**Software Testing Laboratory**

|  |
| --- |
| **Name:S.Hariharan**  **Rollno:1831017**  **Given Date:05-01-2021**  **Submission Date:05-02-2021** |

**EXCERCISE-1**

1. String Comparison

**Source Code:**

**Stringcmpr.java:**

public class stringcmpr {

public boolean stringcompare(String s1,String s2)

{

if(s1.equals(s2))

{

return true;

}

else

{

return false;

}

}

}

**JunitStrcmp.java:**

import junit.framework.Assert;

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

public class JunitStrcmp {

stringcmpr instance;

public JunitStrcmp() {

}

@Before

public void setUp() {

instance=new stringcmpr();

}

@After

public void tearDown() {

instance=null;

}

@Test

public void StringCmprTest()

{

String s1=new String("Hari");

String s2=new String(s1);

boolean result=instance.stringcompare(s1, s2);

boolean expected=true;

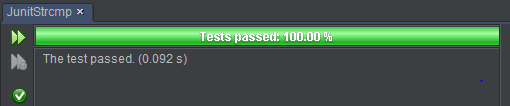
assertEquals(expected, result);

// fail("The Two Strings Are Not equal");

}

}

**Output:**



2.Odd Even Test

**Source Code:**

**OddorEven.java:**

public class OddorEven {

public static String even\_or\_odd(int number) {

return number%2==0 ? "Even" : "Odd";

}

}

**JunitOddorEven:**

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

public class JunitOddorEven {

OddorEven eoo;

public JunitOddorEven() { }

@Before

public void setUp() {

eoo = new OddorEven();

}

@After

public void tearDown() {

eoo=null;

}

@Test

public void OddorEvenTest()

{

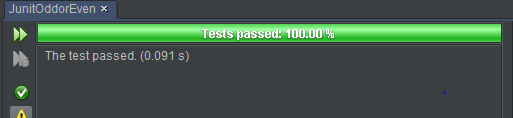
assertEquals(eoo.even\_or\_odd(6), "Even");

assertEquals(eoo.even\_or\_odd(7),"Odd");

}

}

**Output:**



3.Fibonacci Series

**Source Code:**

**Fibonacci.java:**

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

public class fibonacci {

public fibonacci() {

}

static long fibonacciCalc(int n)

{

assert(n>0) && (n<=91):"n out of range";

long curr,prev,past;

if((n==1)|| (n==2))

return 1;

curr=prev=1;

for(int i=3;i<=n;i++)

{

past=prev;

prev=curr;

curr=past+prev;

}

return curr;

}

@Test

public void TestFibonacci()

{

assertEquals(1,fibonacciCalc(1));

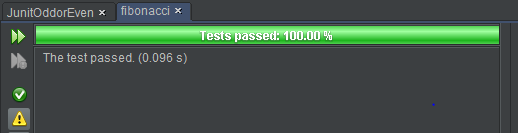
assertEquals(1,fibonacciCalc(2));

assertEquals(7778742049L,fibonacciCalc(49));

}

}

**Output:**



4. Explicit Constructor

**Source Code:**

**ExplicitConstructor.java:**

import static org.hamcrest.CoreMatchers.is;

import org.hamcrest.Matcher;

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

public class ExplicitConstructor{

public ExplicitConstructor() {}

@Test

public void constructorTest()

{

Calculator c=new Calculator(4,5);

asserThat(9,c.sum());

}

@Test(expected=IllegalArgumentException.class)

public void constructorExceptionTest()

{

Calculator c=new Calculator(-4,5);

}

private void asserThat(int i, int is) {

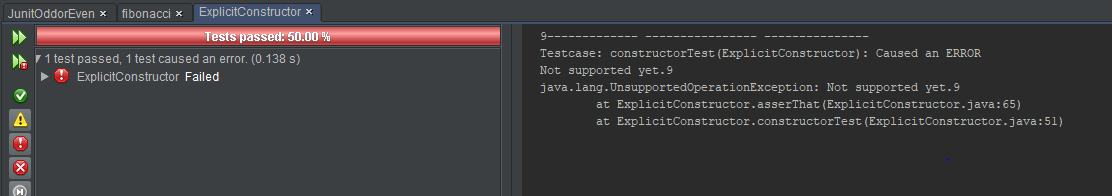
System.out.print(i);

throw new UnsupportedOperationException("Not supported yet."+i); //To change body of generated methods, choose Tools | Templates.

