File for the final project

Student id : 13251450 Name : Harshal Jaspal

**Question no d)**

**BVA for method dms()**

**[Test cases]**

|  |  |  |  |
| --- | --- | --- | --- |
| case | parameter | Boundary value | Test |
| 1 | d(degree) | Int\_min | 1 |
| 2 |  | -359 | 2 |
| 3 |  | -360 | 6 |
| 4 |  | 359 | 4 |
| 5 |  | 360 | 8 |
| 6 |  | Int\_max | 3 |
| 7 | M(minutes) | int\_min | 1 |
| 8 |  | -59 | 5 |
| 9 |  | -60 | 6 |
| 10 |  | 59 | 3 |
| 11 |  | 60 | 7 |
| 12 |  | Int\_max | 4 |
| 13 | S(seconds) | Int\_min | 5 |
| 14 |  | -59 | 2 |
| 15 |  | -60 | 1 |
| 16 |  | 59 | 6 |
| 17 |  | 60 | 8 |
| 18 |  | Int\_max | 3 |
| 19 | Return value | sign(d)\*(abs(d)\*3600+abs(m\*60+(abs(s))) | 1 |

[Test Data ]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Inputs | | |  |
| ID | Test\_case\_covered | d | m | s | Expected Output |
| 1 | 1,7,17,19 | Int\_min | Int\_min | -60 | -(intmin\*3600+(intmin\*60+60)) |
| 2 | 2,7,14,19 | -359 | Int\_min | -59 | -(1292400+(intmin\*60+59)) |
| 3 | 6,10,18,19 | Int\_max | 59 | Int\_max | (intmax\*3600+(3540+intmax)) |
| 4 | 4,7,12,19 | 359 | Int\_max | Int\_max | (359\*3600+(intmax\*60+intmax)) |
| 5 | 1,8,13,19 | Int\_min | -59 | Int\_min | -(intmin\*3600+(59\*60+intmin)) |
| 6 | 3,9,16,19 | -360 | -60 | 59 | -1299659 |
| 7 | 4,11,14,19 | 359 | 60 | -59 | 1296059 |
| 8 | 5,8,17,19 | 360 | -59 | 60 | 1299600 |
| 9 | 2,10,16,19 | -359 | 59 | 59 | -1295999 |
| 10 | 2,9,17,19 | -359 | -60 | 60 | -1296060 |
| 11 | 3,10,16,19 | -360 | 59 | 59 | -1299599 |
| 12 | 4,10,14,19 | 359 | 59 | -59 | 1295999 |

***BVA for setLatitude***

[Test cases]

|  |  |  |  |
| --- | --- | --- | --- |
| Case | parameter | Boundary value | Test |
| 1 | Value | Int\_min/3600\*3600 | 13 |
| 2 |  | -90\*3600 | 14 |
| 3 |  | -89\*3600 | 15 |
| 4 |  | 89\*3600 | 16 |
| 5 |  | 90\*3600 | 17 |
| 6 |  | Int\_max/3600\*3600 | 18 |
|  |  |  |  |
| 7 | Return Value | value | 16 |
| 8 |  | Integer.max\_value | 18 |
|  |  |  |  |

[Test Data]

ID Test\_Case\_Covered Value Expected Output

|  |  |  |  |
| --- | --- | --- | --- |
| 13 | 1,7 | (Int\_min/3600)\*3600 | Integer.max\_value |
| 14 | 2,7 | -90\*3600 | Integer.max\_value |
| 15 | 3,6 | -89\*3600 | -320400 |
| 16 | 4,7 | 89\*3600 | 320400 |
| 17 | 5,6 | 90\*3600 | Integer.max\_value |
| 18 | 6,7 | (Int\_max/3600)\*3600 | Integer.max\_value |
|  |  |  |  |
|  |  |  |  |

**BV for setLongitude**

**[Test Cases]**

|  |  |  |  |
| --- | --- | --- | --- |
| Case | parameter | Boundary value | Test |
| 1 | Value | Int\_min/3600\*3600 | 19 |
| 2 |  | -360\*3600 | 20 |
| 3 |  | -359\*3600 | 21 |
| 4 |  | 359\*3600 | 22 |
| 5 |  | 360\*3600 | 23 |
| 6 |  | Int\_max/3600\*3600 | 24 |
|  |  |  |  |
| 7 | Return Value | value | 22 |
| 8 |  | Integer.max\_value | 24 |

**[Test Data]**

**ID Test cases covered Input Output**

|  |  |  |  |
| --- | --- | --- | --- |
| 19 | 1,7 | (int\_min/3600)\*3600 | Integer.max\_value |
| 20 | 2,7 | -360\*3600 | Integer.max\_value |
| 21 | 3,6 | -359\*3600 | -1292400 |
| 22 | 4,7 | 359\*3600 | 1292400 |
| 23 | 5,6 | 360\*3600 | Integer.max\_value |
| 24 | 6,7 | Int\_max/3600\*3600 | Integer.max\_value |

**BV for login()**

**[Test Case]**

|  |  |  |  |
| --- | --- | --- | --- |
| **Case** | **Parameter** | **Boundary value** | **Test** |
| **1** | **Pwd** | **Blank-password** | **25** |
| **2** |  | **pwd** | **26** |
| **3** | **Return value** | **true** | **25** |
| **4** |  | **false** | **26** |

