

DirectorControllerTest.java X

SourceHistory

44

@Test

45

public void testValidUser() {

46

System.out.println("validUser");

47

String user = "MAJOF";

48

String password = "12345";

49

DirectorController instance = new DirectorController();

50

boolean expResult = true;

51

boolean result = instance.validUser(user, password);

52

assertEquals(expResult, result);

53

54

}

55

56

@Test

57

public void testValidUser1() {

58

System.out.println("validUser");

59

String user = "MAJOF";

60

String password = "123457";

61

DirectorController instance = new DirectorController();

62

boolean expResult = true;

63

boolean result = instance.validUser(user, password);

64

assertEquals(expResult, result);

65

66

}

67

68

@Test

69

public void testValidUser2() {

70

System.out.println("validUser");

71

String user = "MAJO";

72

String password = "12345";

73

DirectorController instance = new DirectorController();

74

boolean expResult = true;

75

boolean result = instance.validUser(user, password);


76

assertEquals(expResult, result);

77

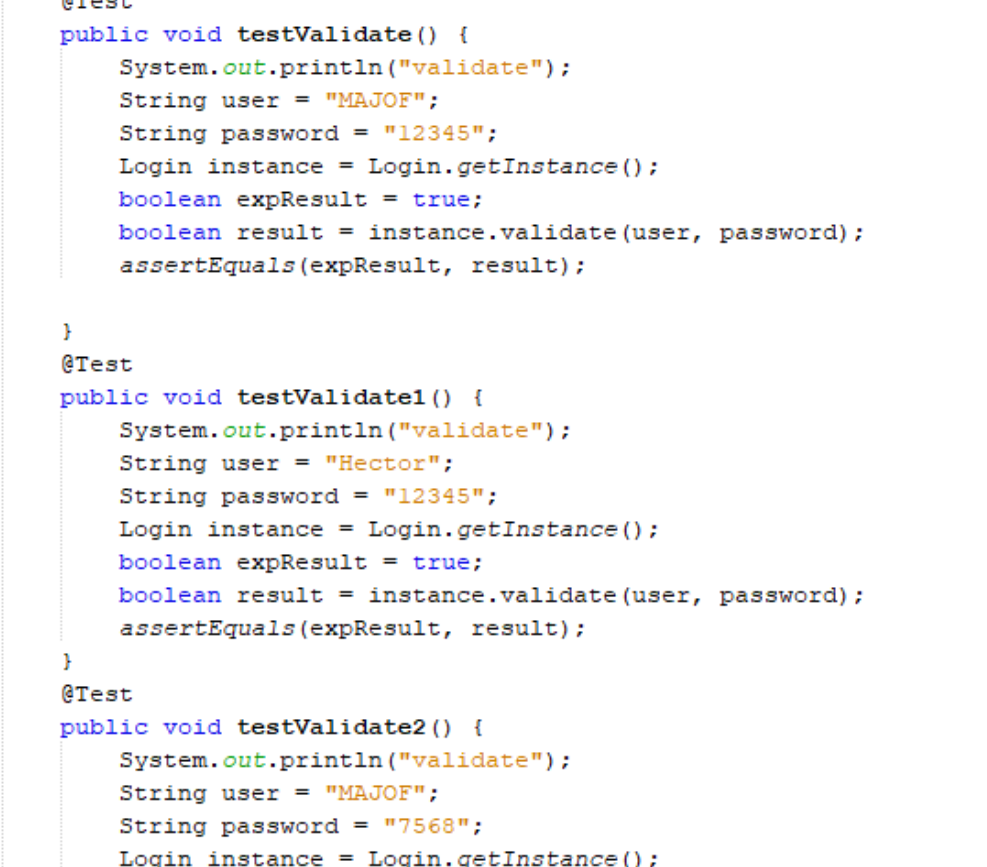
DirectorControllerTest.java X

SourceHistory



```
80      @Test
81      public void testValidUser3() {
82          System.out.println("validUser");
83          String user = "MAJO";
84          String password = "123457";
85          DirectorController instance = new DirectorController();
86          boolean expResult = true;
87          boolean result = instance.validUser(user, password);
88          assertEquals(expResult, result);
89      }
90
91
92      @Test
93      public void testValidUser4() {
94          System.out.println("validUser");
95          String user = "Pedro";
96          String password = "Pedro";
97          DirectorController instance = new DirectorController();
98          boolean expResult = true;
99          boolean result = instance.validUser(user, password);
100          assertEquals(expResult, result);
101      }
102
103
104      @Test
105      public void testValidUser5() {
106          System.out.println("validUser");
107          String user = "12345";
108          String password = "12345";
109          DirectorController instance = new DirectorController();
110          boolean expResult = true;
111          boolean result = instance.validUser(user, password);
112          assertEquals(expResult, result);
113      }
```

```
DirectorControllerTest.java x
Source History
137
138 }
139
140 /**
141  * Test of approveMotion method, of class Director.
142  */
143 @Test
144 public void testApproveMotion() {
145     System.out.println("approveMotion");
146     char answer = ' ';
147     DirectorController instance = new DirectorController();
148     boolean expResult = false;
149     boolean result = instance.approveMotion(answer);
150     assertEquals(expResult, result);
151 }
152
153
154 @Test
155 public void testApproveMotion1() {
156     System.out.println("approveMotion");
157     char answer = ' ';
158     DirectorController instance = new DirectorController();
159     boolean expResult = true;
160     boolean result = instance.approveMotion(answer);
161     assertEquals(expResult, result);
162 }
163
164 }
165
166
```



```
46  @Test
47  public void testValidate() {
48      System.out.println("validate");
49      String user = "MAJOF";
50      String password = "12345";
51      Login instance = Login.getInstance();
52      boolean expResult = true;
53      boolean result = instance.validate(user, password);
54      assertEquals(expResult, result);
55  }
56
57  @Test
58  public void testValidate1() {
59      System.out.println("validate");
60      String user = "Hector";
61      String password = "12345";
62      Login instance = Login.getInstance();
63      boolean expResult = true;
64      boolean result = instance.validate(user, password);
65      assertEquals(expResult, result);
66  }
67
68  @Test
69  public void testValidate2() {
70      System.out.println("validate");
71      String user = "MAJOF";
72      String password = "7568";
73      Login instance = Login.getInstance();
74      boolean expResult = true;
75      boolean result = instance.validate(user, password);
76      assertEquals(expResult, result);
77  }
```

LoginTest.java x

SourceHistory

```
76     }
77     @Test
78     public void testValidate3() {
79         System.out.println("validate");
80         String user = "Carlos";
81         String password = "76214";
