

BCS3343 – SOFTWARE TESTING AND MAINTENANCE

**JUNIT Project**

OnlineTest

Lecture’s Name

MDM ROZLINA BINTI MOHAMED

Member’s Name

RAAJALETCHUMY A/P RAJAH CB11115

NORFATINFAIZAH BINTI ABD. RAHIM CB13011

**1) System Overview**

The TestOnline.java program supports functional testing of the programs. TestOnline.java runs a complete set (file) of test cases, and produces a file of test case execution results. The Functional Test Case Executor has three programs that we can test here. The programs are Constract1IT.java, Constract2IT.java and HomePageTest.java. This system will be test all the case whereby we want to find the solution for the case either all tests are ‘pass’ or ‘fail’. But some other situation, the system might be failed due to it failures to get the solution. So, not all of our inputs are wrong and maybe some of it is because of the system failure

**2) System Test Suite**

**A) Assuming the range of Grade A, B, C, D and E take the following ranges (with minimum**

80 ≤ A ≤ 100

60 ≤ B < 80

50 ≤ C < 60

40 ≤ D < 50

0 ≤ E <40

EP: A) = {-9, -1, 0, 41, 4, 40, 95}

B) = {-1, 0, 65, 35, 29, 15, 30}

C) = {25, 10, 27, -8, -1, 0, 26}

D) = {0, 45, 88, 3, 46, 21, 9}

E) = {-1, 10, 58, 70, -10, 54, -7}

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Conditions** | **Test ID** | **A** | **B** | **C** | **D** | **E** | **Expected Results** | **Actual Results** | **Grade** |
| 80 ≤ A ≤ 100  60 ≤ B < 80  50 ≤ C < 60  40 ≤ D < 50  0 ≤ E <40 | 1 | -9 | -1 | 25 | 0 | -1 | No result | No result | - |
| 2 | -1 | 0 | 10 | 45 | 10 | Result | Fail | E |
| 3 | 0 | 65 | 27 | 88 | 58 | Result | Pass | B |
| 4 | 41 | 35 | -8 | 3 | 70 | No result | No result | - |
| 5 | 4 | 29 | -1 | 46 | -10 | Result | Pass | D |
| 6 | 40 | 15 | 0 | 21 | 54 | No result | No result | - |
| 7 | 95 | 30 | 26 | -9 | -7 | Result | Pass | A |

**3) File TestOnline Package Coding**

***Constract1IT.java***

/\*\*

\* Test of getName method, of class Constract1.

\*/

@Test

public void testGetName() {

String name ="aiman";

String name1= "aiman";

System.out.println("getName");

Constract1 instance = new Constract1();

try{

instance.setName(name);

String expResult = "aiman";

String result = instance.getName();

assertEquals(expResult, result);

System.out.println( "RGetName "+result+": validate");

}

catch(Throwable t)

{

System.out.println( "RGetName : Not validate");

}

try{

instance.setName(name1);

String expResult = "feza";

String result = instance.getName();

assertEquals(expResult, result);

System.out.println( "RGetName "+result+": validate");

}

catch(Throwable t)

{

System.out.println( "RGetName : Not validate");

}

// TODO review the generated test code and remove the default call to fail.

//fail("The test case is a prototype.");

}

/\*\*

\* Test of getId method, of class Constract1.

\*/

@Test

public void testGetId() {

String id = "cb13006";

String id1 = "cb13006";

System.out.println("getId");

Constract1 instance = new Constract1();

try{

instance.setId(id);

String expResult = "cb13006";

String result = instance.getId();

assertEquals(expResult, result);

System.out.println( "RGetId "+result+": validate");

}

catch(Throwable t)

{

System.out.println( "RGetId : Not validate");

}

try{

instance.setId(id1);

String expResult = "aa12122";

String result = instance.getId();

assertEquals(expResult, result);

System.out.println( "RGetId "+result+": validate");

}

catch(Throwable t)

{

System.out.println( "RGetId : Not validate");

}

// TODO review the generated test code and remove the default call to fail.

//fail("The test case is a prototype.");

}

/\*\*

\* Test of getSection method, of class Constract1.

\*/

@Test

public void testGetSection() {

String section ="dcs22";

String section1= "dcs22";

System.out.println("getSection");

Constract1 instance = new Constract1();

try{

instance.setSection(section);

String expResult = "dcs22";

String result = instance.getSection();

assertEquals(expResult, result);

System.out.println( "RGetSection "+result+": validate");

}

catch(Throwable t)

{

System.out.println( "RGetSection : Not validate");

}

try{

instance.setSection(section1);

String expResult = "dcs12";

String result = instance.getSection();

assertEquals(expResult, result);

System.out.println( "RGetSection "+result+": validate");

}

catch(Throwable t)

{

System.out.println( "RGetSection : Not validate");

}

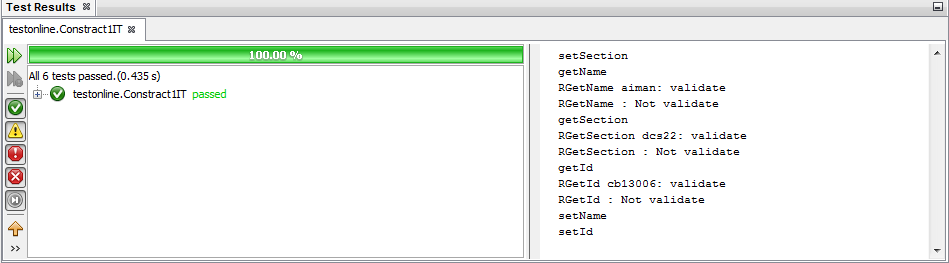
// TODO review the generated test code and remove the default call to fail.

//fail("The test case is a prototype.");

}

}

**Output Test Result**



**Figure 3.1** Test Result Constract1IT

***Constract2Test.java***

@Test

public void testGetGrade() {

float Q12 = 1;

float Q13 = 1;

float Q14 = 1;

System.out.println("getGrade");

Constract2 instance = new Constract2();

try{

instance.setPage4(Q12, Q13, Q14);

String expResult = "E";

String result = instance.getGrade();

assertEquals(expResult, result);

System.out.println( "RGetGrade "+result+": validate");

}

catch(Throwable t)

{

System.out.println( "RGetGrade : Not validate");

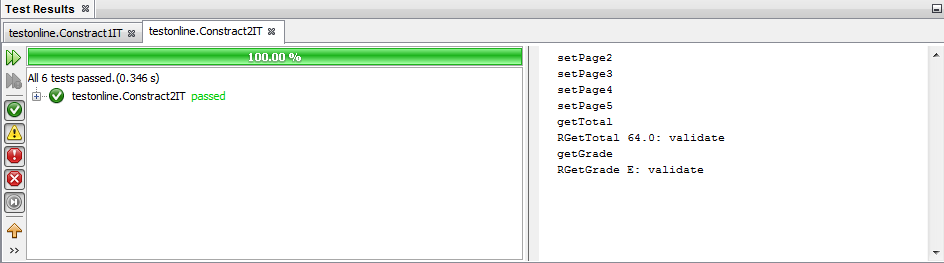
}

// TODO review the generated test code and remove the default call to fail.

//fail("The test case is a prototype.");

}

**Output Test Result**



**Figure 3.2** Test Result Constract2IT