

```
StudentTest.java x LoginTest.java x ConnectionDataBaseTest.java x
Source History
42 */
43 @Test
44 public void testConectionDataBase() {
45     System.out.println("ConectionDataBase");
46     String collection = "bry";
47     ConnectionDataBase instance = new ConnectionDataBase();
48     instance.ConnectionDataBase(collection);
49 }
50
51
52 @Test
53 public void testConectionDataBase1() {
54     System.out.println("ConectionDataBase");
55     String collection = "Majof";
56     ConnectionDataBase instance = new ConnectionDataBase();
57     instance.ConnectionDataBase(collection);
58 }
59
60 @Test
61 public void testConectionDataBase2() {
62     System.out.println("ConectionDataBase");
63     String collection = "Mjf";
64     ConnectionDataBase instance = new ConnectionDataBase();
65     instance.ConnectionDataBase(collection);
66 }
67
68 @Test
69 public void testConectionDataBase3() {
70     System.out.println("ConectionDataBase");
71     String collection = "FJOC1";
72     ConnectionDataBase instance = new ConnectionDataBase();
73     instance.ConnectionDataBase(collection);
74 }
75
```

```
StudentTest.java X LoginTest.java X ConectionDataBaseTest.java X
Source History
86 /**
87  * Test of ConectionDataBaseDiagnosis method, of class ConectionDataBase.
88  */
89 @Test
90 public void testConectionDataBaseDiagnosis() {
91     System.out.println("ConectionDataBaseDiagnosis");
92     String collection = "";
93     ConectionDataBase instance = new ConectionDataBase();
94     instance.ConectionDataBaseDiagnosis(collection);
95 }
96 /**
97  * Test of ConectionDataBaseAppointment method, of class ConectionDataBase.
98  */
99 @Test
100 public void testConectionDataBaseAppointment() {
101     System.out.println("ConectionDataBaseAppointment");
102     String collection = "Mishu";
103     ConectionDataBase instance = new ConectionDataBase();
104     instance.ConectionDataBaseAppointment(collection);
105 }
106 /**
107  * Test of create method, of class ConectionDataBase.
108  */
109 @Test
110 public void testCreate_7args() {
111     System.out.println("create");
112     String name = "Bryan Pazmino";
113     String id = "L00354835";
114     String career = "TI";
115     String email = "bapazmino@espe.edu.ec";
116     String address = "Conocoto";
117     int age = 28;
118     String gender = "male";
119     ConectionDataBase instance = new ConectionDataBase();
```

```
StudentTest.java x LoginTest.java x ConectionDataBaseTest.java x
Source History
119 ConectionDataBase instance = new ConectionDataBase();
120 boolean expResult = false;
121 boolean result = instance.create(name, id, career, email, address, age, gender);
122 assertEquals(expResult, result);
123 // TODO review the generated test code and remove the default call to fail.
124 fail("The test case is a prototype.");
125 }
126 /**
127  * Test of create method, of class ConectionDataBase.
128  */
129 @Test
130 public void testCreate_3args() {
131     System.out.println("create");
132     String nameDoctor = "";
133     String workingHour = "";
134     String specialty = "";
135     ConectionDataBase instance = new ConectionDataBase();
136     boolean expResult = false;
137     boolean result = instance.create(nameDoctor, workingHour, specialty);
138     assertEquals(expResult, result);
139     // TODO review the generated test code and remove the default call to fail.
140     fail("The test case is a prototype.");
141 }
142 @Test
143 public void testCreate_3args1() {
144     System.out.println("create");
145     String nameDoctor = "";
146     String workingHour = "";
147     String specialty = "";
148     ConectionDataBase instance = new ConectionDataBase();
149     boolean expResult = false;
150     boolean result = instance.create(nameDoctor, workingHour, specialty);
151     assertEquals(expResult, result);
152     // TODO review the generated test code and remove the default call to fail.
```

```
StudentTest.java x LoginTest.java x ConectionDataBaseTest.java x
Source History
152 // TODO review the generated test code and remove the default call to fail.
153 fail("The test case is a prototype.");
154 }
155 @Test
156 public void testCreate_3args2() {
157     System.out.println("create");
158     String nameDoctor = "256.2";
159     String workingHour = "9,6";
160     String specialty = "Frak";
161     ConectionDataBase instance = new ConectionDataBase();
162     boolean expResult = false;
163     boolean result = instance.create(nameDoctor, workingHour, specialty);
164     assertEquals(expResult, result);
165     // TODO review the generated test code and remove the default call to fail.
166     fail("The test case is a prototype.");
167 }
168 /**
169  * Test of create method, of class ConectionDataBase.
170  */
171 @Test
172 public void testCreate_String_String() {
173     System.out.println("create");
174     String data = "29/09/2021";
175     String time = "09:30";
176     ConectionDataBase instance = new ConectionDataBase();
177     boolean expResult = false;
178     boolean result = instance.create(data, time);
179     assertEquals(expResult, result);
180     // TODO review the generated test code and remove the default call to fail.
181     fail("The test case is a prototype.");
182 }
183
184 /**
185  * Test of delete method, of class ConectionDataBase.
```