**[Test data]**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test\_cases\_covered** | **Input(password)** | **Expected Output** |
| **25** | **1,3** | **Blank-password** | **true** |
| **26** | **2,4** | **pwd** | **false** |
|  |  |  |  |
|  |  |  |  |

**Question no e)**

**[Test case]**

**Truth table for isAT() , getLOngOffset() and getLatoffset()**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **------------------------------------rules-----------------------------------------------------** | | | | | | | |
|  | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **Causes** |  |  |  |  |  |  |  |  |
| **TestMode** | **False** | **false** | **false** | **false** | **true** | **true** | **true** | **True** |
| -2< (longitude-targetLong)<=2 | **False** | **False** | **True** | **True** | **False** | **True** | **False** | **True** |
| -2< (latitude-targetLat)<=2 | **False** | **True** | **False** | **True** | **False** | **False** | **True** | **True** |
| **Effects** |  |  |  |  |  |  |  |  |
| **nearby** | **False** | **False** | **False** | **True** | **False** | **False** | **False** | **True** |

**\*assuming that login is always true**

**[Test data]**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Inputs** | | | **Expected Output** | | |
| **Id** | **Test\_cases\_covered** | longitude-targetLong | latitude-targetLat | **Test\_mode** | **nearby** | **offsetlong** | **offsetlat** |
| **27** | **1** | **(-6,36,12)-**  **(-6,36,9)** | **(53,22,60)-**  **(53,22,57)** | **False** | **False** | **\*** | **\*** |
| **28** | **2** | **(-6,36,12)-**  **(-6,36,9)** | **(53,22,57)-**  **(53,22,57)** | **False** | **False** | **\*** | **\*** |
| **29** | **3** | **(-6,36,11)-**  **(-6,36,9)** | **(53,22,63)-**  **(53,22,57)** | **False** | **False** | **\*** | **\*** |
| **30** | **4** | **(-6,36,9)-**  **(-6,36,9)** | **(53,22,59)-**  **(53,22,57)** | **False** | **True** | **0** | **2** |
| **31** | **5** | **(0,0,5)-**  **(0,0,0)** | **(51,28,70)-**  **(51,28,44)** | **True** | **False** | **\*** | **\*** |
| **32** | **6** | **(0,0,2)-**  **(0,0,0)** | **(51,28,50)-**  **(51,28,44)** | **True** | **False** | **\*** | **\*** |
| **33** | **7** | **(0,0,8)-**  **(0,0,0)** | **(51,28,45)-**  **(51,28,44)** | **True** | **False** | **\*** | **\*** |
| **34** | **8** | **(0,0,1)-**  **(0,0,0)** | **(51,28,46)-**  **(51,28,44)** | **True** | **True** | **1** | **2** |

**Question no i:**

**MC/DC coverage for line 181**

**if ( (test || loggedIn) && (abs(longitude-targetLong)<=maxd) &&**

**(abs(latitude-targetLat)<=maxd) )**

**[Test case]**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Inputs** | | | |  |
| **case** | **Testmode** | **LoggedIN** | **(abs(longitude-targetLong)<=maxd)** | **(abs(latitude-targetLat)<=maxd)** | **Test** |
| 1 | true | false | false | true | 1 |
| 2 | true | false | true | true | 2 |
| 3 | false | false | true | true | 3 |
| 4 | false | true | true | true | 4 |
| 5 | false | true | true | false | 5 |
|  |  |  |  |  |  |

**[Test Data]**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **Inputs** | | | |  |
| **ID** | **Test\_cases\_**  **covered** | **Test**  **mode** | **LoggedIN** | **(abs(longitude-targetLong)<=maxd)** | **(abs(latitude-targetLat)<=maxd)** | **Output** |
| 35 | 1 | true | Blank | (45,54,21)-(0,0,0) | (51,28,43)-(51,28,44) | False |
| 36 | 2 | true | Blank | (0,0,2)-(0,0,0) | (51,28,46)-(51,28,44) | True |
| 37 | 3 | false | Blank | (-36,6,11)-(-6,36,9) | (53,22,58)-(53,22,57) | False |
| 38 | 4 | false | blank-password | (-6,36,11)-(-6,36,9) | (53,22,58)-(53,22,57) | True |
| 39 | 5 | false | blank-password | (-6,36,11)-(-6,36,9) | (53,22,70)-(53,22,57) | false |

**Question no f**

**State machine testing for location server**

**[Test case]**

|  |  |  |
| --- | --- | --- |
| **case** | **Transition** | **Test** |
| **S1** | **1** | **41** |
| **S2** | **2** | **40,41** |
| **S3** | **3** | **40,41** |
| **S4** | **4** | **40,41** |
| **S5** | **5** | **40,41** |
| **S6** | **6** | **40,41** |
| **S7** | **7** | **40,41** |
| **S8** | **8** | **40,41** |