}

}

**Output:**



1. List Comparison

**Source Code:**

**Listcmpr.java:**

import java.util.Arrays;

import java.util.List;

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Assert;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

public class Listcmpr {

public Listcmpr() { }

@Test

public void ListTest()

{

List<Integer>list1;

List<Integer>list2;

List<Integer>list3;

list1=Arrays.asList(1,2,3,4);

list2=Arrays.asList(1,2,3,4);

list3=Arrays.asList(1,2,2,4);

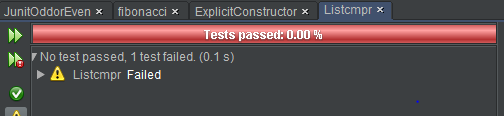
Assert.assertArrayEquals(list1.toArray(),list2.toArray());

Assert.assertArrayEquals(list1.toArray(),list3.toArray());

}

}

**Output:**



6. NULL OR NOT NULL TEST

**Source Code:**

**NullTest.java:**

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

public class NullTest {

public NullTest() {

}

@Test

public void NULLTEST()

{

String data=null;

int number2=2;

assertNull(data);

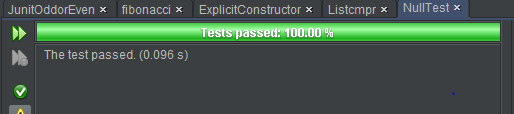
assertNotNull(2);

assertNull(null);

}

}

**Output:**



7. Login Validation

**SOURCE CODE:**

**Login.java:**

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

public class Login {

public Login() { }

@Test

public void LoginTest()

{

boolean logintest=true;

String username="hari";

String password="pass123";

assertNotEquals("please enter username",username,"");

assertNotEquals("please enter password",password,"");

assertEquals("please Enter correct Username","hari",username);

assertEquals("please Enter correct password","pass12",password);

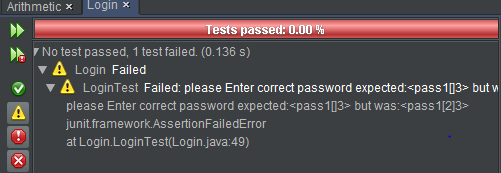
System.out.println("Logged in Successfully");

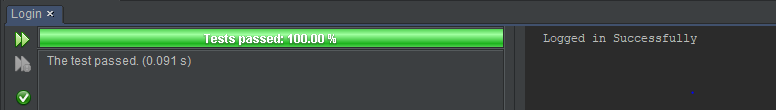
}

}

**Output:**

When expectedpassword=”pass12”;ActualPassword=”pass123”





8.  Calculator program that performs the basic arithmetic operations

**SOURCE CODE:**

**ArithmeticOperation.java:**

class ArithmeticOperation {

int num1;

int num2;

ArithmeticOperation(int n1,int n2)

{

num1=n1;

num2=n2;

}

public int add()

{

return this.num1+this.num2;

}

public int sub()

{

return this.num1-this.num2;

}

public int mul()

{

return this.num1\*this.num2;

}

public int div()

{

int res = 0;

try{

res=this.num1/this.num2;

}

catch(ArithmeticException e)

{

System.out.println("The second no is zero");

}

return res;

}

}

**Arithmetic.java:**

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

public class Arithmetic {

public Arithmetic() {}

@Test

public void Arithmeticopr()

{

ArithmeticOperation a=new ArithmeticOperation(1,2);

assertEquals("Error while adding the nummber",3,a.add());

assertEquals("Error while subracting the nummber",-1,a.sub());

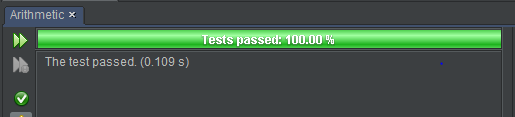
assertEquals("Error while Multiplying the nummber",2,a.mul());

assertEquals("Error while dividing the nummber",0,a.div());

}

}

**Output:**



9. Write a program to list out the various annotations in JUnit which provide the following information about test methods

->Which methods are going to run before and after test methods.

->Which methods run before and after all the methods

->Which methods or classes will be ignored during the execution.