82         Login instance = Login.getInstance();
83         boolean expectedResult = true;
84         boolean result = instance.validate(user, password);
85         assertEquals(expResult, result);
86     }
87     @Test
88     public void testValidate4() {
89         System.out.println("validate");
90         String user = "MAJOF";
91         String password = "12345";
92         Login instance = Login.getInstance();
93         boolean expectedResult = false;
94         boolean result = instance.validate(user, password);
95         assertEquals(expResult, result);
96     }
97     @Test
98     public void testValidate5() {
99         System.out.println("validate");
100        String user = "Majof";
101        String password = "123457";
102        Login instance = Login.getInstance();
103        boolean expectedResult = true;
104        boolean result = instance.validate(user, password);
105        assertEquals(expResult, result);
106    }
```

LoginTest.java x

SourceHistory

106

}

107

@Test

108

public void testValidate6() {

109

System.out.println("validate");

110

String user = "user";

111

String password = "password";

112

Login instance = Login.getInstance();

113

boolean expResult = true;

114

boolean result = instance.validate(user, password);

115

assertEquals(expResult, result);

116

}

117

@Test

118

public void testValidate7() {

119

System.out.println("validate");

120

String user = "Rick";

121

String password = "rick";

122

Login instance = Login.getInstance();

123

boolean expResult = true;

124

boolean result = instance.validate(user, password);

125

assertEquals(expResult, result);

126

}

127

@Test

128

public void testValidate8() {

129

System.out.println("validate");

130

String user = "Juan";

131

String password = "4814162342";

132

Login instance = Login.getInstance();

133

boolean expResult = true;

134


boolean result = instance.validate(user, password);

135

assertEquals(expResult, result);

136

}




```
136     }
137     @Test
138     public void testValidate9() {
139         System.out.println("validate");
140         String user = "MAJOF7";
141         String password = "42";
142         Login instance = Login.getInstance();
143         boolean expResult = true;
144         boolean result = instance.validate(user, password);
145         assertEquals(expResult, result);
146     }
147     @Test
148     public void testValidate10() {
149         System.out.println("validate");
150         String user = "Majof";
151         String password = "456373434834835438763534";
152         Login instance = Login.getInstance();
153         boolean expResult = true;
154         boolean result = instance.validate(user, password);
155         assertEquals(expResult, result);
156     }
157     @Test
158     public void testValidate11() {
159         System.out.println("validate");
160         String user = "Melissa";
161         String password = "12345";
162         Login instance = Login.getInstance();
163         boolean expResult = false;
164         boolean result = instance.validate(user, password);
165         assertEquals(expResult, result);
166     }
```



```
PolyclinicControllerTest.java x
Source History
47 @Test
48 public void testShowMedicaments() {
49     System.out.println("showMedicaments");
50     File fileName = new File("Request.json");
51     String record = "Bryan Pazmino";
52     PolyclinicController instance = new PolyclinicController();
53     instance.showMedicaments(fileName, record);
54 }
55
56 @Test
57 public void testShowMedicaments1() {
58     System.out.println("showMedicaments");
59     File fileName = new File("Request.json");
60     String record = "dthndj";
61     PolyclinicController instance = new PolyclinicController();
62     instance.showMedicaments(fileName, record);
63 }
64
65 @Test
66 public void testShowMedicaments2() {
67     System.out.println("showMedicaments");
68     File fileName = new File("DoctorData.json");
69     String record = "Viviana Gordon";
70     PolyclinicController instance = new PolyclinicController();
71     instance.showMedicaments(fileName, record);
72 }
73
74 @Test
75 public void testShowMedicaments3() {
76     System.out.println("showMedicaments");
77     File fileName = new File("DoctorData.json");
78     String record = "Ivan Beltran";
79     PolyclinicController instance = new PolyclinicController();
80     instance.showMedicaments(fileName, record);
```