**[Test Data]**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Test cases Covered** | **Inputs** | **Expected Output(nearby)** |
| **40** | **S2,s3,s4,s5,s6,s7,s8** | **LocationServer(true)**  **setLatitude(185324)**  **setLongitude(0)**  **getLatitude()**  **getLongitude()** | **Non null**  **183524**  **0**  **183524**  **0** |
| **41** | **S1,s3,s4,s5,s6,s7,s8** | **LocationServer(false)**  **Login(blank-password)**  **setLatitude(dms(192177)**  **setLongitude(dms(-23769)**  **getLatitude()**  **getLongitude()** | **-1**  **true**  **192177**  **-23769**  **192177**  **-23769** |

**Extra test cases added**

**[Test Case]**

|  |  |  |  |
| --- | --- | --- | --- |
| Case | parameter | Input value | Test |
| 1 | Value | 241200 | 42 |
| 2 |  | -241200 | 43 |

**[Test Data]**

|  |  |  |  |
| --- | --- | --- | --- |
| Case | Test\_cases\_covered | Input value | Expected output |
| 42 | 1 | 241200 | 241200 |
| 43 | 2 | -241200 | -241200 |

**J-Unit class code**

**package cs608;**

**import static org.junit.Assert.\*;**

**import org.junit.Test;**

**public class LocationServerTest {**

**LocationServer p=new LocationServer(true);**

**//BVA test for function dms**

**@Test**

**public void test1() {**

**assertEquals(-(Integer.MIN\_VALUE\*3600+(Integer.MIN\_VALUE\*60+60)) , LocationServer.dms (Integer.MIN\_VALUE,Integer.MIN\_VALUE,-60));**

**}**

**@Test**

**public void test2() {**

**assertEquals(-(1292400+(Integer.MIN\_VALUE\*60+59)) , LocationServer.dms (-359,Integer.MIN\_VALUE,-59));**

**}**

**@Test**

**public void test3() {**

**assertEquals( (Integer.MAX\_VALUE\*3600+(3540+Integer.MAX\_VALUE)), LocationServer.dms (Integer.MAX\_VALUE,59,Integer.MAX\_VALUE));**

**}**

**@Test**

**public void test4() {**

**assertEquals( (1292400+(Integer.MAX\_VALUE\*60+Integer.MAX\_VALUE)), LocationServer.dms (359,Integer.MAX\_VALUE,Integer.MAX\_VALUE));**

**}**

**@Test**

**public void test5() {**

**assertEquals(-(Integer.MIN\_VALUE\*3600+(59\*60+Integer.MIN\_VALUE)) , LocationServer.dms(Integer.MIN\_VALUE,-59,Integer.MIN\_VALUE));**