StudentTest.java X LoginTest.java X ConectionDataBaseTest.java X

Source History

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

```
/**
 * Test of delete method, of class ConectionDataBase.
 */
@Test
public void testDelete() {
    System.out.println("delete");
    String name = "";
    ConectionDataBase instance = new ConectionDataBase();
    boolean expResult = false;
    boolean result = instance.delete(name);
    assertEquals(expResult, result);
}

@Test
public void testDelete7() {
    System.out.println("delete");
    String name = "Frank";
    ConectionDataBase instance = new ConectionDataBase();
    boolean expResult = false;
    boolean result = instance.delete(name);
    assertEquals(expResult, result);
}

@Test
public void testDelete1() {
    System.out.println("delete");
    String name = "1593";
    ConectionDataBase instance = new ConectionDataBase();
    boolean expResult = false;
    boolean result = instance.delete(name);
    assertEquals(expResult, result);
}
```

StudentTest.java x LoginTest.java x ConectionDataBaseTest.java x

Source History

216

}

217

@Test

218

public void testDelet2() {

219

System.out.println("delete");

220

String name = "Majof56";

221

ConectionDataBase instance = new ConectionDataBase();

222

boolean expResult = false;

223

boolean result = instance.delete(name);

224

assertEquals(expResult, result);

225

226

}

227

@Test

228

public void testDelete3() {

229

System.out.println("delete");

230

String name = "Brake6";

231

ConectionDataBase instance = new ConectionDataBase();

232

boolean expResult = false;

233

boolean result = instance.delete(name);

234

assertEquals(expResult, result);

235

236

}

237

@Test

238

public void testDelete4() {

239

System.out.println("delete");

240

String name = "Jhanny";

241

ConectionDataBase instance = new ConectionDataBase();

242

boolean expResult = false;

243

boolean result = instance.delete(name);

244

assertEquals(expResult, result);

245

246

}

247

@Test

248

public void testDelete5() {

249

System.out.println("delete");

```
237     @Test
238     public void testDelete4() {
239         System.out.println("delete");
240         String name = "Jhanny";
241         ConectionDataBase instance = new ConectionDataBase();
242         boolean expResult = false;
243         boolean result = instance.delete(name);
244         assertEquals(expResult, result);
245     }
246
247     @Test
248     public void testDelete5() {
249         System.out.println("delete");
250         String name = "juan89";
251         ConectionDataBase instance = new ConectionDataBase();
252         boolean expResult = false;
253         boolean result = instance.delete(name);
254         assertEquals(expResult, result);
255     }
256
257
258     /**
259     * Test of read method, of class ConectionDataBase.
260     */
261     @Test
262     public void testRead() {
263         System.out.println("read");
264         ConectionDataBase instance = new ConectionDataBase();
265         instance.read();
266     }
267
268 }
```

