**LAB # 12**

**JUnit Testing and**

**Introduction to Test Suite**

**OBJECTIVE:** Study the concept of Test Driven Development under the framework of JUnit

Testing and Grouping the multiple JUnit test cases and constructing a Test Suite program.

**Lab Task:**

* Add another test case for countA function (which is given in code).

**Solution:**

**Source Code:**

public class JUnitTesting {

public int square (int x) {return x\*x;}

public int countA(String word) {

int count =0;

for(int i=0;i<word.length();i++) {

if(word.charAt(i)=='a' || word.charAt(i)=='A' ) {

count++;

}

}

return count;

}}

import static org.junit.Assert.\*;

import org.junit.Test;

**public class CountATest {**

@Test

public void test() {

JUnitTesting obj = new JUnitTesting();

int output = obj.countA("ROAIM");

assertEquals(1,output);

}}

**public class SquareTest {**

@Test

public void test() {

JUnitTesting obj = new JUnitTesting();

int output = obj.square(5);

assertEquals(25,output);

}

}

**Output:**

A screenshot of a computer error

Description automatically generated

* Make new project, make a class. Add 2 methods in it. One method will find the max integer present in the input integer array. The other method will find the min integer. Now create test cases for both these methods and test your code. Follow all the steps as mentioned above in the manual.

**Solution:**

**Source Code:**

**public class ArrayFunctions {**

public int min(int[] array) {

int minNum = array[0];

for(int no:array) {

if(no<minNum) {

minNum=no;

}

}

return minNum;

}

public int max(int[] array) {

int maxNum = array[0];

for(int no:array) {

if(no>maxNum) {

maxNum=no;

}

}

return maxNum;

}}

import static org.junit.Assert.\*;

import org.junit.Test;

public class minTest {

@Test

public void test() {

int[] array= {10,20,30,50,40};

ArrayFunctions minFunction = new ArrayFunctions();

int output = minFunction.min(array);

assertEquals(10,output);

}}

import static org.junit.Assert.\*;

import org.junit.Test;

public class maxTest {

@Test

public void test() {

int[] array= {10,20,30,50,40};

ArrayFunctions maxFunction = new ArrayFunctions();

int output = maxFunction.max(array);

assertEquals(50,output);

}}

**Output:**

A screenshot of a error message

Description automatically generated

A green bar with white text

Description automatically generated

* Make a class having four functions for determining,

 Whether the input integer is odd.

 Whether the input integer is even.

* Whether the input integer is prime.

 For calculating the factorial of that input integer.

Write their test cases and execute them in a single test suite class. Follow all the steps mentioned above in the manual.

**Solution:**

**Source Code:**

**public class Functions {**

public boolean isOdd(int no) {

if(no%2==0)

return false;

else {return true;}

}

public boolean isEven(int no) {

if(no%2==0)

return true;

else {return false;}

}

public boolean isPrime(int no) {

if(no<=1) {

return false;

}

else if(no==2) {

return true;

}

else {

for (int i=3;i<no;i++) {

if(no%i==0) {

return false;

}}}

return true;

}

public int factorial(int no) {

if(no==1) {

return 1;

}

else {

return no\*factorial(no-1);

}}}

import static org.junit.Assert.\*;

import org.junit.Test;

**public class IsEvenTest {**

@Test

public void test() {

Functions obj = new Functions();

boolean output = obj.isEven(2);

assertEquals(true,output);

}}

public class IsOddTest {

@Test

public void test() {

Functions obj = new Functions();

boolean output = obj.isOdd(3);

*assertEquals*(true,output);

}}

**public class IsPrimeTest {**

@Test

public void test() {

Functions obj = new Functions();

boolean output = obj.isPrime(11);

assertEquals(true,output);

}}

**public class FactorialTest {**

@Test

public void test() {

Functions obj = new Functions();

int output = obj.factorial(5);

assertEquals(120,output);

}}

import org.junit.runner.RunWith;

import org.junit.runners.Suite;

import org.junit.runners.Suite.SuiteClasses;

@RunWith(Suite.class)

@SuiteClasses({ FactorialTest.class, IsEvenTest.class, IsOddTest.class, IsPrimeTest.class })

**public class AllTests {**

}

**Output:**

A screenshot of a computer error

Description automatically generated