PolyclinicControllerTest.java X

SourceHistory




```
83     @Test
84     public void testShowMedicaments4() {
85         System.out.println("showMedicaments");
86         File fileName = new File("Request.json");
87         String record = "Allison Cedeno";
88         PolyclinicController instance = new PolyclinicController();
89         instance.showMedicaments(fileName, record);
90     }
91
92     @Test
93     public void testShowMedicaments5() {
94         System.out.println("showMedicaments");
95         File fileName = new File("ClinicHistory.json");
96         String record = "Bryan Pazmino";
97         PolyclinicController instance = new PolyclinicController();
98         instance.showMedicaments(fileName, record);
99     }
100
101     @Test
102     public void testShowMedicaments6() {
103         System.out.println("showMedicaments");
104         File fileName = new File("MedicalCheck.json");
105         String record = "22/02/2021";
106         PolyclinicController instance = new PolyclinicController();
107         instance.showMedicaments(fileName, record);
108     }
109
110     @Test
111     public void testShowMedicaments7() {
112         System.out.println("showMedicaments");
113         File fileName = new File("DoctorData.json");
114         String record = "Rosa Mosquera";
115         PolyclinicController instance = new PolyclinicController();
116         instance.showMedicaments(fileName, record);
```

```
PolyclinicControllerTest.java x
Source History
119 @Test
120 public void testShowMedicaments8() {
121     System.out.println("showMedicaments");
122     File fileName = new File("DoctorData.json");
123     String record = "Mateo Beltran";
124     PolyclinicController instance = new PolyclinicController();
125     instance.showMedicaments(fileName, record);
126 }
127 }
128 @Test
129 public void testShowMedicaments9() {
130     System.out.println("showMedicaments");
131     File fileName = new File("DoctorData.json");
132     String record = "ClinicLaboratory";
133     PolyclinicController instance = new PolyclinicController();
134     instance.showMedicaments(fileName, record);
135 }
136 }
137 @Test
138 public void testShowMedicaments10() {
139     System.out.println("showMedicaments");
140     File fileName = new File("DoctorData.json");
141     String record = "Morning";
142     PolyclinicController instance = new PolyclinicController();
143     instance.showMedicaments(fileName, record);
144 }
145 }
146 public void testShowMedicaments12() {
147     System.out.println("showMedicaments");
148     File fileName = new File("DoctorData.json");
149     String record = "Odontology";
150     PolyclinicController instance = new PolyclinicController();
151     instance.showMedicaments(fileName, record);
152 }
```

PolyclinicControllerTest.java X

Source History



```
163     @Test
164     public void testShowMedicaments14() {
165         System.out.println("showMedicaments");
166         File fileName = new File("DoctorData.json");
167         String record = "ClinicLaboratory";
168         PolyclinicController instance = new PolyclinicController();
169         instance.showMedicaments(fileName, record);
170
171     }
172     @Test
173     public void testShowMedicaments15() {
174         System.out.println("showMedicaments");
175         File fileName = new File("Request.json");
176         String record = "Bryan Pazmino";
177         PolyclinicController instance = new PolyclinicController();
178         instance.showMedicaments(fileName, record);
179
180     }
181     @Test
182     public void testShowMedicaments16() {
183         System.out.println("showMedicaments");
184         File fileName = new File("Request.json");
185         String record = "L00354835";
186         PolyclinicController instance = new PolyclinicController();
187         instance.showMedicaments(fileName, record);
188
189     }
190     @Test
191     public void testShowMedicaments17() {
192         System.out.println("showMedicaments");
193         File fileName = new File("Request.json");
194         String record = "L00354835";
195         PolyclinicController instance = new PolyclinicController();
196         instance.showMedicaments(fileName, record);
```


VirtualCardControllerTest.java X

SourceHistory

106

@Test

107

public void testGiveWay6() {

108

System.out.println("giveWay");

109

int dataID = 78;

110

VirtualCardController instance = new VirtualCardController();

111

boolean expResult = false;

112

boolean result = instance.giveWay(dataID);

113

assertEquals(expResult, result);

114

115

}

116

@Test

117

public void testGiveWay7() {

118

System.out.println("giveWay");

119

int dataID = 12;

120

VirtualCardController instance = new VirtualCardController();

121

boolean expResult = false;

122

boolean result = instance.giveWay(dataID);

123

assertEquals(expResult, result);

124

125

}

126

@Test

127

public void testGiveWay8() {

128

System.out.println("giveWay");

129

int dataID = 90;

130

VirtualCardController instance = new VirtualCardController();

131

boolean expResult = false;