**}**

**@Test**

**public void test6() {**

**assertEquals(-1299659 , LocationServer.dms (-360,-60,59));**

**}**

**@Test**

**public void test7() {**

**assertEquals(1296059 , LocationServer.dms (359,60,-59));**

**}**

**@Test**

**public void test8() {**

**assertEquals(1299600 , LocationServer.dms (360,-59,60));**

**}**

**@Test**

**public void test9() {**

**assertEquals(-1295999, LocationServer.dms (-359,59,59));**

**}**

**@Test**

**public void test10() {**

**assertEquals(-1296060 , LocationServer.dms (-359,-60,60));**

**}**

**@Test**

**public void test11() {**

**assertEquals(-1299599 , LocationServer.dms (-360,59,59));**

**}**

**@Test**

**public void test12() {**

**assertEquals(1295999 , LocationServer.dms (359,59,-59));**

**}**

**//bva test for function setLatitude**

**@Test**

**public void test13() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLatitude(Integer.MIN\_VALUE);**

**assertEquals(Integer.MAX\_VALUE, ls.getLatitude ());**

**}**

**@Test**

**public void test14() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLatitude(-324000);**

**assertEquals(Integer.MAX\_VALUE , ls.getLatitude ());**

**}**

**@Test**

**public void test15() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLatitude(-320400);**

**assertEquals(-320400, ls.getLatitude ());**

**}**

**@Test**

**public void test16() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLatitude(320400);**

**assertEquals(320400, ls.getLatitude ());**

**}**

**@Test**

**public void test17() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLatitude(324000);**

**assertEquals(Integer.MAX\_VALUE, ls.getLatitude ());**

**}**

**@Test**

**public void test18() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLatitude(Integer.MAX\_VALUE);**

**assertEquals(Integer.MAX\_VALUE , ls.getLatitude ());**

**}**

**//BVA for function setLongitude**

**@Test**

**public void test19() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLongitude(Integer.MIN\_VALUE);**

**assertEquals(Integer.MAX\_VALUE, ls.getLongitude ());**

**}**

**@Test**

**public void test20() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLongitude(-1296000);**

**assertEquals(Integer.MAX\_VALUE , ls.getLongitude ());**

**}**

**@Test**

**public void test21() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLongitude(-1292400);**

**assertEquals(-1292400, ls.getLongitude ());**

**}**

**@Test**

**public void test22() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLongitude(1292400);**

**assertEquals(1292400, ls.getLongitude ());**

**}**

**@Test**

**public void test23() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLongitude(1296000);**

**assertEquals(Integer.MAX\_VALUE, ls.getLongitude ());**

**}**

**@Test**

**public void test24() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLongitude(Integer.MAX\_VALUE);**

**assertEquals(Integer.MAX\_VALUE , ls.getLongitude ());**

**}**

**//BVA for login()**

**@Test**

**public void test25() {**

**LocationServer ls=new LocationServer(true);**

**assertEquals(true , ls.login ("blank-password"));**

**}**

**@Test**

**public void test26() {**

**LocationServer ls=new LocationServer(true);**

**assertEquals(false , ls.login ("pwd"));**

**}**

**//combinational testing**

**@Test**

**public void test27() {**

**LocationServer ls =new LocationServer(false);**

**ls.login("blank-password");**

**ls.setLongitude(ls.dms(-6, 36, 12));**

**ls.setLatitude(ls.dms(53, 22, 60));**

**ls.getLongitude();**

**ls.getLatitude();**

**assertEquals(false,ls.isAt());**

**}**

**@Test**

**public void test28() {**

**LocationServer ls =new LocationServer(false);**

**ls.login("blank-password");**

**ls.setLongitude(ls.dms(-6, 36, 12));**

**ls.setLatitude(ls.dms(53, 22, 57));**

**ls.getLongitude();**

**ls.getLatitude();**

**assertEquals(false,ls.isAt());**

**}**

**@Test**

**public void test29() {**

**LocationServer ls =new LocationServer(false);**

**ls.login("blank-password");**

**ls.setLongitude(ls.dms(-6, 36, 11));**

**ls.setLatitude(ls.dms(53, 22, 63));**

**ls.getLongitude();**

**ls.getLatitude();**

**assertEquals(false,ls.isAt());**