StudentTest.java | ConexionDataBaseTest.java

Source | History

```
34 }
35
36 @After
37 public void tearDown() {
38 }
39
40 /**
41  * Test of ConexionDataBase method, of class ConexionDataBase.
42  */
43 @Test
44 public void testConexionDataBase() {
45     System.out.println("ConexionDataBase");
46     String collection = "bry";
```

Test Results

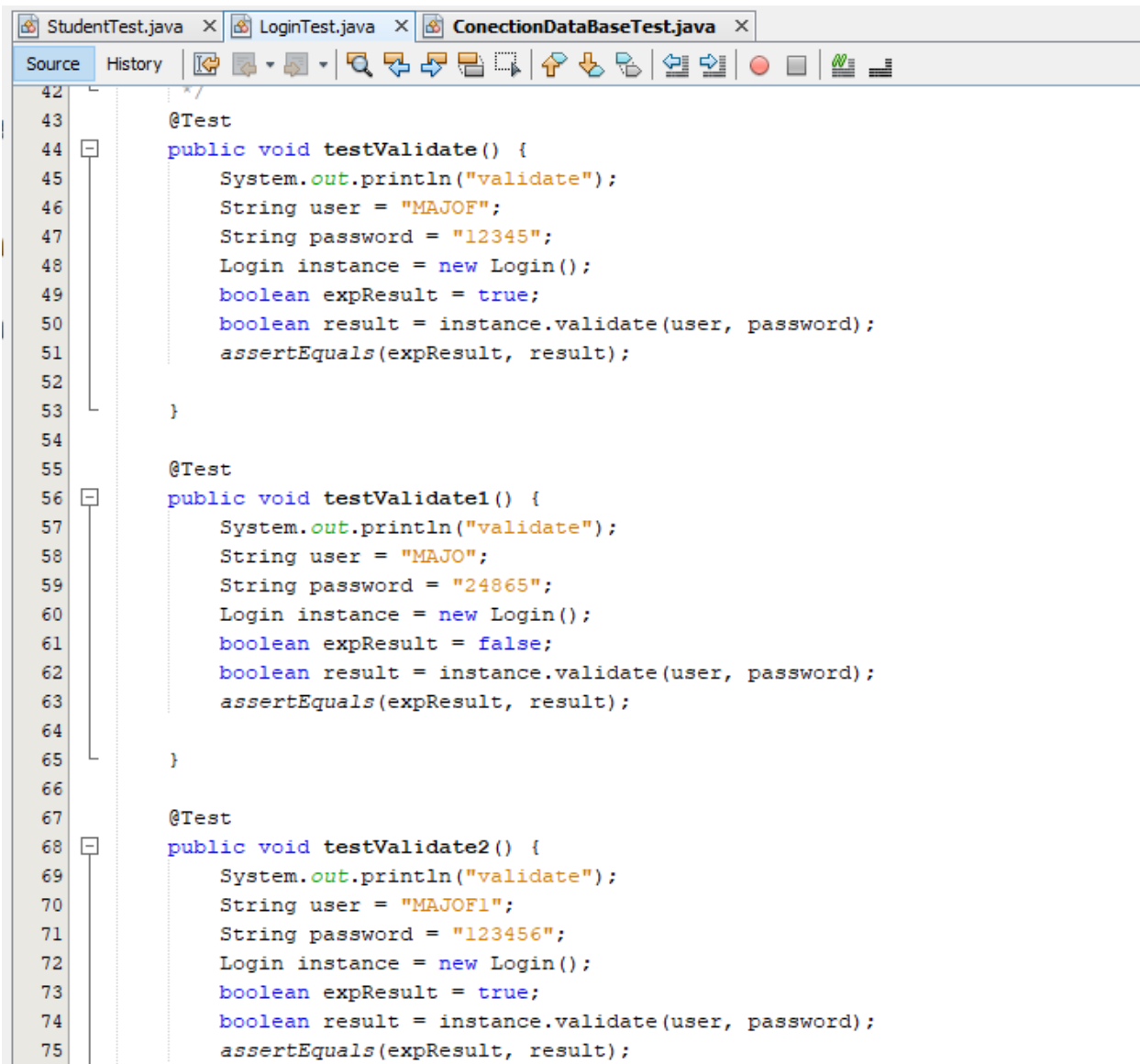
ec.edu.espe.prototypevirtualid.controller.ConexionDataBaseTest

