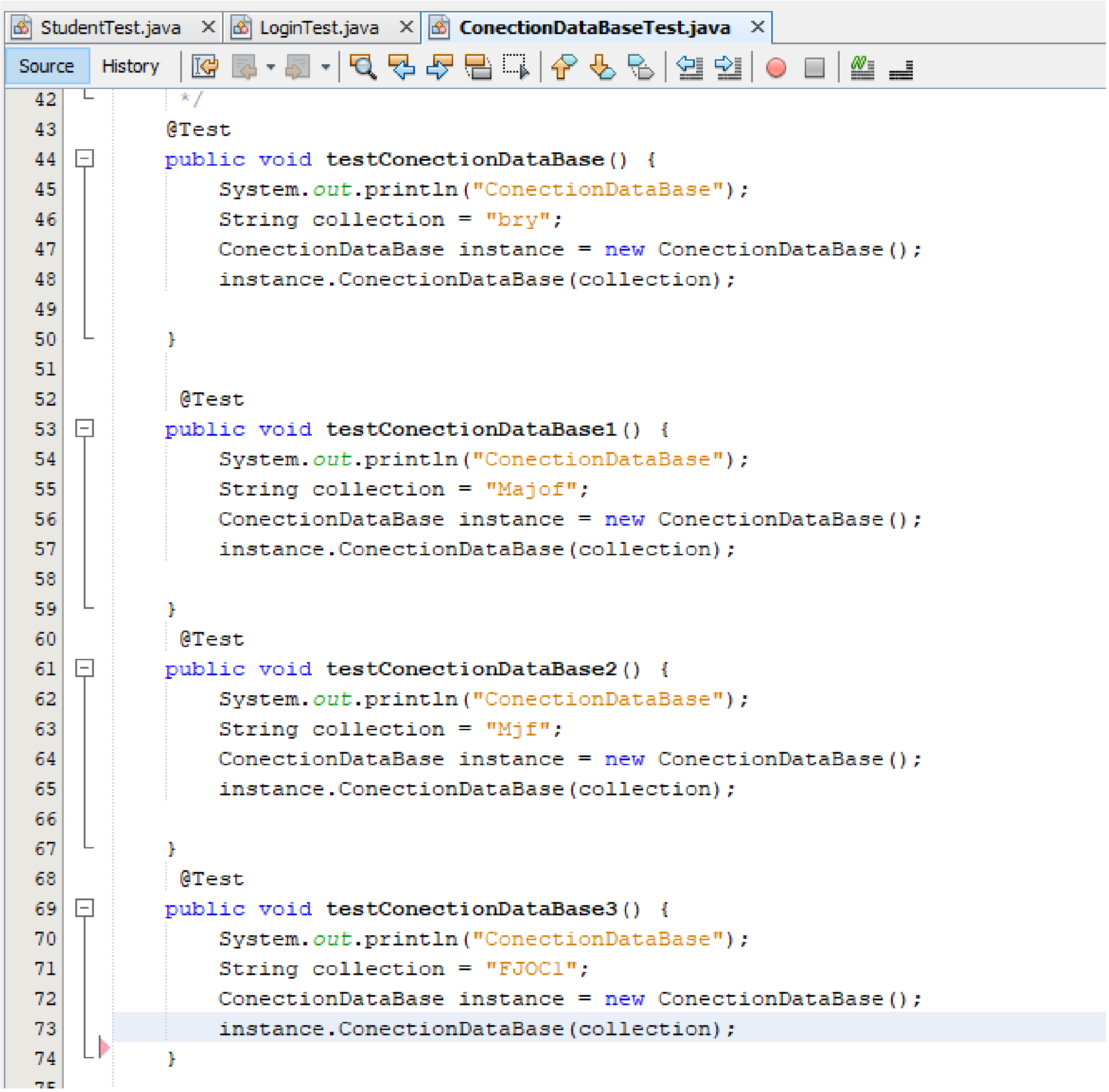
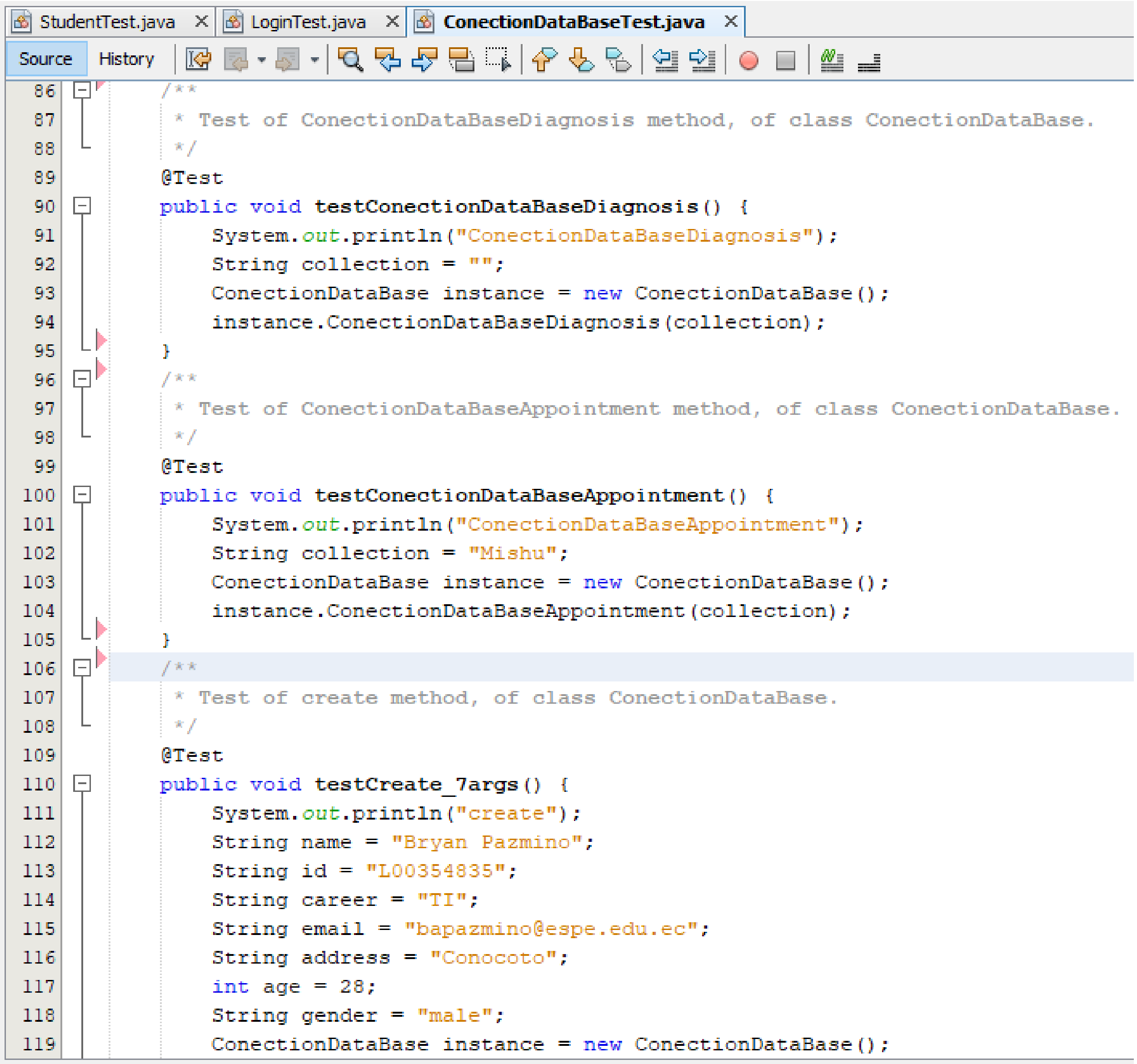
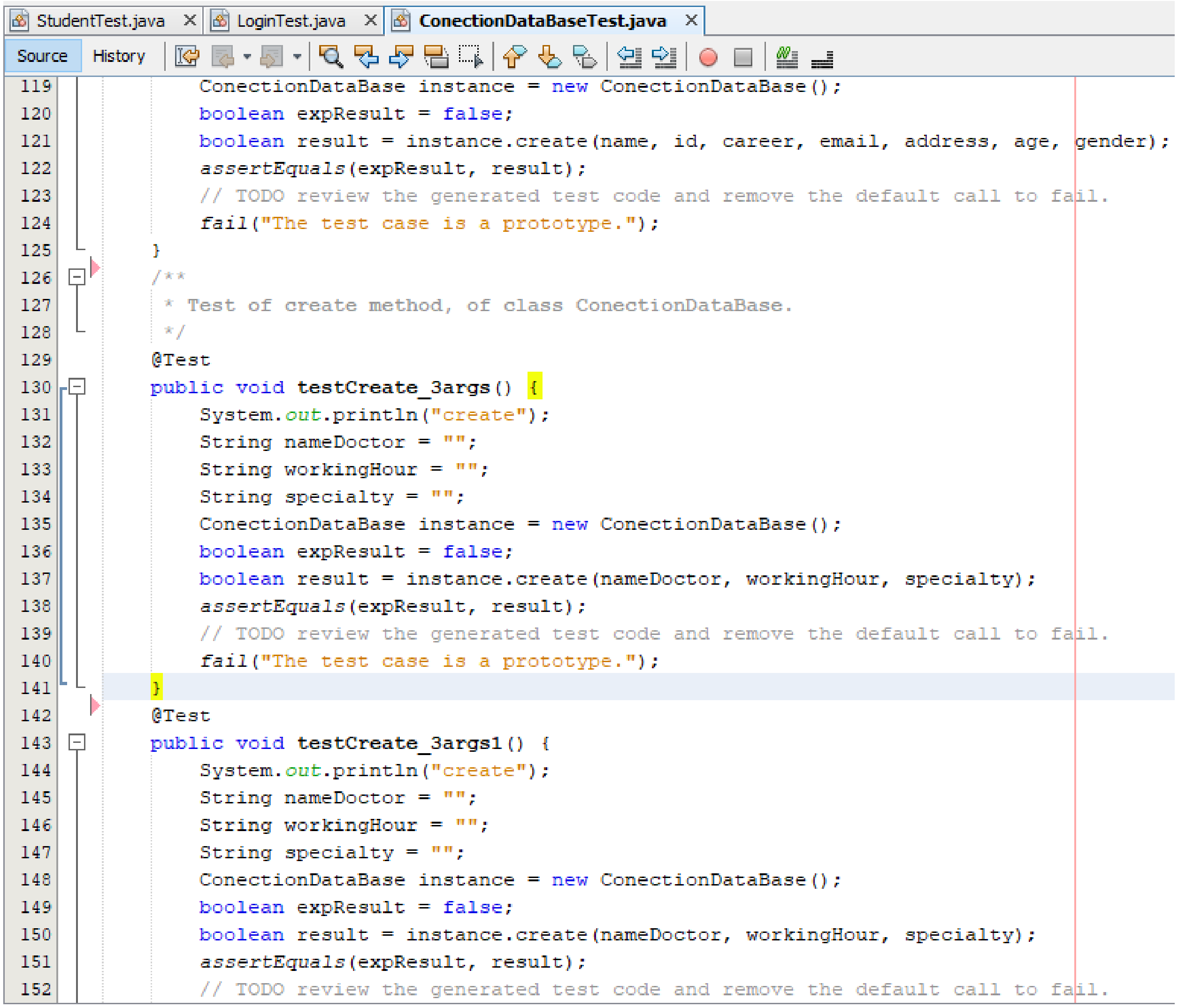