**SOURCE CODE:**

**Annotations.java:**

import org.junit.After;

import org.junit.AfterClass;

import org.junit.Before;

import org.junit.BeforeClass;

import org.junit.Test;

import static org.junit.Assert.\*;

public class annotations {

public annotations() { }

@BeforeClass

public static void setUpClass() {

System.out.println("1-> Tt Runs Once Before the Class");

}

@AfterClass

public static void tearDownClass() {

System.out.println("5-> It Runs Once after the Class");

}

@Before

public void setUp() {

System.out.println("2-> It Runs Before the Test Method is called");

}

@After

public void tearDown() {

System.out.println("4-> It Runs after the Test Method is called");

}

@Test

public void testfun()

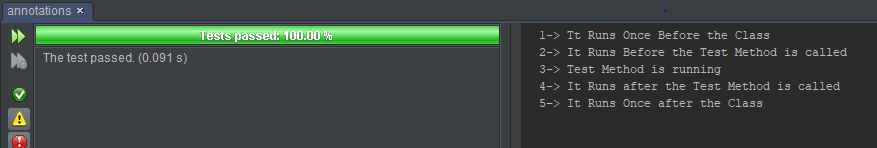
{

System.out.println("3-> Test Method is running");

}

}

**Output:**



10)Create EmployeeDetails.java in junit workspace which is a POJO class.

* EmployeeDetails class is used to -

o get/set the value of employee's name.

o get/set the value of employee's monthly salary.

o get/set the value of employee's age.

* Create a file called EmpBusinessLogic.java in junit workspace, which contains the business logic.

EmpBusinessLogic class is used for calculating −

o the yearly salary of an employee.

o the appraisal amount of an employee.

Create a file called TestEmployeeDetails.java in junit workspace, which contains the test cases to be tested. TestEmployeeDetails class is used for testing the methods of EmpBusinessLogic class. It

o tests the yearly salary of the employee.

o tests the appraisal amount of the employee.

Create a java class filed named TestRunner.java in junit workspace to execute test case(s).

**SOURCE CODE:**

**EmpBusinessLogic.java:**

public class EmpBusinessLogic {

// Calculate the yearly salary of employee

public double calculateYearlySalary(EmployeeDetails employeeDetails){

double yearlySalary=0;

yearlySalary = employeeDetails.getMonthlySalary() \* 12;

return yearlySalary;

}

// Calculate the appraisal amount of employee

public double calculateAppraisal(EmployeeDetails employeeDetails){

double appraisal=0;

if(employeeDetails.getMonthlySalary() <= 50000){

appraisal = 5000;

}else{

appraisal = 7000;

}

return appraisal;

}

}

**EmployeeDetails.java:**

public class EmployeeDetails {

private String name;

private double monthlySalary;

private int age;

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public double getMonthlySalary() {

return monthlySalary;

}

public void setMonthlySalary(double monthlySalary) {

this.monthlySalary = monthlySalary;

}

public int getAge() {

return age;

}

public void setAge(int age) {

this.age = age;

}

}

**TestEmployeeDetails.java:**

import org.junit.Test;

import static org.junit.Assert.assertEquals;

public class TestEmployeeDetails {

EmpBusinessLogic empBusinessLogic =new EmpBusinessLogic();

EmployeeDetails employee = new EmployeeDetails();

//test to check appraisal

@Test

public void testCalculateAppriasal() {

employee.setName("Jagapradeep");

employee.setAge(20);

employee.setMonthlySalary(11000);

double appraisal= empBusinessLogic.calculateAppraisal(employee);

assertEquals(5000, appraisal, 0.0);

}

// test to check yearly salary

@Test

public void testCalculateYearlySalary() {

employee.setName("Karan");

employee.setAge(29);

employee.setMonthlySalary(52000);

double salary= empBusinessLogic.calculateYearlySalary(employee);

assertEquals(624000, salary, 0.0);

}

}

**TestRunner.java:**

import org.junit.runner.JUnitCore;

import org.junit.runner.Result;

import org.junit.runner.notification.Failure;

public class TestRunner {

public static void main(String[] args) {

Result result = JUnitCore.runClasses(TestEmployeeDetails.class);

for (Failure failure : result.getFailures()) {

System.out.println(failure.toString());

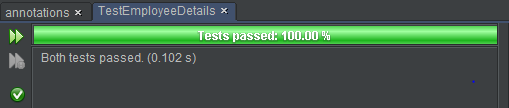
}

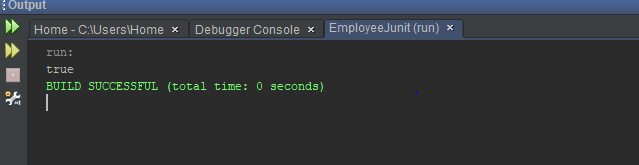
System.out.println(result.wasSuccessful());

}

}

**Output:**





11.Create Studentdetails.java in junit workspace, which is a POJO class.