Tests passed: 30,00 %

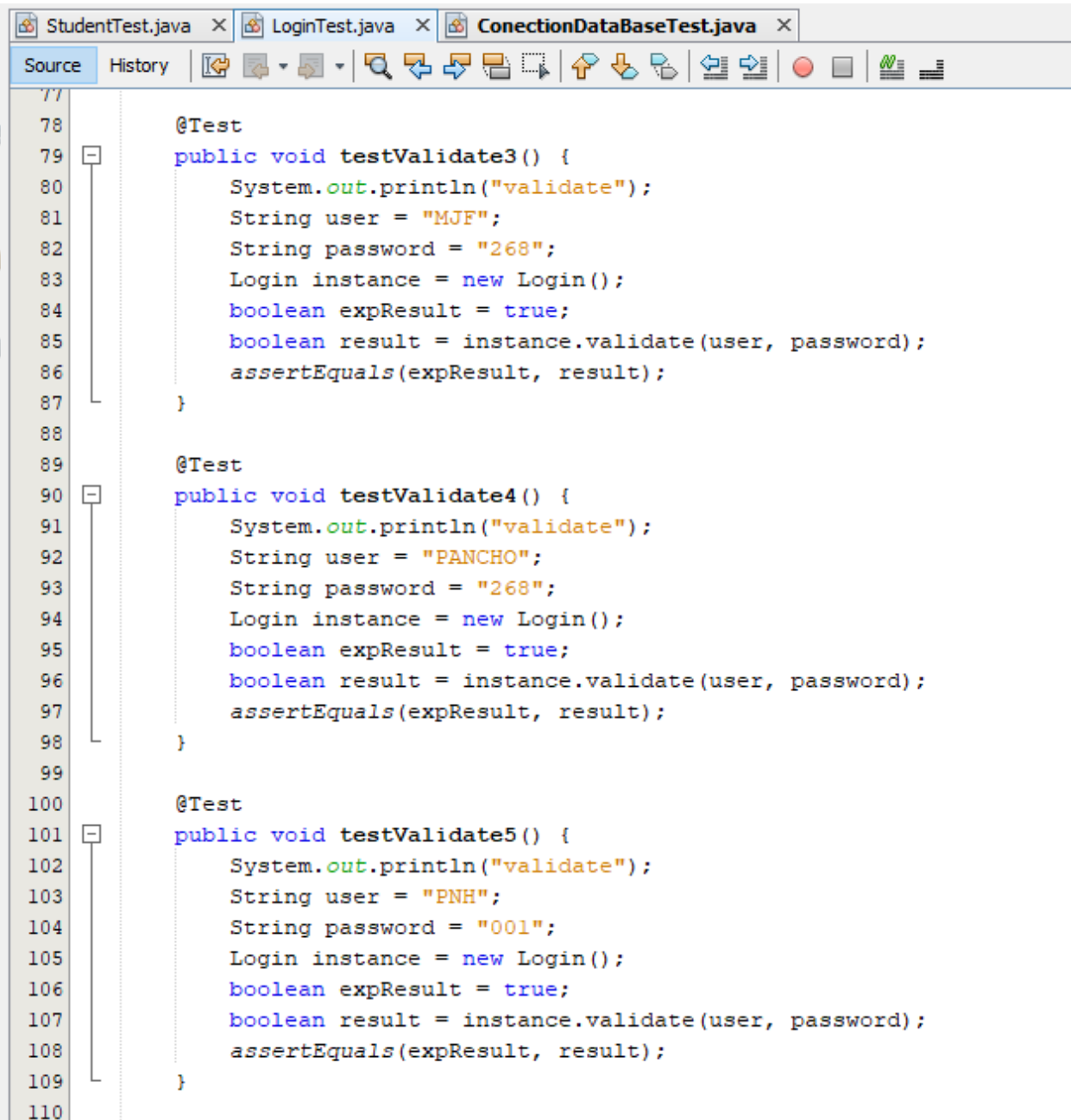
6 tests passed, 14 tests caused an error. (1,555 s)