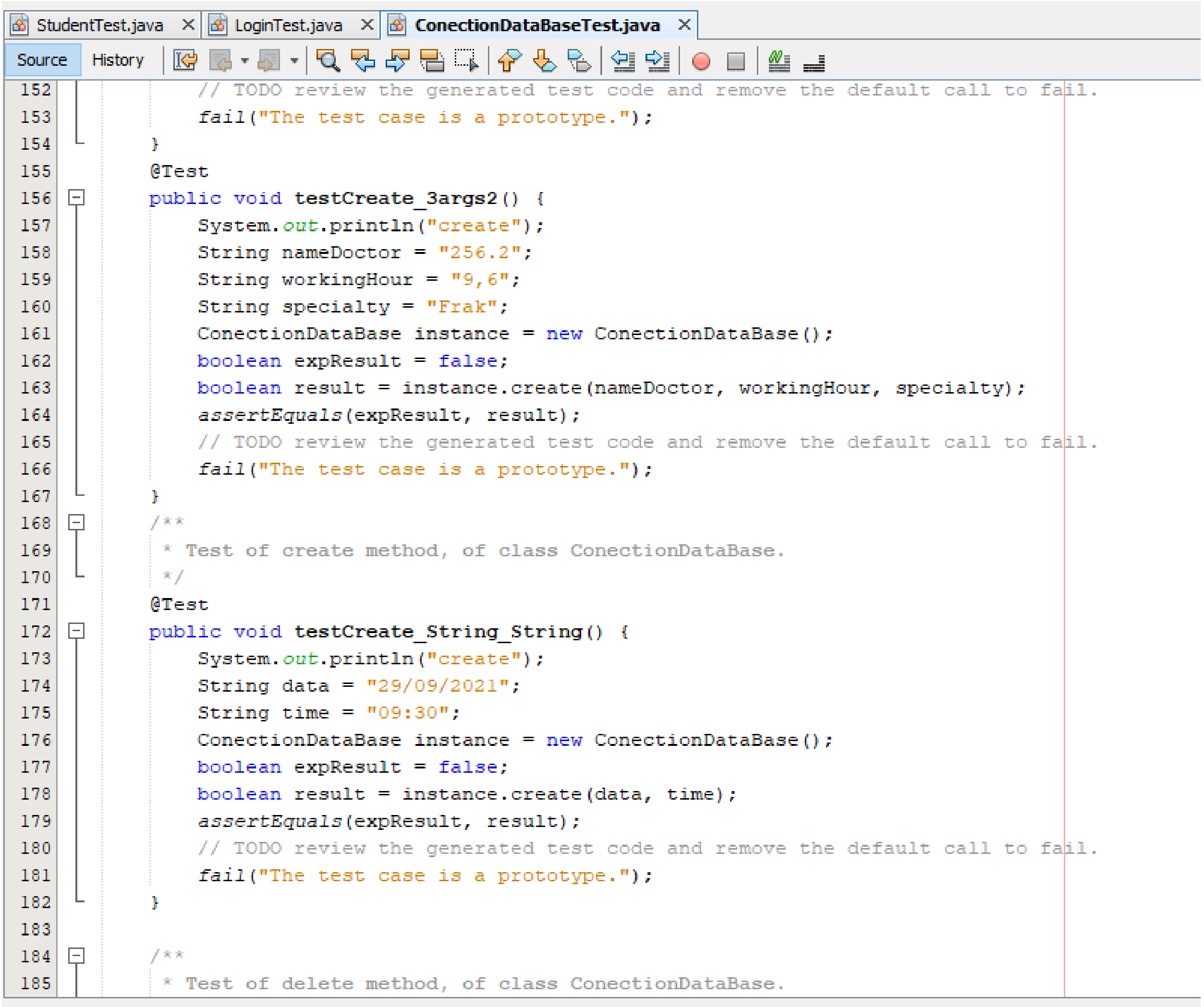
```java
     */
    @Test
    public void testConectionDataBase() {
        System.out.println("ConectionDataBase");
        String collection = "bry";
        ConectionDataBase instance = new ConectionDataBase();
        instance.ConectionDataBase(collection);

    }

     @Test
    public void testConectionDataBase1() {
        System.out.println("ConectionDataBase");
        String collection = "Majof";
        ConectionDataBase instance = new ConectionDataBase();
        instance.ConectionDataBase(collection);

    }
     @Test
    public void testConectionDataBase2() {
        System.out.println("ConectionDataBase");
        String collection = "Mjf";
        ConectionDataBase instance = new ConectionDataBase();
        instance.ConectionDataBase(collection);

    }
     @Test
    public void testConectionDataBase3() {
        System.out.println("ConectionDataBase");
        String collection = "FJOC1";
        ConectionDataBase instance = new ConectionDataBase();
        instance.ConectionDataBase(collection);
    }
```



