test testAdd () and testSub () as a part of regression test. Use the

testing.xml file to group the test.

c. Create a testComplete () Method to print a Message “Test cases are executed

Successfully” at the end of all the tests executed.

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<suite name="Calculator Test Suite">

<test name="regression Test">

<groups>

<run>

<include name = "regression"/>

</run>

</groups>

</test>

</suite>

package com.testng.TestNG\_Projects;

import org.testng.Assert;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class CalculatorTest {

Calculator cal;

@BeforeClass

public void calculatorObjectCreation() {

cal = new Calculator();

}

@Test(groups = "regression")

public void testAdd() {

int actual = new Calculator().add(5, 5);

int expected = 10;

Assert.assertEquals(actual, expected,"testAdd Failed!");

}

@Test(groups = "regression")

public void testSub() {

int actual = new Calculator().sub(6, 2);

int expected = 4;

Assert.assertEquals(actual, expected,"testSub Failed!");

}

@Test(groups = "basic")

public void testMul() {

int actual = cal.mul(4, 5);

int expected = 20;

Assert.assertEquals(actual, expected,"testMul Failed!");

}

@Test(groups = "basic")

public void testDiv() {

double actual = cal.div(15, 5);

double expected = 3.0;

Assert.assertEquals(actual, expected,"testDiv Failed!");

}

@AfterClass(groups = "regression")

public void testComplete() {

System.out.println("Test cases are executed Successfully");

}

}

**5. TestNG with Parallel Execution:**

**To understand the basic TestNG related annotations.**

**Test Procedure:**

a. Create testing.xml file

b. Make all the 4 tests to run parallelly

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">

<suite name="Calculator Test Suite" parallel="tests" thread-count="1">

<test name="Parallel Test">

<classes>

<class name="CalculatorTest"/>

</classes>

</test>

</suite>

package com.testng.TestNG\_Projects;

import org.testng.Assert;

import org.testng.annotations.AfterClass;

import org.testng.annotations.BeforeClass;

import org.testng.annotations.Test;

public class CalculatorTest {

Calculator cal;

@BeforeClass

public void calculatorObjectCreation() {

cal = new Calculator();

}

@Test

public void testAdd() {

int actual = new Calculator().add(5, 5);

int expected = 10;

Assert.assertEquals(actual, expected,"testAdd Failed!");

}

@Test

public void testSub() {

int actual = new Calculator().sub(6, 2);

int expected = 4;

Assert.assertEquals(actual, expected,"testSub Failed!");

}

@Test

public void testMul() {

int actual = cal.mul(4, 5);

int expected = 20;

Assert.assertEquals(actual, expected,"testMul Failed!");

}

@Test

public void testDiv() {

double actual = cal.div(15, 5);

double expected = 3.0;

Assert.assertEquals(actual, expected,"testDiv Failed!");

}

@AfterClass

public void testComplete() {

System.out.println("Test cases are executed Successfully");

}

}