StudentDetails class is used to –[10]

o get/set the value of student's name.

o get/set the value of student's marks.

o get/set the value of student's average.

ii) Create a file called StuBusinessLogic.java in junit workspace, which

contains the business logic.

StuBusinessLogic class is used for calculating −

● the grade point of a student .

● the Rank of the student .

Create a file called TestStudentDetails.java in junit workspace, which

contains the test cases to be tested.

TestStudentDetails class is used for testing the methods

of StuBusinessLogic class. It

o the grade point of a student .

o the Rank of the student .

create a java class filed named TestRunner.java in junit workspace to

execute test case(s).

**Source Code:**

**StudentDetails.java:**

public class StudentDetails {

private String Name;

private double marks[]=new double[5],total;

private double average;

public double gradepoints;

public void setGP(double n)

{

gradepoints=n;

}

public void setName(String n)

{

Name=n;

}

public String getName()

{

return this.Name;

}

public void setMarks(double []m)

{

int i;

for(i=0;i<5;i++)

{

marks[i]=m[i];

}

}

public double[] getMarks()

{

return marks;

}

public void setAverage()

{

int i,sum=0;

for(i=0;i<5;i++)

{

sum+=marks[i];

}

total=sum;

average=sum/i;

}

public double getAverage()

{

return average;

}

public double getTotal()

{

return total;

}

public void setgradepoints(double g)

{

gradepoints=g;

}}

**StuBusinessLogic:**

import java.util.Arrays;

public class StuBusinessLogic {

public double CalculateGradepoints(StudentDetails s)

{

double gradepoints=(s.getTotal()/100)\*2;

s.setGP(gradepoints);

return s.gradepoints;

}

public int CalculateRank(StudentDetails d[],StudentDetails student)

{

StudentDetails s=new StudentDetails();

StudentDetails[] c=new StudentDetails[d.length];

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

{

c[i]=d[i];

}

System.out.print(c[0].gradepoints);

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

{

for(int j=i+1;j<c.length;j++)

{

if(c[i].gradepoints<c[j].gradepoints)

{

s=c[i];

c[i]=c[j];

c[j]=s;

}

}

}

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

{

System.out.println(c[i].gradepoints);

if(c[i]==student)

return i+1;

}

return c.length;

}

}

**TestStudentDetails.java:**

import org.junit.Test;

import static org.junit.Assert.\*;

import java.util.\*;

public class TestStudentDetails {

StuBusinessLogic stuBusinessLogic =new StuBusinessLogic();

StudentDetails[] students = new StudentDetails[3];

public TestStudentDetails() {

double[] marks ;

marks=new double[5];

students[0]=new StudentDetails();

students[1]=new StudentDetails();

students[2]=new StudentDetails();

students[0].setName("Abilash");

marks[0]=98;

marks[1]=80;

marks[2]=70;

marks[3]=60;

marks[4]=90;

students[0].setMarks(marks);

students[0].setAverage();

students[1].setName("Abinaya");

marks[0]=97;

marks[1]=80;

marks[2]=70;

marks[3]=90;

marks[4]=90;

students[1].setMarks(marks);

students[1].setAverage();

students[2].setName("Karan");

marks[0]=97;

marks[1]=80;

marks[2]=70;

marks[3]=90;

marks[4]=90;

students[2].setMarks(marks);

students[2].setAverage();

}

@Test

public void Test\_CalculateGp\_\_Rank\_Student() {

assertEquals(7.96, stuBusinessLogic.CalculateGradepoints(students[0]),0.001);

assertEquals(8.54, stuBusinessLogic.CalculateGradepoints(students[1]),0.001);

assertEquals(8.54, stuBusinessLogic.CalculateGradepoints(students[2]),0.001);

assertEquals(3, stuBusinessLogic.CalculateRank(students,students[0]),0.001);

assertEquals(1, stuBusinessLogic.CalculateRank(students,students[1]),0.001);

assertEquals(2, stuBusinessLogic.CalculateRank(students,students[2]),0.001);

}

}

**TestRunner.java:**

import org.junit.runner.JUnitCore;

import org.junit.runner.Result;

import org.junit.runner.notification.Failure;

public class TestRunner {

public static void main(String[] args) {

Result result = JUnitCore.runClasses(TestStudentDetails.class);

for (Failure failure : result.getFailures()) {

System.out.println(failure.toString());

}

System.out.println(result.wasSuccessful());

}

}

**Output:**