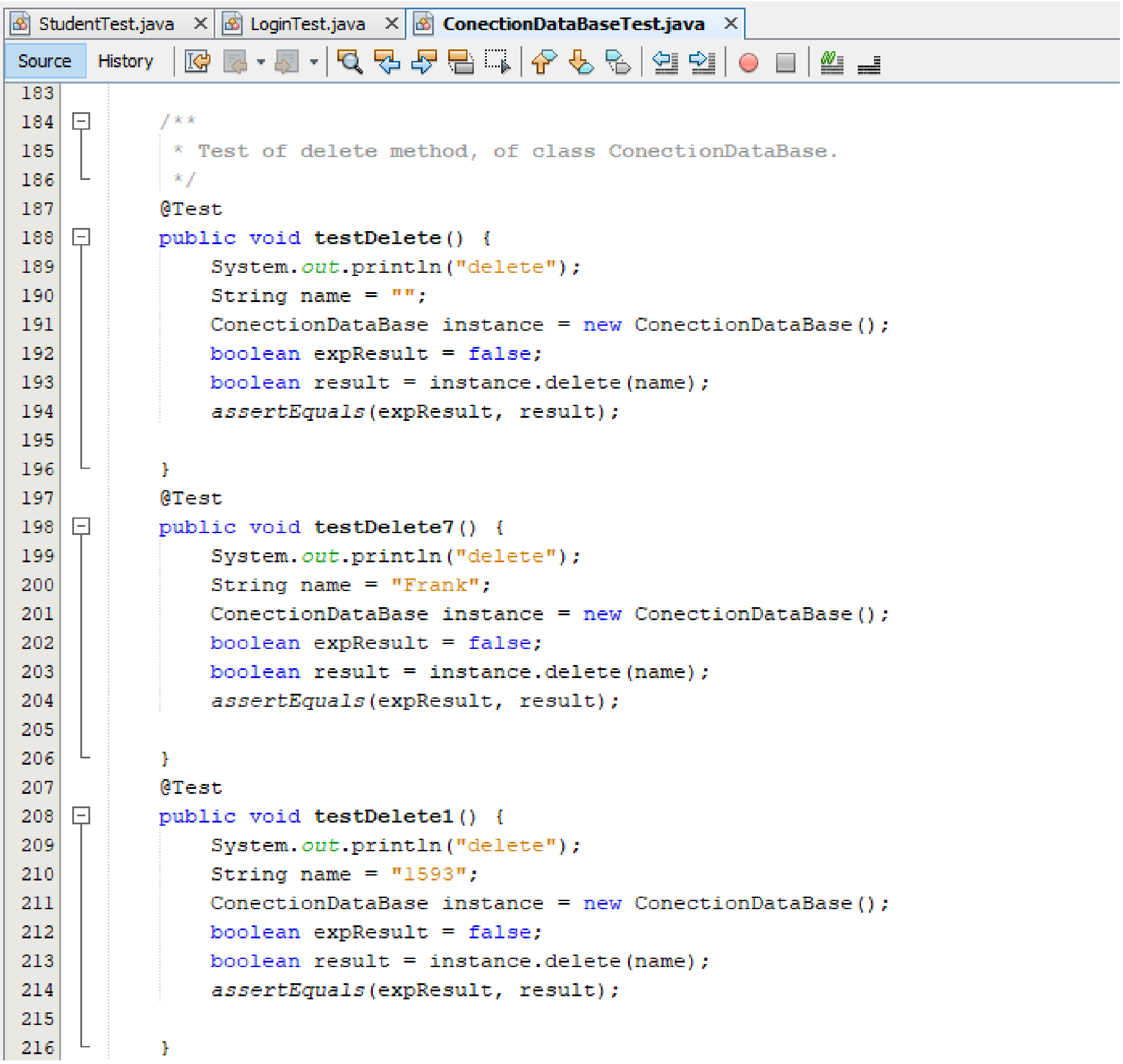
```java
    /**
     * Test of ConectionDataBaseDiagnosis method, of class ConectionDataBase.
     */
    @Test
    public void testConectionDataBaseDiagnosis() {
        System.out.println("ConectionDataBaseDiagnosis");
        String collection = "";
        ConectionDataBase instance = new ConectionDataBase();
        instance.ConectionDataBaseDiagnosis(collection);
    }
    /**
     * Test of ConectionDataBaseAppointment method, of class ConectionDataBase.
     */
    @Test
    public void testConectionDataBaseAppointment() {
        System.out.println("ConectionDataBaseAppointment");
        String collection = "Mishu";
        ConectionDataBase instance = new ConectionDataBase();
        instance.ConectionDataBaseAppointment(collection);
    }
    /**
     * Test of create method, of class ConectionDataBase.
     */
    @Test
    public void testCreate_7args() {
        System.out.println("create");
        String name = "Bryan Pazmino";
        String id = "L00354835";
        String career = "TI";
        String email = "bapazmino@espe.edu.ec";
        String address = "Conocoto";
        int age = 28;
        String gender = "male";
        ConectionDataBase instance = new ConectionDataBase();
```



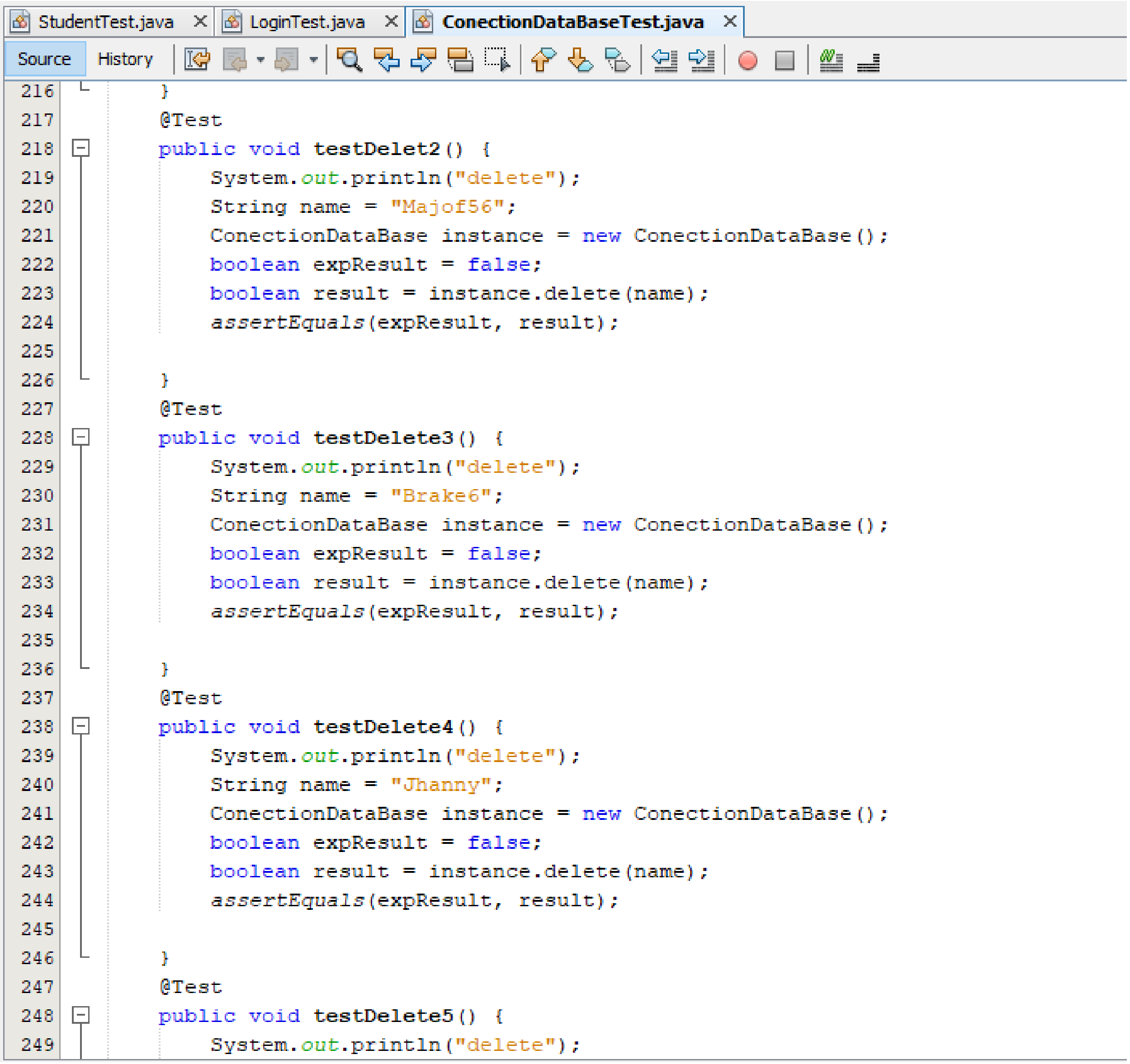
```java
        ConectionDataBase instance = new ConectionDataBase();
        boolean expResult = false;
        boolean result = instance.create(name, id, career, email, address, age, gender);
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default call to fail.
        fail("The test case is a prototype.");
    }
    /**
     * Test of create method, of class ConectionDataBase.
     */
    @Test
    public void testCreate_3args() {
        System.out.println("create");
        String nameDoctor = "";
        String workingHour = "";
        String specialty = "";
        ConectionDataBase instance = new ConectionDataBase();
        boolean expResult = false;
        boolean result = instance.create(nameDoctor, workingHour, specialty);
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default call to fail.
        fail("The test case is a prototype.");
    }
    @Test
    public void testCreate_3args1() {
        System.out.println("create");
        String nameDoctor = "";
        String workingHour = "";
        String specialty = "";
        ConectionDataBase instance = new ConectionDataBase();
        boolean expResult = false;
        boolean result = instance.create(nameDoctor, workingHour, specialty);
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default call to fail.
```