**}**

**@Test**

**public void test30() {**

**LocationServer ls =new LocationServer(false);**

**ls.login("blank-password");**

**ls.setLongitude(ls.dms(-6, 36, 9));**

**ls.setLatitude(ls.dms(53, 22, 59));**

**ls.getLongitude();**

**ls.getLatitude();**

**ls.getLatOffset();**

**ls.getLongOffset();**

**assertEquals(true,ls.isAt());**

**}**

**@Test**

**public void test31() {**

**LocationServer ls =new LocationServer(true);**

**ls.login("blank-password");**

**ls.setLongitude(ls.dms(0, 0, 5));**

**ls.setLatitude(ls.dms(51, 28, 70));**

**ls.getLongitude();**

**ls.getLatitude();**

**assertEquals(false,ls.isAt());**

**}**

**@Test**

**public void test32() {**

**LocationServer ls =new LocationServer(true);**

**ls.login("blank-password");**

**ls.setLongitude(ls.dms(0, 0, 2));**

**ls.setLatitude(ls.dms(51, 28, 50));**

**ls.getLongitude();**

**ls.getLatitude();**

**assertEquals(false,ls.isAt());**

**}**

**@Test**

**public void test33() {**

**LocationServer ls =new LocationServer(true);**

**ls.login("blank-password");**

**ls.setLongitude(ls.dms(0, 0, 8));**

**ls.setLatitude(ls.dms(51, 28, 45));**

**ls.getLongitude();**

**ls.getLatitude();**

**assertEquals(false,ls.isAt());**

**}**

**@Test**

**public void test34() {**

**LocationServer ls =new LocationServer(true);**

**ls.login("blank-password");**

**ls.setLongitude(ls.dms(0, 0, 1));**

**ls.setLatitude(ls.dms(51, 28, 46));**

**ls.getLongitude();**

**ls.getLatitude();**

**ls.getLatOffset();**

**ls.getLongOffset();**

**assertEquals(true,ls.isAt());**

**}**

**//----------------------MC/DC for isAt()-----------------------------**

**@Test**

**public void test35() {**

**LocationServer ls =new LocationServer(true);**

**ls.login("blank");**

**ls.setLongitude(ls.dms(-45, 54, 21));**

**ls.setLatitude(ls.dms(51, 28, 43));**

**ls.getLongitude();**

**ls.getLatitude();**

**assertEquals(false,ls.isAt());**

**}**

**@Test**

**public void test36() {**

**LocationServer ls =new LocationServer(true);**

**ls.login("blank");**

**ls.setLongitude(ls.dms(0, 0, 2));**

**ls.setLatitude(ls.dms(51, 28, 46));**

**ls.getLongitude();**

**ls.getLatitude();**

**assertEquals(true,ls.isAt());**

**}**

**@Test**

**public void test37() {**

**LocationServer ls =new LocationServer(false);**

**ls.login("blank");**

**ls.setLongitude(ls.dms(-36, 6, 11));**

**ls.setLatitude(ls.dms(53, 22, 58));**

**ls.getLongitude();**

**ls.getLatitude();**

**assertEquals(false,ls.isAt());**

**}**

**@Test**

**public void test38() {**

**LocationServer ls =new LocationServer(false);**

**ls.login("blank-password");**

**ls.setLongitude(ls.dms(-6, 36, 11));**

**ls.setLatitude(ls.dms(53, 22, 58));**

**ls.getLongitude();**

**ls.getLatitude();**

**assertEquals(true,ls.isAt());**

**}**

**@Test**

**public void test39() {**

**LocationServer ls =new LocationServer(false);**

**ls.login("blank-password");**

**ls.setLongitude(ls.dms(-6, 36, 11));**

**ls.setLatitude(ls.dms(53, 22, 70));**

**ls.getLongitude();**

**ls.getLatitude();**

**assertEquals(false,ls.isAt());**

**}**

**//--------------------------------state machine testing------------------------**

**@Test**

**public void test40()**

**{**

**LocationServer ls = new LocationServer(true);**

**ls.setLatitude(ls.dms(51, 28, 43));**

**ls.setLongitude(ls.dms(0, 0, 0));**

**ls.getLatitude();**

**ls.getLongitude();**

**assertEquals(true, ls.isAt());**

**}**

**@Test**

**public void test41()**

**{**

**LocationServer ls = new LocationServer(false);**

**ls.setLatitude(ls.dms(51, 21, 57));**

**ls.setLongitude(ls.dms(-6, 36, 9));**

**ls.getLatitude();**

**ls.getLongitude();**

**assertEquals(false, ls.isAt());**

**}**

**//---------------------cases added to satisfy coverage -----------**

**@Test**

**public void test42() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLatitude(241200);**

**assertEquals(241200, ls.getLatitude ());**

**}**

**@Test**

**public void test43() {**

**LocationServer ls=new LocationServer(true);**

**ls.setLatitude(-241200);**

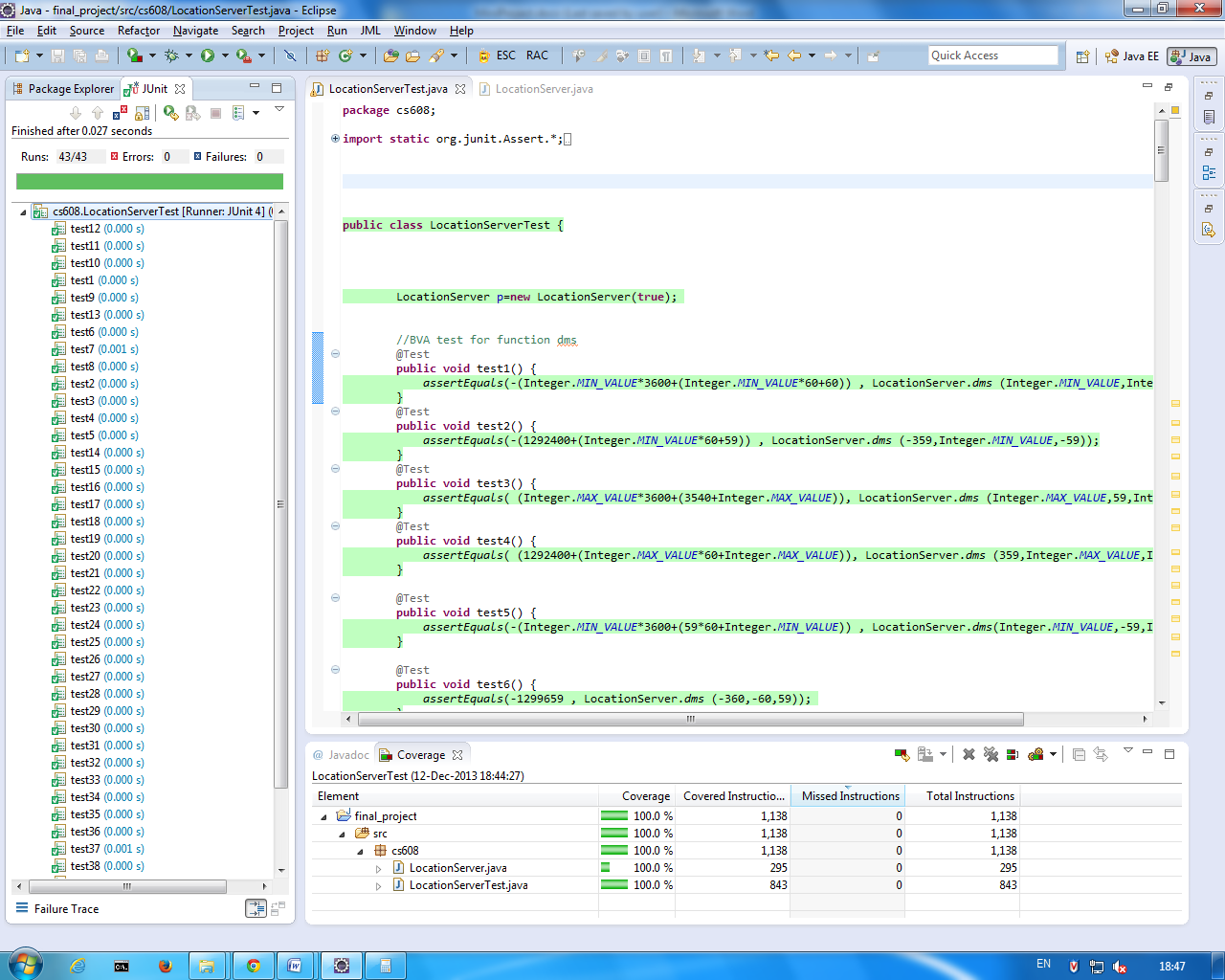
**assertEquals(-241200, ls.getLatitude ());**