132

boolean result = instance.giveWay(dataID);

133

assertEquals(expResult, result);


134

135

}

VirtualCardControllerTest.java X


SourceHistory



```
136     @Test
137     public void testGiveWay9() {
138         System.out.println("giveWay");
139         int dataID = 78;
140         VirtualCardController instance = new VirtualCardController();
141         boolean expResult = false;
142         boolean result = instance.giveWay(dataID);
143         assertEquals(expResult, result);
144     }
145
146     @Test
147     public void testGiveWay10() {
148         System.out.println("giveWay");
149         int dataID = 57;
150         VirtualCardController instance = new VirtualCardController();
151         boolean expResult = false;
152         boolean result = instance.giveWay(dataID);
153         assertEquals(expResult, result);
154     }
155
156     @Test
157     public void testGiveWay11() {
158         System.out.println("giveWay");
159         int dataID = 76;
160         VirtualCardController instance = new VirtualCardController();
161         boolean expResult = false;
162         boolean result = instance.giveWay(dataID);
163         assertEquals(expResult, result);
164     }
165 }
```

VirtualCardControllerTest.java X

SourceHistory



```
166     @Test
167     public void testGiveWay12() {
168         System.out.println("giveWay");
169         int dataID = 46;
170         VirtualCardController instance = new VirtualCardController();
171         boolean expResult = false;
172         boolean result = instance.giveWay(dataID);
173         assertEquals(expResult, result);
174     }
175
176     @Test
177     public void testGiveWay13() {
178         System.out.println("giveWay");
179         int dataID = 56;
180         VirtualCardController instance = new VirtualCardController();
181         boolean expResult = false;
182         boolean result = instance.giveWay(dataID);
183         assertEquals(expResult, result);
184     }
185
186     @Test
187     public void testGiveWay14() {
188         System.out.println("giveWay");
189         int dataID = 900;
190         VirtualCardController instance = new VirtualCardController();
191         boolean expResult = false;
192         boolean result = instance.giveWay(dataID);
193         assertEquals(expResult, result);
194     }
195 }
```



VirtualCardControllerTest.java X

SourceHistory

```
196     @Test
197     public void testGiveWay15() {
198         System.out.println("giveWay");
199         int dataID = 43;
200         VirtualCardController instance = new VirtualCardController();
201         boolean expResult = false;
202         boolean result = instance.giveWay(dataID);
203         assertEquals(expResult, result);
204     }
205
206     @Test
207     public void testGiveWay16() {
208         System.out.println("giveWay");
209         int dataID = 0;
210         VirtualCardController instance = new VirtualCardController();
211         boolean expResult = false;
212         boolean result = instance.giveWay(dataID);
213         assertEquals(expResult, result);
214     }
215
216     @Test
217     public void testGiveWay17() {
218         System.out.println("giveWay");
219         int dataID = 54;
220         VirtualCardController instance = new VirtualCardController();
221         boolean expResult = false;
222         boolean result = instance.giveWay(dataID);
223         assertEquals(expResult, result);
224     }
225 }
```

VirtualCardControllerTest.java X

SourceHistory




```
226     @Test
227     public void testGiveWay18() {
228         System.out.println("giveWay");
229         int dataID = 73;
230         VirtualCardController instance = new VirtualCardController();
231         boolean expResult = false;
232         boolean result = instance.giveWay(dataID);
233         assertEquals(expResult, result);
234     }
235
236     @Test
237     public void testGiveWay19() {
238         System.out.println("giveWay");
239         int dataID = 74;
240         VirtualCardController instance = new VirtualCardController();
241         boolean expResult = false;
242         boolean result = instance.giveWay(dataID);
243         assertEquals(expResult, result);
244     }
245
246     @Test
247     public void testGiveWay20() {
248         System.out.println("giveWay");
249         int dataID = 735;
250         VirtualCardController instance = new VirtualCardController();
251         boolean expResult = false;
252         boolean result = instance.giveWay(dataID);
253         assertEquals(expResult, result);
254     }
255 }
```

```
AdministratorTest.java X
Source History
46  L      */
47      @Test
48      public void testGetDeparment() {
49          System.out.println("getDeparment");
50          Administrator instance = new AdministratorImpl();
51          instance.setDeparment("Computer's science");
52          String expResult = "Computer's science";
53          String result = instance.getDeparment();
54          assertEquals(expResult, result);
55      }
56
57      @Test
58      public void testGetDeparment1() {
59          System.out.println("getDeparment");
60          Administrator instance = new AdministratorImpl();
61          instance.setDeparment("Computer's science");
62          String expResult = "Computer's scienc";
63          String result = instance.getDeparment();
64          assertEquals(expResult, result);
65      }
66
67      @Test
68      public void testGetDeparment2() {
69          System.out.println("getDeparment");
70          Administrator instance = new AdministratorImpl();
71          instance.setDeparment("Computer's science");
72          String expResult = "Computer'sscience";
73          String result = instance.getDeparment();
74          assertEquals(expResult, result);
75      }
```