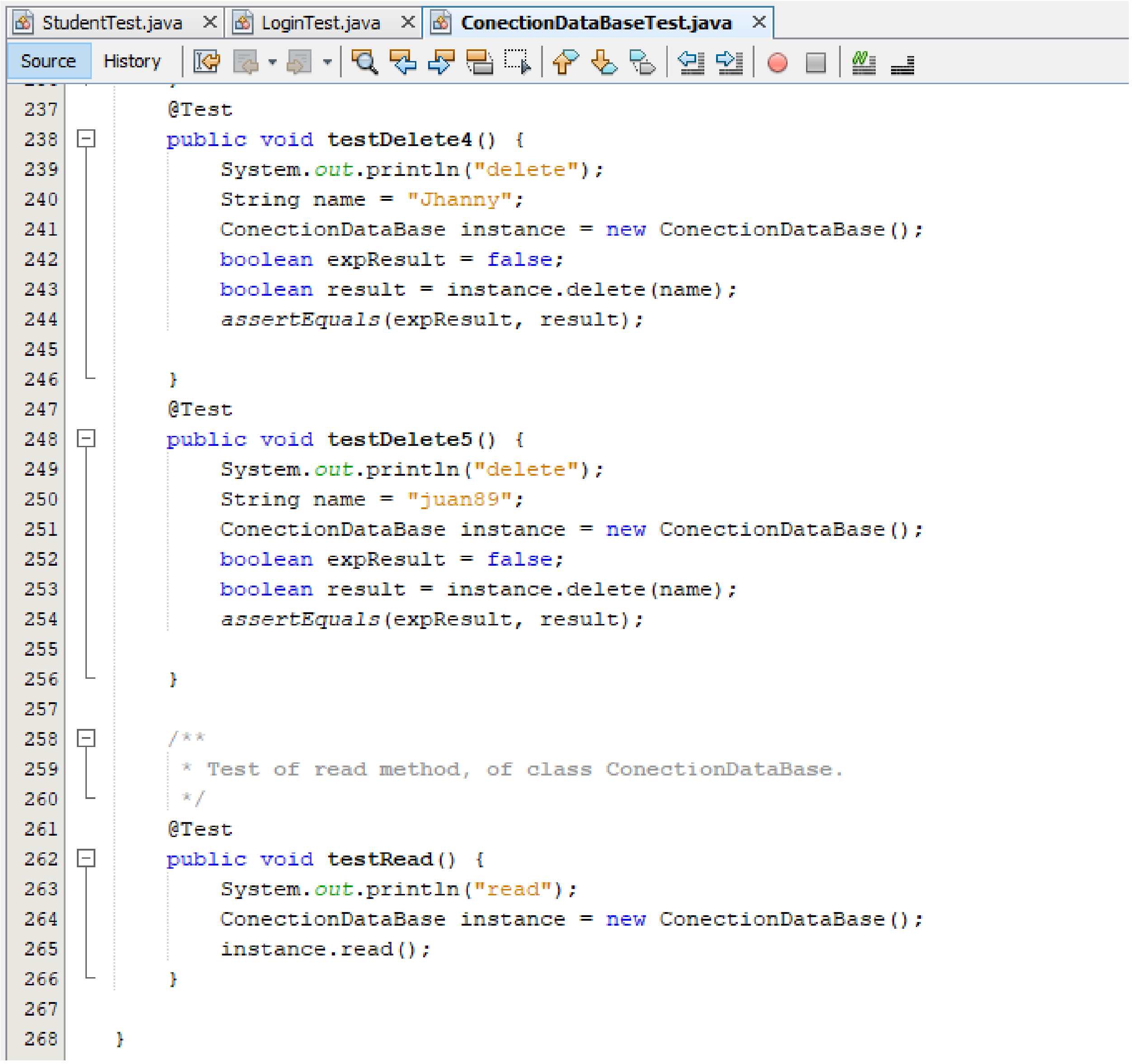
```java
        // TODO review the generated test code and remove the default call to fail.
        fail("The test case is a prototype.");
    }
    @Test
    public void testCreate_3args2() {
        System.out.println("create");
        String nameDoctor = "256.2";
        String workingHour = "9,6";
        String specialty = "Frak";
        ConectionDataBase instance = new ConectionDataBase();
        boolean expResult = false;
        boolean result = instance.create(nameDoctor, workingHour, specialty);
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default call to fail.
        fail("The test case is a prototype.");
    }
    /**
     * Test of create method, of class ConectionDataBase.
     */
    @Test
    public void testCreate_String_String() {
        System.out.println("create");
        String data = "29/09/2021";
        String time = "09:30";
        ConectionDataBase instance = new ConectionDataBase();
        boolean expResult = false;
        boolean result = instance.create(data, time);
        assertEquals(expResult, result);
        // TODO review the generated test code and remove the default call to fail.
        fail("The test case is a prototype.");
    }

    /**
     * Test of delete method, of class ConectionDataBase.
```