- ec.edu.espe.prototypevirtualid.controller.ConexionDataBaseTest Failed
- testCreate_3args1 caused an ERROR: java.lang.NullPointerException
- testCreate_3args2 caused an ERROR: java.lang.NullPointerException
- testCreate_3args caused an ERROR: java.lang.NullPointerException
- testCreate_7args caused an ERROR: java.lang.NullPointerException
- testRead caused an ERROR: java.lang.NullPointerException
- testConexionDataBaseDiagnosis caused an ERROR: state should be: collectionName is not empty
- testDelete1 caused an ERROR: java.lang.NullPointerException
- testDelete3 caused an ERROR: java.lang.NullPointerException
- testDelete4 caused an ERROR: java.lang.NullPointerException
- testDelete5 caused an ERROR: java.lang.NullPointerException
- testDelete7 caused an ERROR: java.lang.NullPointerException
- testDelet2 caused an ERROR: java.lang.NullPointerException
- testDelete caused an ERROR: java.lang.NullPointerException
- testCreate_String_String caused an ERROR: java.lang.NullPointerException

ConexionDatabase
ConexionDataBaseDiagnosis
ConexionDataBaseAppointment
delete
delete
delete
delete
delete
delete
ConexionDataBaseDoctor
create
ConexionDataBase
ConexionDataBase
ConexionDataBase
mar 03, 2021 4:56:22 AM com.mongodb.diagnostics.logging.JULLogger log
INFORMACIÓN: Cluster created with settings {hosts=[127.0.0.1:27017], srvHost=cluste
mar 03, 2021 4:56:22 AM com.mongodb.diagnostics.logging.JULLogger log
INFORMACIÓN: Cluster created with settings {hosts=[127.0.0.1:27017], srvHost=cluste



```
42
43
44 @Test
45 public void testValidate() {
46     System.out.println("validate");
47     String user = "MAJOF";
48     String password = "12345";
49     Login instance = new Login();
50     boolean expResult = true;
51     boolean result = instance.validate(user, password);
52     assertEquals(expResult, result);
53 }
54
55 @Test
56 public void testValidate1() {
57     System.out.println("validate");
58     String user = "MAJO";
59     String password = "24865";
60     Login instance = new Login();
61     boolean expResult = false;
62     boolean result = instance.validate(user, password);
63     assertEquals(expResult, result);
64 }
65
66 @Test
67 public void testValidate2() {
68     System.out.println("validate");
69     String user = "MAJOF1";
70     String password = "123456";
71     Login instance = new Login();
72     boolean expResult = true;
73     boolean result = instance.validate(user, password);
74     assertEquals(expResult, result);
75 }
```



```
77
78     @Test
79     public void testValidate3() {
80         System.out.println("validate");
81         String user = "MJF";
82         String password = "268";
83         Login instance = new Login();
84         boolean expResult = true;
85         boolean result = instance.validate(user, password);
86         assertEquals(expResult, result);
87     }
88
89     @Test
90     public void testValidate4() {
91         System.out.println("validate");
92         String user = "PANCHO";
93         String password = "268";
94         Login instance = new Login();
95         boolean expResult = true;
96         boolean result = instance.validate(user, password);
97         assertEquals(expResult, result);
98     }
99
100    @Test
101    public void testValidate5() {
102        System.out.println("validate");
103        String user = "PNH";
104        String password = "001";
105        Login instance = new Login();
106        boolean expResult = true;
107        boolean result = instance.validate(user, password);
108        assertEquals(expResult, result);
109    }
110
```

```
StudentTest.java x LoginTest.java x ConectionDataBaseTest.java x
Source History
110
111 @Test
112 public void testValidate6() {
113     System.out.println("validate");
114     String user = "PACH";
115     String password = "8203";
116     Login instance = new Login();
117     boolean expResult = true;
118     boolean result = instance.validate(user, password);
119     assertEquals(expResult, result);
120 }
121
122 @Test
123 public void testValidate7() {
124     System.out.println("validate");
125     String user = "PO";
126     String password = "9242";
127     Login instance = new Login();
128     boolean expResult = true;
129     boolean result = instance.validate(user, password);
130     assertEquals(expResult, result);
131 }
132
133 @Test
134 public void testValidate8() {
135     System.out.println("validate");
136     String user = "Majol";
137     String password = "0001";
138     Login instance = new Login();
139     boolean expResult = true;
140     boolean result = instance.validate(user, password);
141     assertEquals(expResult, result);
142 }
143
```

```
StudentTest.java x LoginTest.java x ConectionDataBaseTest.java x
Source History
143
144 @Test
145 public void testValidate9() {
146     System.out.println("validate");
147     String user = "Family";
148     String password = "333";
149     Login instance = new Login();
150     boolean expResult = true;
151     boolean result = instance.validate(user, password);
152     assertEquals(expResult, result);
153 }
154
155 @Test
156 public void testValidate10() {
157     System.out.println("validate");
158     String user = "Fam";
159     String password = "985";
160     Login instance = new Login();
161     boolean expResult = true;
162     boolean result = instance.validate(user, password);
163     assertEquals(expResult, result);
164 }
165
166 @Test
167 public void testValidate11() {
168     System.out.println("validate");
169     String user = "Fernand";
170     String password = "645";
171     Login instance = new Login();
172     boolean expResult = true;
173     boolean result = instance.validate(user, password);
174     assertEquals(expResult, result);
175 }
176
```