AdministratorTest.java X

SourceHistory



```
76
77     @Test
78     public void testGetDeparment3() {
79         System.out.println("getDeparment");
80         Administrator instance = new AdministratorImpl();
81         instance.setDeparment("exact Sciences");
82         String expResult = "exact Sciences";
83         String result = instance.getDeparment();
84         assertEquals(expResult, result);
85     }
86     @Test
87     public void testGetDeparment4() {
88         System.out.println("getDeparment");
89         Administrator instance = new AdministratorImpl();
90         instance.setDeparment("exact Sciences");
91         String expResult = "exactSciences";
92         String result = instance.getDeparment();
93         assertEquals(expResult, result);
94     }
95     @Test
96     public void testGetDeparment5() {
97         System.out.println("getDeparment");
98         Administrator instance = new AdministratorImpl();
99         instance.setDeparment("Computer's science");
100        String expResult = "computer's science";
101        String result = instance.getDeparment();
102        assertEquals(expResult, result);
103    }
```

AdministratorTest.java x

SourceHistory

103

}

104

@Test

105

public void testGetDeparment6() {

106

System.out.println("getDeparment");

107

Administrator instance = new AdministratorImpl();

108

instance.setDeparment("Computer's Science");

109

String expResult = "computer's science";

110

String result = instance.getDeparment();

111

assertEquals(expResult, result);

112

}

113

@Test

114

public void testGetDeparment7() {

115

System.out.println("getDeparment");

116

Administrator instance = new AdministratorImpl();

117

instance.setDeparment("exact Sciences");

118

String expResult = "exactSciences";

119

String result = instance.getDeparment();

120

assertEquals(expResult, result);

121

}

122

@Test

123

public void testGetDeparment8() {

124

System.out.println("getDeparment");

125

Administrator instance = new AdministratorImpl();

126

instance.setDeparment("exact sciences");

127

String expResult = "exact Sciences";

128

String result = instance.getDeparment();

129

assertEquals(expResult, result);

130

}

AreaTest.java

SourceHistory

46

@Test

47

public void testIsAvailability() {

48

System.out.println("isAvailability");

49

Area instance = new Area();

50

boolean availability = true;

51

boolean expResult = availability;

52

boolean result = instance.isAvailability();

53

assertEquals(expResult, result);

54

}

55

@Test

56

public void testIsAvailability1() {

57

System.out.println("isAvailability");

58

Area instance = new Area();

59

boolean availability = false;

60

boolean expResult = availability;

61

boolean result = instance.isAvailability();

62

assertEquals(expResult, result);

63

}

64

/**

65

* Test of getArea method, of class Area.

66

*/

67

@Test

68

public void testGetArea() {

69

System.out.println("getArea");

70

Area instance = new Area();

71

instance.setArea("general medicine");

72

String expResult = "general medicine";

73

String result = instance.getArea();


74

assertEquals(expResult, result);

75

76

}



```
AreaTest.java
Source History
76     }
77     @Test
78     public void testGetArea1() {
79         System.out.println("getArea");
80         Area instance = new Area();
81         instance.setArea("General medicine");
82         String expResult = "general medicine";
83         String result = instance.getArea();
84         assertEquals(expResult, result);
85     }
86     @Test
87     public void testGetArea2() {
88         System.out.println("getArea");
89         Area instance = new Area();
90         instance.setArea("general medicine");
91         String expResult = "general Medicine";
92         String result = instance.getArea();
93         assertEquals(expResult, result);
94     }
95     @Test
96     public void testGetArea3() {
97         System.out.println("getArea");
98         Area instance = new Area();
99         instance.setArea("general medicine");
100        String expResult = "generalMedicine";
101        String result = instance.getArea();
102        assertEquals(expResult, result);
103    }
104    }
105    }
106 }
```

DiagnosisTest.java x

SourceHistory

46

@Test

47

public void testGetPatientName() {

48

System.out.println("getPatientName");

49

Diagnosis instance = new Diagnosis();

50

instance.setPatientName("Juan Castillo");

51

String expResult = "Juan Castillo";

52

String result = instance.getPatientName();

53

assertEquals(expResult, result);

54

55

}

56

@Test

57

public void testGetPatientName1() {

58

System.out.println("getPatientName");

59

Diagnosis instance = new Diagnosis();

60

instance.setPatientName("juan Castillo");

61

String expResult = "Juan Castillo";

62

String result = instance.getPatientName();

63

assertEquals(expResult, result);

64

65

}

66

@Test

67

public void testGetPatientName2() {

68

System.out.println("getPatientName");

69

Diagnosis instance = new Diagnosis();

70

instance.setPatientName("Melissa Peralta");

71

String expResult = "Melisa Peralta";

72

String result = instance.getPatientName();

73

assertEquals(expResult, result);