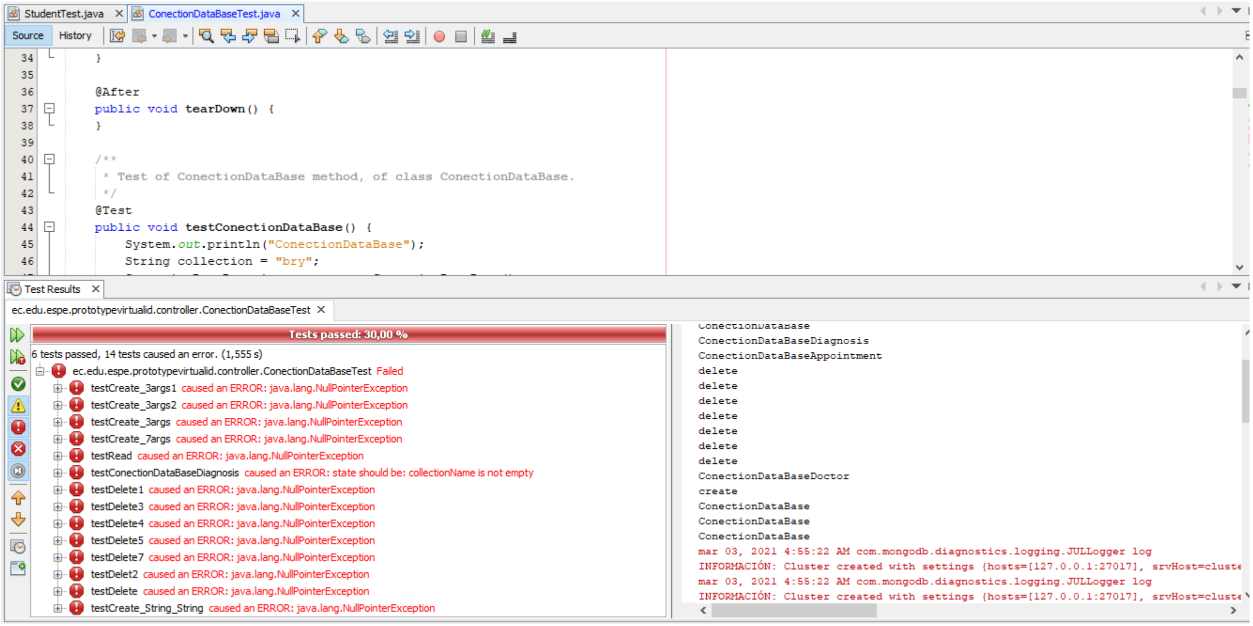
```java

    /**
     * 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);

    }
```



```
    }
    @Test
    public void testDelet2() {
        System.out.println("delete");
        String name = "Majof56";
        ConectionDataBase instance = new ConectionDataBase();
        boolean expResult = false;
        boolean result = instance.delete(name);
        assertEquals(expResult, result);

    }
    @Test
    public void testDelete3() {
        System.out.println("delete");
        String name = "Brake6";
        ConectionDataBase instance = new ConectionDataBase();
        boolean expResult = false;
        boolean result = instance.delete(name);
        assertEquals(expResult, result);

    }
    @Test
    public void testDelete4() {
        System.out.println("delete");
        String name = "Jhanny";
        ConectionDataBase instance = new ConectionDataBase();
        boolean expResult = false;
        boolean result = instance.delete(name);
        assertEquals(expResult, result);

    }
    @Test
    public void testDelete5() {
        System.out.println("delete");
```



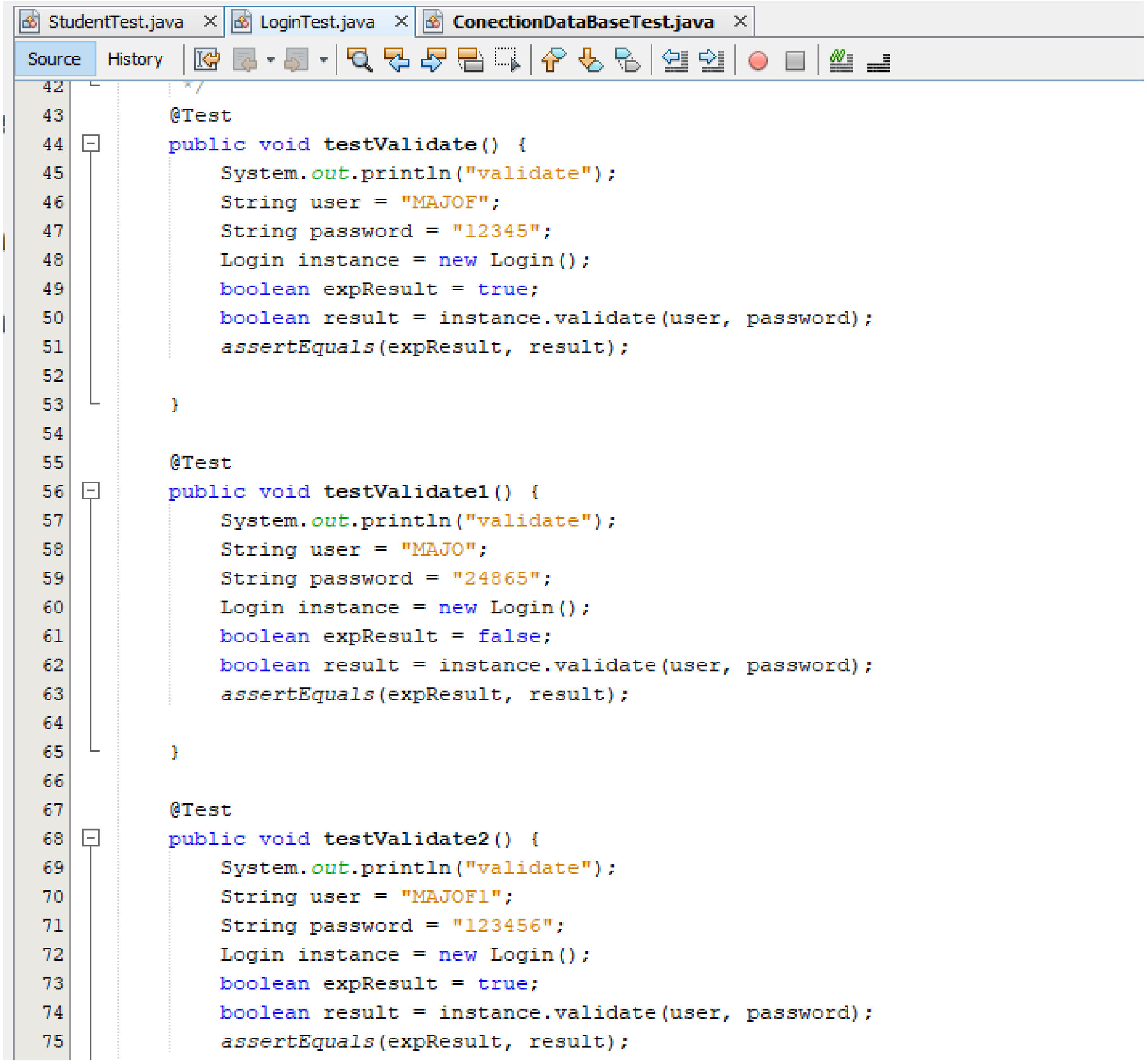
```java
    @Test
    public void testDelete4() {
        System.out.println("delete");
        String name = "Jhanny";
        ConectionDataBase instance = new ConectionDataBase();
        boolean expResult = false;
        boolean result = instance.delete(name);
        assertEquals(expResult, result);

    }
    @Test
    public void testDelete5() {
        System.out.println("delete");
        String name = "juan89";
        ConectionDataBase instance = new ConectionDataBase();
        boolean expResult = false;
        boolean result = instance.delete(name);
        assertEquals(expResult, result);

    }

    /**
     * Test of read method, of class ConectionDataBase.
     */
    @Test
    public void testRead() {
        System.out.println("read");
        ConectionDataBase instance = new ConectionDataBase();
        instance.read();
    }

}
```


StudentTest.java
ConectionDataBaseTest.java
Source
History
Test Results
ec.edu.espe.prototypevirtualid.controller.ConectionDataBaseTest
Tests passed: 30,00 %