StudentTest.java x LoginTest.java x ConectionDataBaseTest.java x

Source History

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

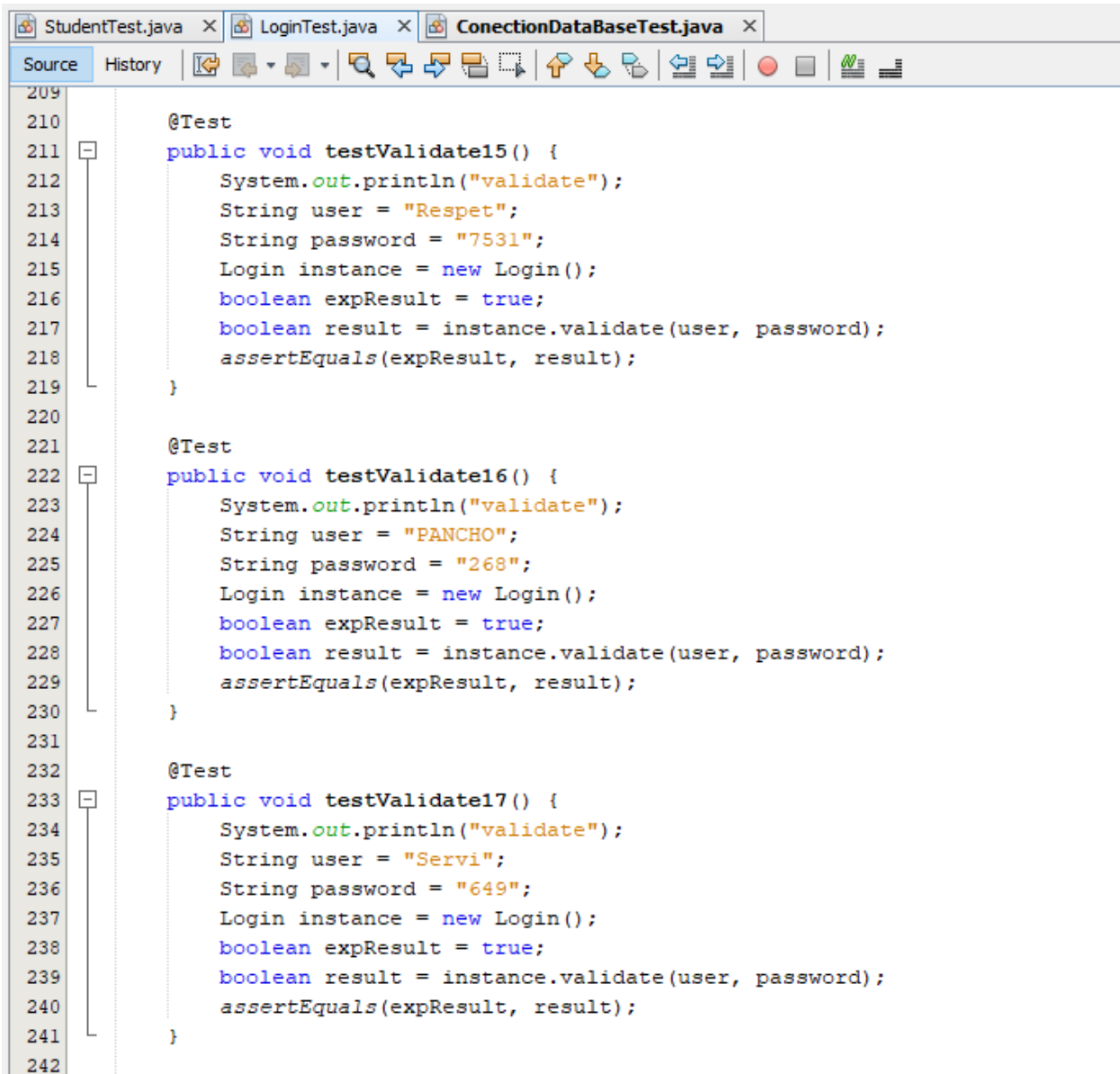
208

209

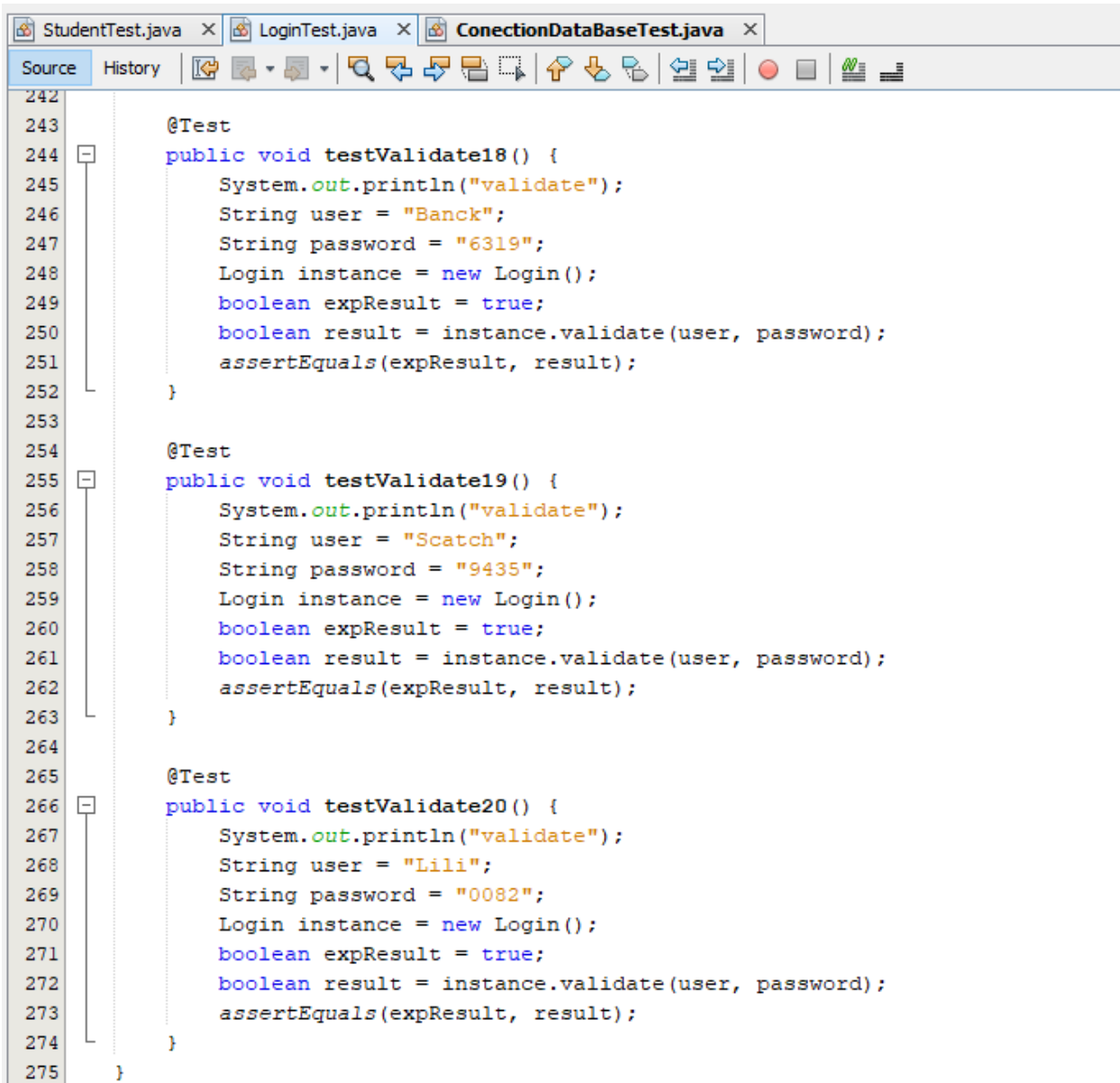
```
@Test
public void testValidate12() {
    System.out.println("validate");
    String user = "Bry";
    String password = "632";
    Login instance = new Login();
    boolean expResult = true;
    boolean result = instance.validate(user, password);
    assertEquals(expResult, result);
}

@Test
public void testValidate13() {
    System.out.println("validate");
    String user = "Jhon";
    String password = "092";
    Login instance = new Login();
    boolean expResult = true;
    boolean result = instance.validate(user, password);
    assertEquals(expResult, result);
}

@Test
public void testValidate14() {
    System.out.println("validate");
    String user = "Valorant";
    String password = "2582";
    Login instance = new Login();
    boolean expResult = true;
    boolean result = instance.validate(user, password);
    assertEquals(expResult, result);