74

75

}

-

DiagnosisTest.java X

SourceHistory

```
76      @Test
77      public void testGetPatientName3() {
78          System.out.println("getPatientName");
79          Diagnosis instance = new Diagnosis();
80          instance.setPatientName("Pedro Castillo");
81          String expResult = "pedro Castillo";
82          String result = instance.getPatientName();
83          assertEquals(expResult, result);
84      }
85
86      @Test
87      public void testGetPatientName4() {
88          System.out.println("getPatientName");
89          Diagnosis instance = new Diagnosis();
90          instance.setPatientName("Juan Castillo");
91          String expResult = "Juan";
92          String result = instance.getPatientName();
93          assertEquals(expResult, result);
94      }
95
96      @Test
97      public void testGetPatientName5() {
98          System.out.println("getPatientName");
99          Diagnosis instance = new Diagnosis();
100         instance.setPatientName("Juan Castillo");
101         String expResult = "";
102         String result = instance.getPatientName();
103         assertEquals(expResult, result);
104     }
105 }
```

DirectorTest.java X

SourceHistory

46

@Test

47

public void testGetCareer() {

48

System.out.println("getCareer");

49

Director instance = new Director();

50

instance.setCareer("Information technology");

51

String expResult = "Information technology";

52

String result = instance.getCareer();

53

assertEquals(expResult, result);

54

55

}

56

@Test

57

public void testGetCareer1() {

58

System.out.println("getCareer");

59

Director instance = new Director();

60

instance.setCareer("Information Technology");

61

String expResult = "Information technology";

62

String result = instance.getCareer();

63

assertEquals(expResult, result);

64

65

}

66

@Test

67

public void testGetCareer2() {

68

System.out.println("getCareer");

69

Director instance = new Director();

70

instance.setCareer("information technology");

71

String expResult = "Information technology";

72

String result = instance.getCareer();

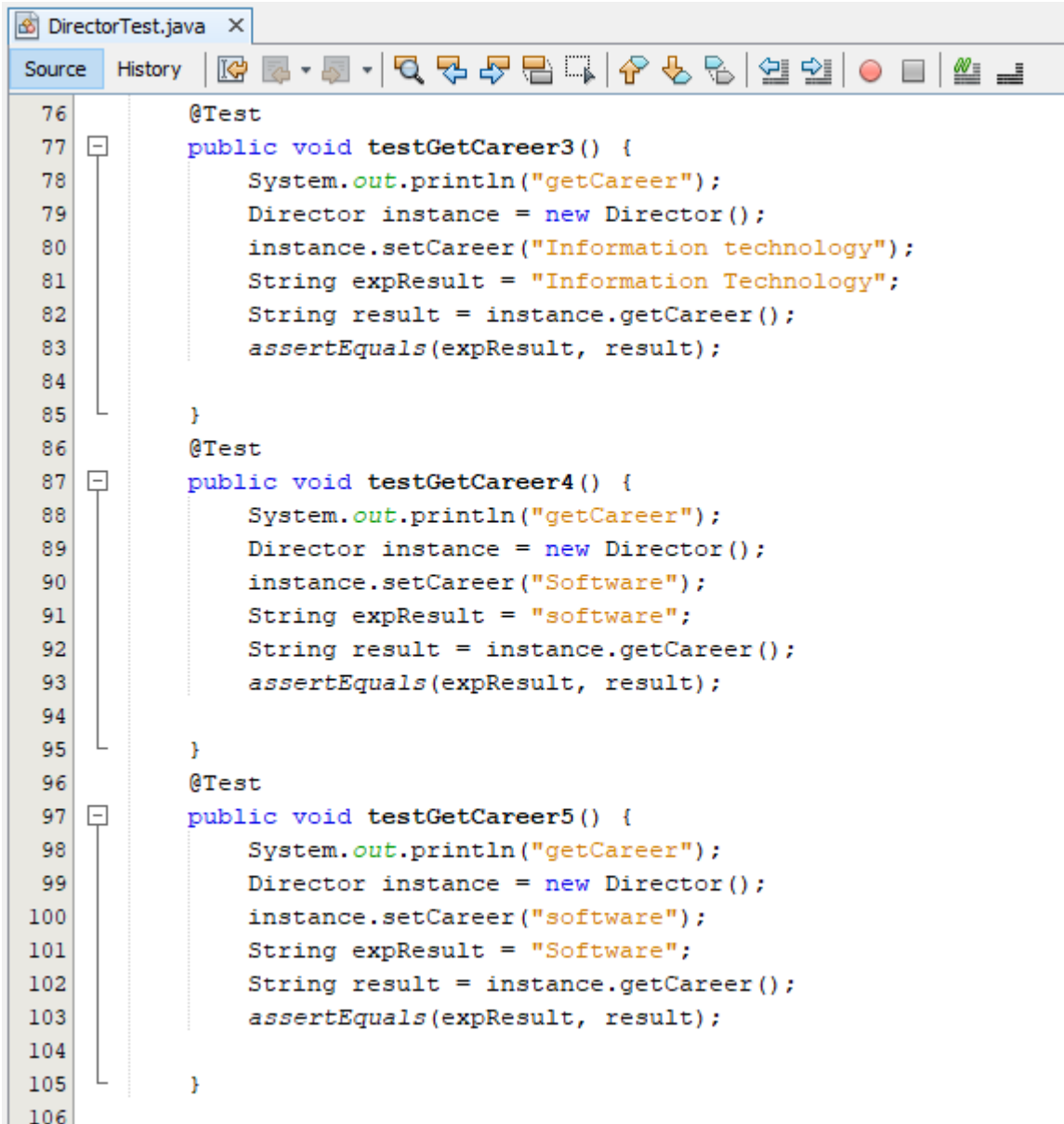
73

assertEquals(expResult, result);