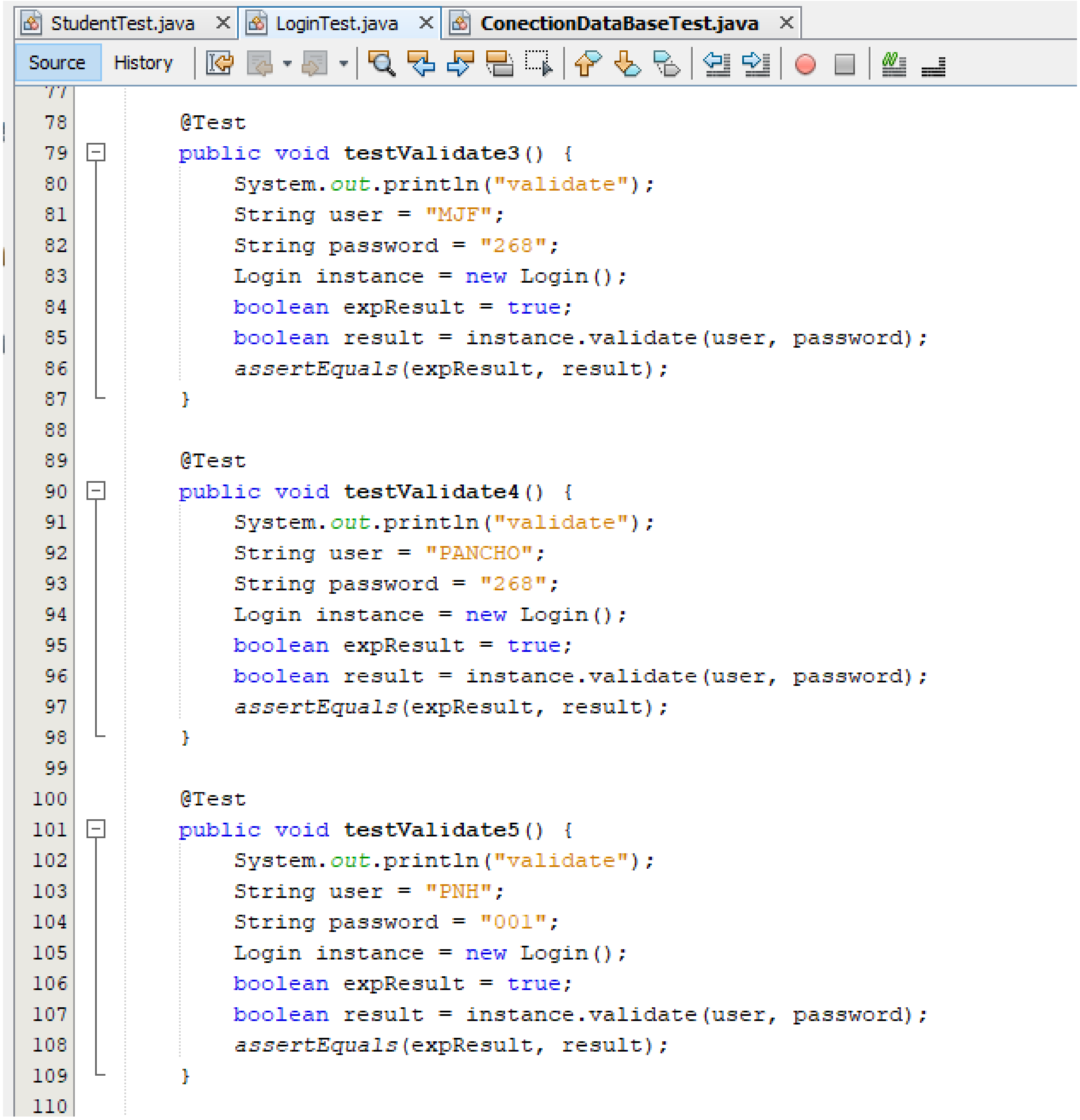
```
    }

    @After
    public void tearDown() {
    }

    /**
     * Test of ConectionDataBase method, of class ConectionDataBase.
     */
    @Test
    public void testConectionDataBase() {
        System.out.println("ConectionDataBase");
        String collection = "bry";
```

```
6 tests passed, 14 tests caused an error. (1,555 s)
ec.edu.espe.prototypevirtualid.controller.ConectionDataBaseTest 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
    testConectionDataBaseDiagnosis 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
```

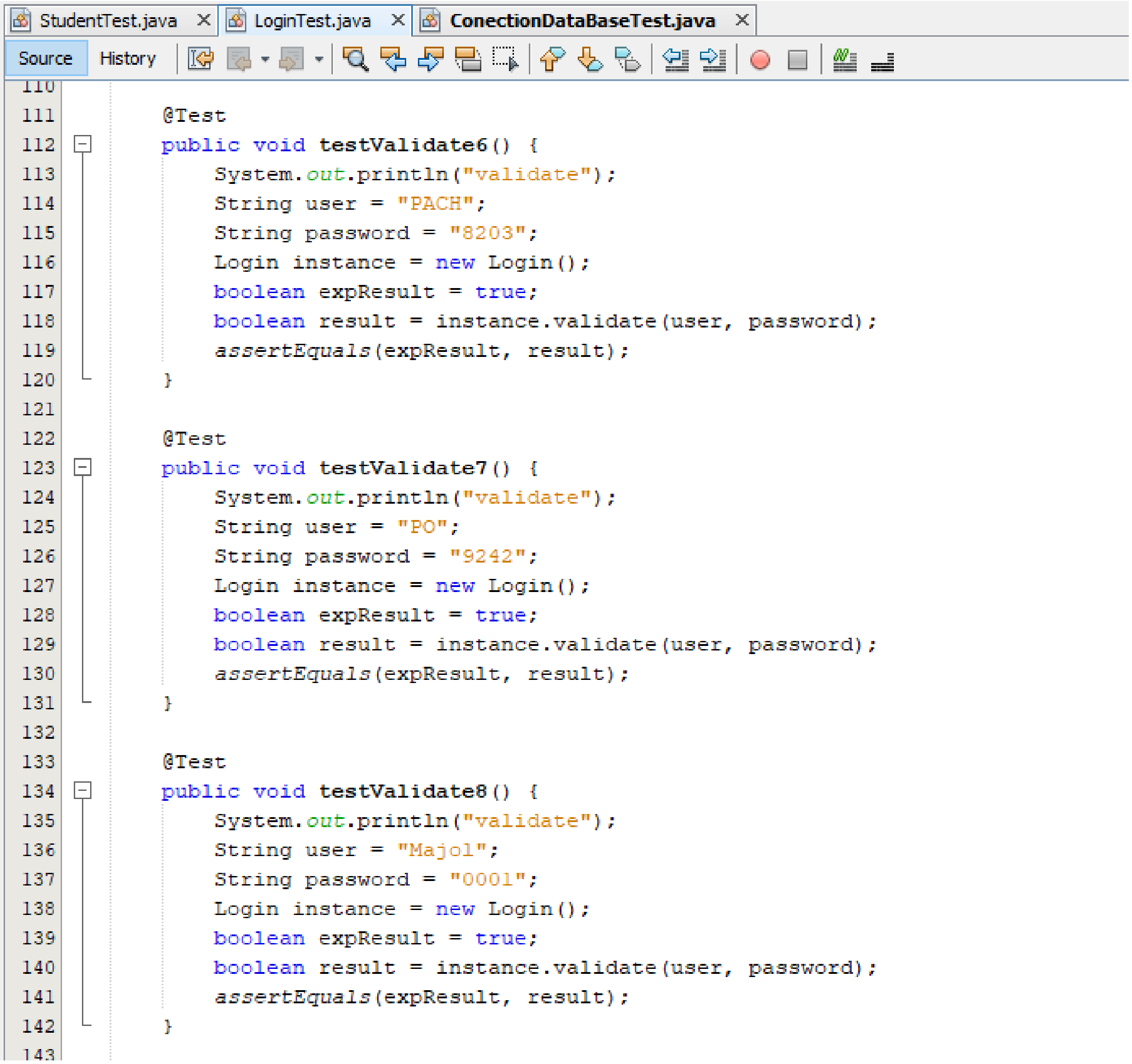
```
ConectionDataBase
ConectionDataBaseDiagnosis
ConectionDataBaseAppointment
delete
delete
delete
delete
delete
delete
delete
ConectionDataBaseDoctor
create
ConectionDataBase
ConectionDataBase
ConectionDataBase
mar 03, 2021 4:55:22 AM com.mongodb.diagnostics.logging.JULLogger log
INFORMACIÓN: Cluster created with settings {hosts=[127.0.0.1:27017], srvHost=cluste
mar 03, 2021 4:55:22 AM com.mongodb.diagnostics.logging.JULLogger log
INFORMACIÓN: Cluster created with settings {hosts=[127.0.0.1:27017], srvHost=cluste
```