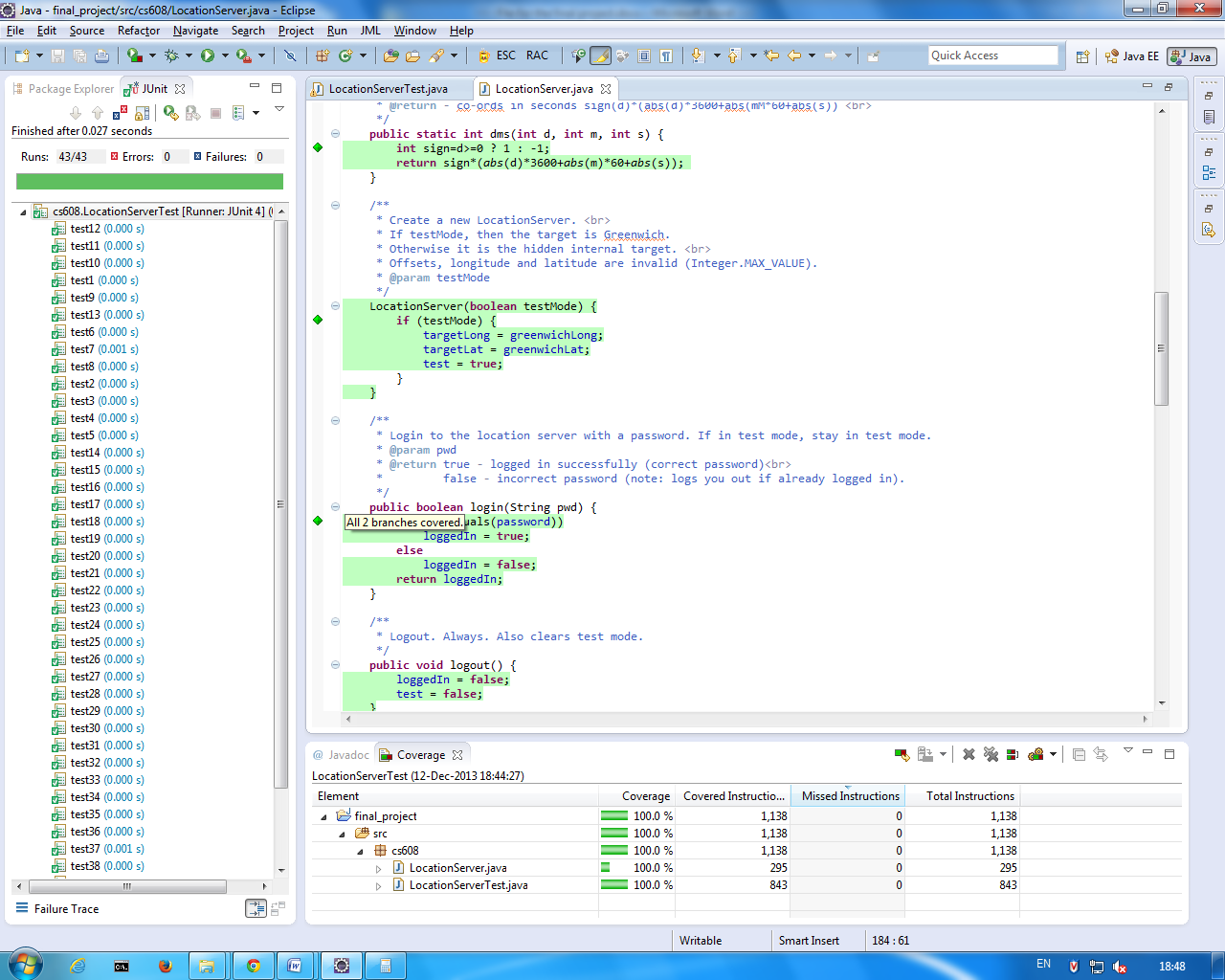
**ls.logout();**

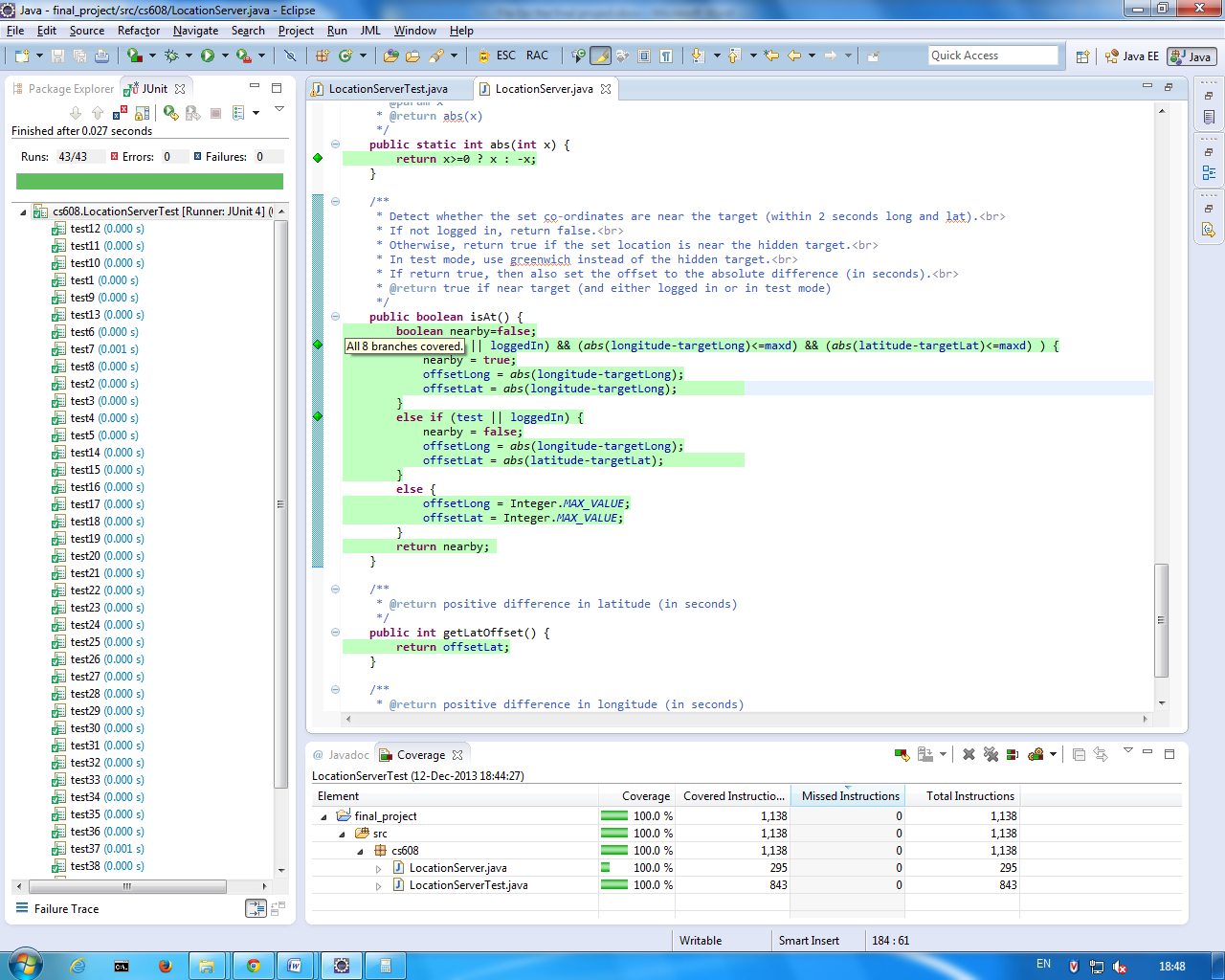
**}**

**}**

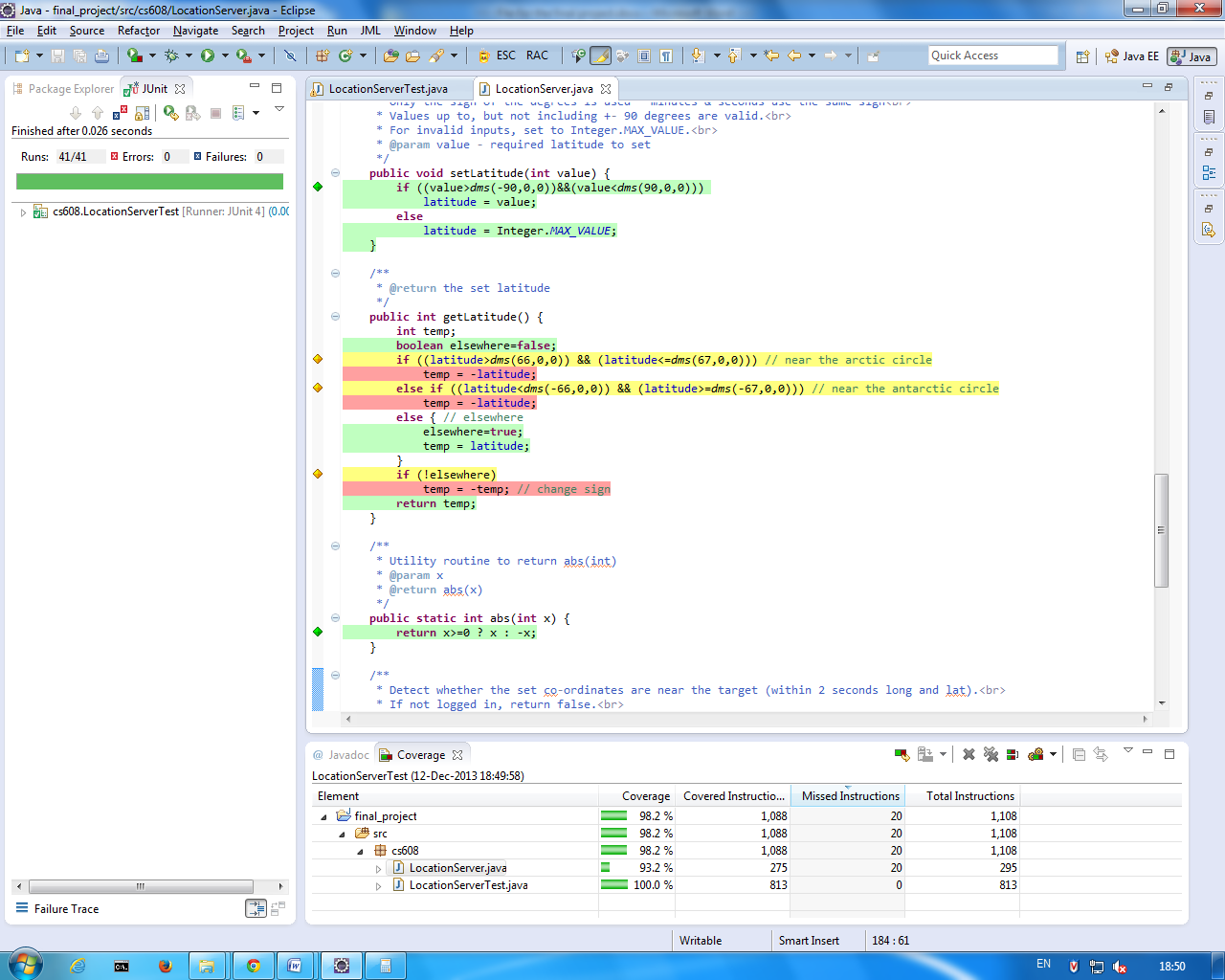
**Screenshots:**

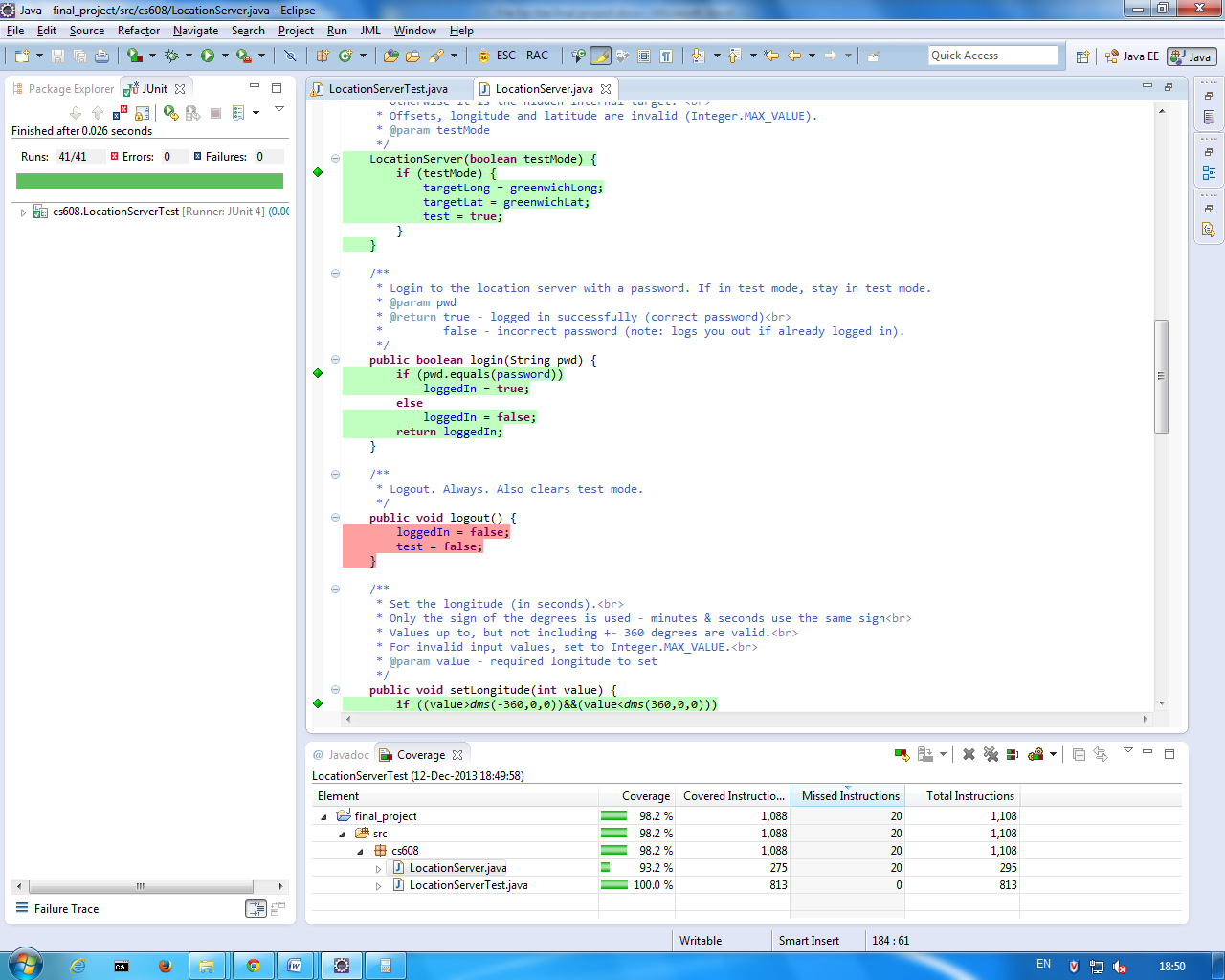
****

****

****

**Branch coverage without the use of exception cases**

****

****

**Without exception cases the coverage get reduced to 93.2%**

* **Because**

**The logout function has not been called anywhere in the code , so to execute the function i added the logout function in a tes case so that it can be covered in coverage**

* **Because in getlatitude function when the latitude is near to arctic circle or near to antartic circle , the the co-ordinate values are not getting covered in any case**