74

75

}



DoctorTest.java X

SourceHistory

46

@Test

47

public void testGetSpecialty() {

48

System.out.println("getSpecialty");

49

Doctor instance = new Doctor();

50

instance.setSpecialty("Medicine");

51

String expResult = "Medicine";

52

String result = instance.getSpecialty();

53

assertEquals(expResult, result);

54

55

}

56

@Test

57

public void testGetSpecialty1() {

58

System.out.println("getSpecialty");

59

Doctor instance = new Doctor();

60

instance.setSpecialty("Medicine");

61

String expResult = "medicine";

62

String result = instance.getSpecialty();

63

assertEquals(expResult, result);

64

65

}

66

@Test

67

public void testGetSpecialty2() {

68

System.out.println("getSpecialty");

69

Doctor instance = new Doctor();

70

instance.setSpecialty("medicine");

71

String expResult = "Medicine";

72

String result = instance.getSpecialty();

73

assertEquals(expResult, result);

74

75

}

76

MedicalCheckTest.java X

SourceHistory

46

@Test

47

public void testGetDate() {

48

System.out.println("getDate");

49

MedicalCheck instance = new MedicalCheck();

50

instance.setDate("01/07/2015");

51

String expResult = "01/07/2015";

52

String result = instance.getDate();

53

assertEquals(expResult, result);

54

55

}

56

@Test

57

public void testGetDate1() {

58

System.out.println("getDate");

59

MedicalCheck instance = new MedicalCheck();

60

instance.setDate("01/07/2015");

61

String expResult = "01/07/2016";

62

String result = instance.getDate();

63

assertEquals(expResult, result);

64

65

}

66

@Test

67

public void testGetDate2() {

68

System.out.println("getDate");

69

MedicalCheck instance = new MedicalCheck();

70

instance.setDate("01/07/2015");

71

String expResult = "01-07-2015";

72

String result = instance.getDate();

73

assertEquals(expResult, result);


74

75

}

MedicalCheckTest.java X

SourceHistory



```
64
65     }
66     @Test
67     public void testGetDate2() {
68         System.out.println("getDate");
69         MedicalCheck instance = new MedicalCheck();
70         instance.setDate("01/07/2015");
71         String expResult = "01-07-2015";
72         String result = instance.getDate();
73         assertEquals(expResult, result);
74     }
75
76     @Test
77     public void testGetDate3() {
78         System.out.println("getDate");
79         MedicalCheck instance = new MedicalCheck();
80         instance.setDate("01/07/2015");
81         String expResult = "01/7/2015";
82         String result = instance.getDate();
83         assertEquals(expResult, result);
84     }
85
86     @Test
87     public void testGetDate4() {
88         System.out.println("getDate");
89         MedicalCheck instance = new MedicalCheck();
90         instance.setDate("01/07/2015");
91         String expResult = "1/07/2015";
92         String result = instance.getDate();
93         assertEquals(expResult, result);
94     }
95 }
```

PersonTest.java X

SourceHistory

46

@Test

47

public void testGetName() {

48

System.out.println("getName");

49

Person instance = new PersonImpl();

50

instance.setName("Johnny Navarro");

51

String expResult = "Johnny Navarro";

52

String result = instance.getName();

53

assertEquals(expResult, result);

54

55

}

56

57

@Test

58

public void testGetName1() {

59

System.out.println("getName");

60

Person instance = new PersonImpl();

61

instance.setName("Johnny Navarro");

62

String expResult = "JohnnyNavarro";

63

String result = instance.getName();

64

assertEquals(expResult, result);

65

66

}

67

68

@Test

69

public void testGetName2() {

70

System.out.println("getName");

71

Person instance = new PersonImpl();

72

instance.setName("Johnny Navarro");

73

String expResult = "Jhony Navarro";

74

String result = instance.getName();

75

assertEquals(expResult, result);


76

77

}

PersonTest.java X


SourceHistory



```
79      @Test
80      public void testGetName3() {
81          System.out.println("getName");
82          Person instance = new PersonImpl();
83          instance.setName("Johnny Navarro");
84          String expResult = "Johnny Nabarro";
85          String result = instance.getName();
86          assertEquals(expResult, result);
87      }
88
89
90      @Test
91      public void testGetName4() {
92          System.out.println("getName");
93          Person instance = new PersonImpl();
94          instance.setName("Carlos Tevez");
95          String expResult = "Carlos Tevez";
96          String result = instance.getName();
97          assertEquals(expResult, result);
98      }
99
100
101      @Test
102      public void testGetName5() {
103          System.out.println("getName");
104          Person instance = new PersonImpl();
105          instance.setName("Carlos Tevez");
106          String expResult = "carlos Tevez";
107          String result = instance.getName();
108          assertEquals(expResult, result);
109      }
110
111
```