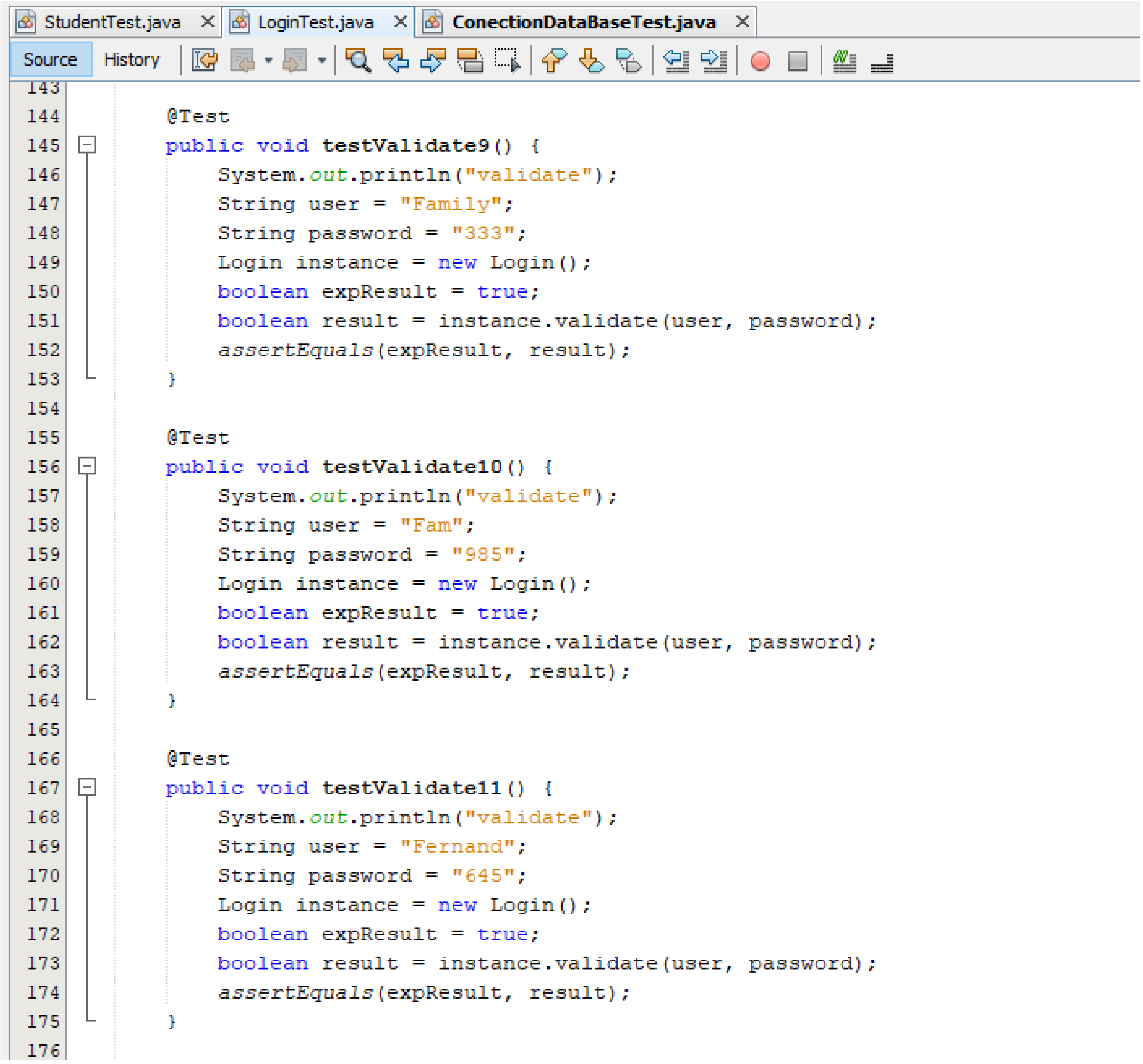
```java
     */
    @Test
    public void testValidate() {
        System.out.println("validate");
        String user = "MAJOF";
        String password = "12345";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);

    }

    @Test
    public void testValidate1() {
        System.out.println("validate");
        String user = "MAJO";
        String password = "24865";
        Login instance = new Login();
        boolean expResult = false;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);

    }

    @Test
    public void testValidate2() {
        System.out.println("validate");
        String user = "MAJOF1";
        String password = "123456";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
```



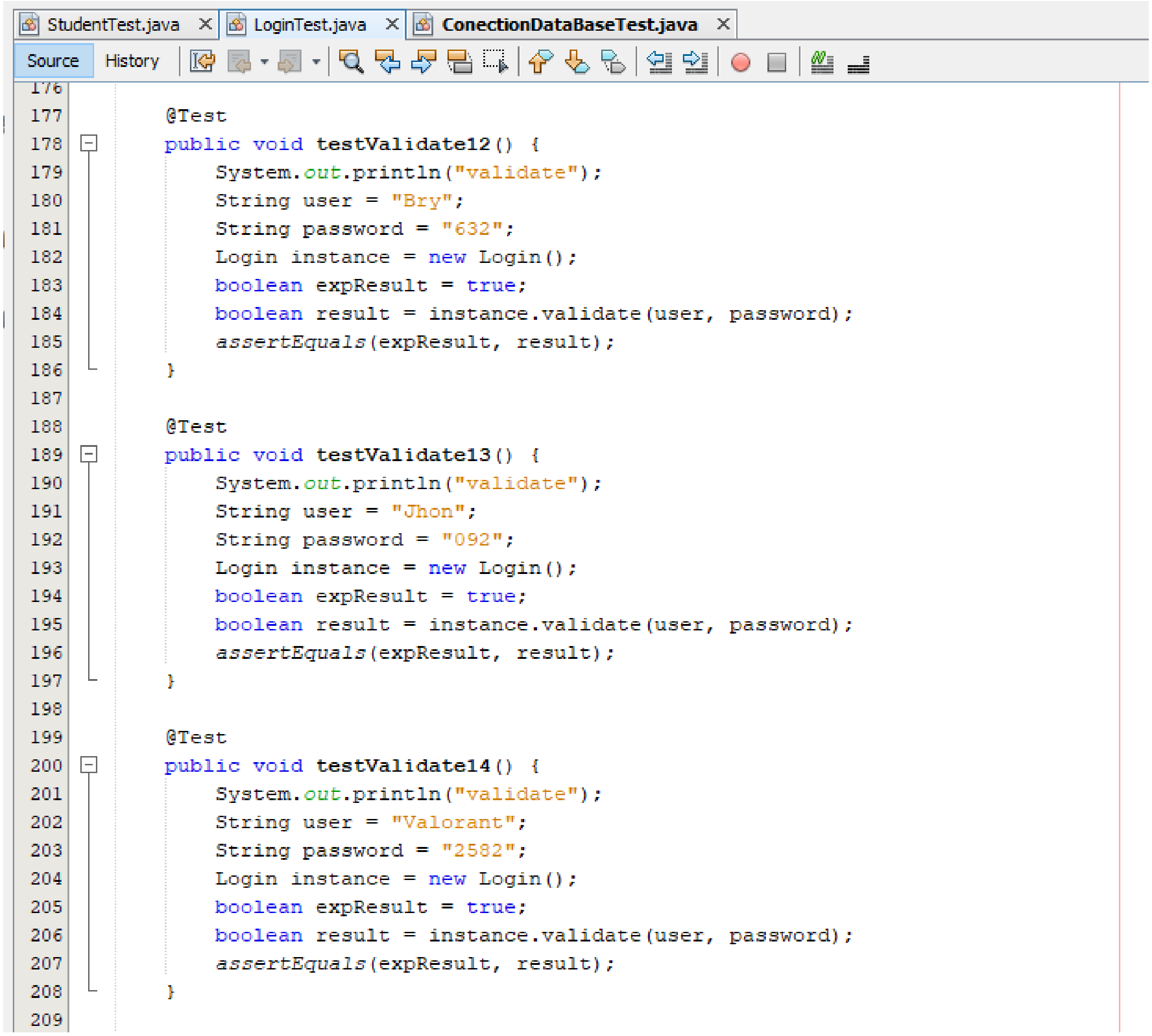
```java
    @Test
    public void testValidate3() {
        System.out.println("validate");
        String user = "MJF";
        String password = "268";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

    @Test
    public void testValidate4() {
        System.out.println("validate");
        String user = "PANCHO";
        String password = "268";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

    @Test
    public void testValidate5() {
        System.out.println("validate");
        String user = "PNH";
        String password = "001";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