}
```



```
209
210
211 @Test
212 public void testValidate15() {
213     System.out.println("validate");
214     String user = "Respet";
215     String password = "7531";
216     Login instance = new Login();
217     boolean expResult = true;
218     boolean result = instance.validate(user, password);
219     assertEquals(expResult, result);
220 }
221
222 @Test
223 public void testValidate16() {
224     System.out.println("validate");
225     String user = "PANCHO";
226     String password = "268";
227     Login instance = new Login();
228     boolean expResult = true;
229     boolean result = instance.validate(user, password);
230     assertEquals(expResult, result);
231 }
232
233 @Test
234 public void testValidate17() {
235     System.out.println("validate");
236     String user = "Servi";
237     String password = "649";
238     Login instance = new Login();
239     boolean expResult = true;
240     boolean result = instance.validate(user, password);
241     assertEquals(expResult, result);
242 }
```



```
242
243     @Test
244     public void testValidate18() {
245         System.out.println("validate");
246         String user = "Banck";
247         String password = "6319";
248         Login instance = new Login();
249         boolean expectedResult = true;
250         boolean result = instance.validate(user, password);
251         assertEquals(expectedResult, result);
252     }
253
254     @Test
255     public void testValidate19() {
256         System.out.println("validate");
257         String user = "Scatch";
258         String password = "9435";
259         Login instance = new Login();
260         boolean expectedResult = true;
261         boolean result = instance.validate(user, password);
262         assertEquals(expectedResult, result);
263     }
264
265     @Test
266     public void testValidate20() {
267         System.out.println("validate");
268         String user = "Lili";
269         String password = "0082";
270         Login instance = new Login();
271         boolean expectedResult = true;
272         boolean result = instance.validate(user, password);
273         assertEquals(expectedResult, result);
274     }
275 }
```