PolyclinicTest.java X

SourceHistory



```
46      @Test
47      public void testGetVirtualid() {
48          System.out.println("getVirtualid");
49          Polyclinic instance = new Polyclinic();
50          instance.setVirtualid(384524);
51          int expResult = 384524;
52          int result = instance.getVirtualid();
53          assertEquals(expResult, result);
54      }
55
56      @Test
57      public void testGetVirtualid1() {
58          System.out.println("getVirtualid");
59          Polyclinic instance = new Polyclinic();
60          instance.setVirtualid(38454);
61          int expResult = 384524;
62          int result = instance.getVirtualid();
63          assertEquals(expResult, result);
64      }
65
66      @Test
67      public void testGetVirtualid2() {
68          System.out.println("getVirtualid");
69          Polyclinic instance = new Polyclinic();
70          instance.setVirtualid(3845214);
71          int expResult = 384524;
72          int result = instance.getVirtualid();
73          assertEquals(expResult, result);
74      }
75  }
```

```
PolyclinicTest.java X
Source History
76 @Test
77 public void testGetVirtualid3() {
78     System.out.println("getVirtualid");
79     Polyclinic instance = new Polyclinic();
80     instance.setVirtualid(38524);
81     int expResult = 3844524;
82     int result = instance.getVirtualid();
83     assertEquals(expResult, result);
84 }
85
86 /**
87  * Test of isQrcode method, of class Polyclinic.
88  */
89 @Test
90 public void testIsQrcode() {
91     System.out.println("isQrcode");
92     Polyclinic instance = new Polyclinic();
93     instance.setQrcode(true);
94     boolean expResult = true;
95     boolean result = instance.isQrcode();
96     assertEquals(expResult, result);
97 }
98
99 @Test
100 public void testIsQrcode1() {
101     System.out.println("isQrcode");
102     Polyclinic instance = new Polyclinic();
103     instance.setQrcode(true);
104     boolean expResult = false;
105     boolean result = instance.isQrcode();
106     assertEquals(expResult, result);
107 }
108
109 }
```

StudentTest.java X

SourceHistory

46

@Test

47

public void testGetCareer() {

48

System.out.println("getCareer");

49

Student instance = new Student();

50

instance.setCareer("Software");

51

String expResult = "Software";

52

String result = instance.getCareer();

53

assertEquals(expResult, result);

54

55

}

56

@Test

57

public void testGetCareer1() {

58

System.out.println("getCareer");

59

Student instance = new Student();

60

instance.setCareer("Software");

61

String expResult = "software";

62

String result = instance.getCareer();

63

assertEquals(expResult, result);

64

65

}

66

@Test

67

public void testGetCareer2() {

68

System.out.println("getCareer");

69

Student instance = new Student();

70

instance.setCareer("Software");

71

String expResult = "Software ";

72

String result = instance.getCareer();

73

assertEquals(expResult, result);

74

75

}

VirtualCardTest.java X

SourceHistory

46

@Test

47

public void testGetDepartment() {

48

System.out.println("getDepartment");

49

VirtualCard instance = new VirtualCard();

50

instance.setDepartment("Computer's science");

51

String expResult = "Computer's science";

52

String result = instance.getDepartment();

53

assertEquals(expResult, result);

54

55

}

56

@Test

57

public void testGetDepartment1() {

58

System.out.println("getDepartment");

59

VirtualCard instance = new VirtualCard();

60

instance.setDepartment("Computer's science");

61

String expResult = "Computer's Science";

62

String result = instance.getDepartment();

63

assertEquals(expResult, result);

64

65

}

66

@Test

67

public void testGetDepartment2() {

68

System.out.println("getDepartment");

69

VirtualCard instance = new VirtualCard();

70

instance.setDepartment("Computer's Science");

71

String expResult = "Computer's science";

72

String result = instance.getDepartment();

73

assertEquals(expResult, result);

74

75

}

VirtualCardTest.java X

SourceHistory

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

```
/**
 * Test of getId method, of class VirtualCard.
 */
@Test
public void testGetId() {
    System.out.println("getId");
    VirtualCard instance = new VirtualCard();
    instance.setId(374568);
    int expResult = 374568;
    int result = instance.getId();
    assertEquals(expResult, result);
}

@Test
public void testGetId1() {
    System.out.println("getId");
    VirtualCard instance = new VirtualCard();
    instance.setId(3745678);
    int expResult = 374568;
    int result = instance.getId();
    assertEquals(expResult, result);
}

@Test
public void testGetId2() {
    System.out.println("getId");
    VirtualCard instance = new VirtualCard();
    instance.setId(374568);
    int expResult = 3747568;
    int result = instance.getId();
    assertEquals(expResult, result);
}
```