```



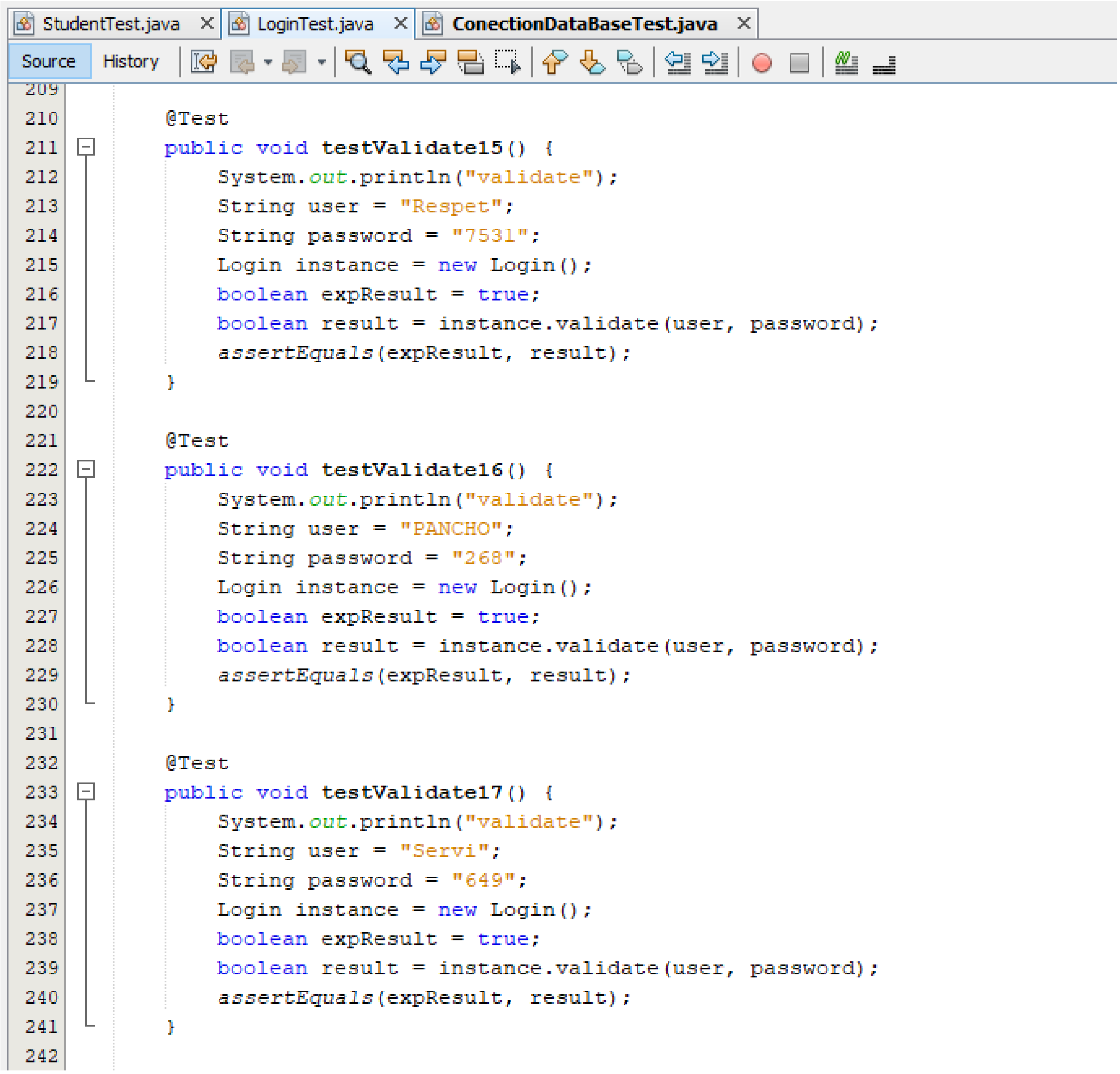
```java
    @Test
    public void testValidate6() {
        System.out.println("validate");
        String user = "PACH";
        String password = "8203";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

    @Test
    public void testValidate7() {
        System.out.println("validate");
        String user = "PO";
        String password = "9242";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

    @Test
    public void testValidate8() {
        System.out.println("validate");
        String user = "Majol";
        String password = "0001";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }
```



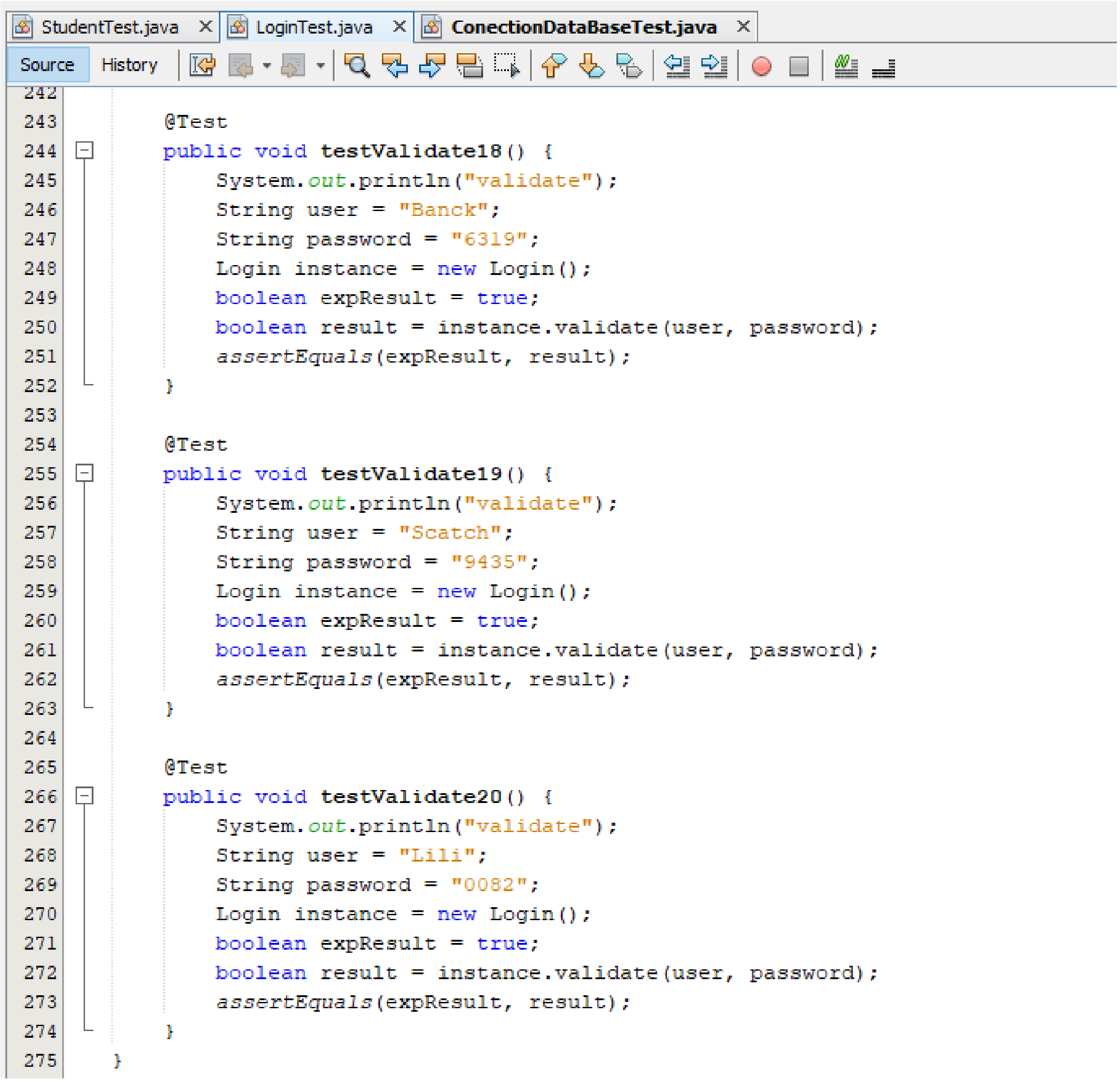
```
    @Test
    public void testValidate9() {
        System.out.println("validate");
        String user = "Family";
        String password = "333";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

    @Test
    public void testValidate10() {
        System.out.println("validate");
        String user = "Fam";
        String password = "985";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

    @Test
    public void testValidate11() {
        System.out.println("validate");
        String user = "Fernand";
        String password = "645";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

```



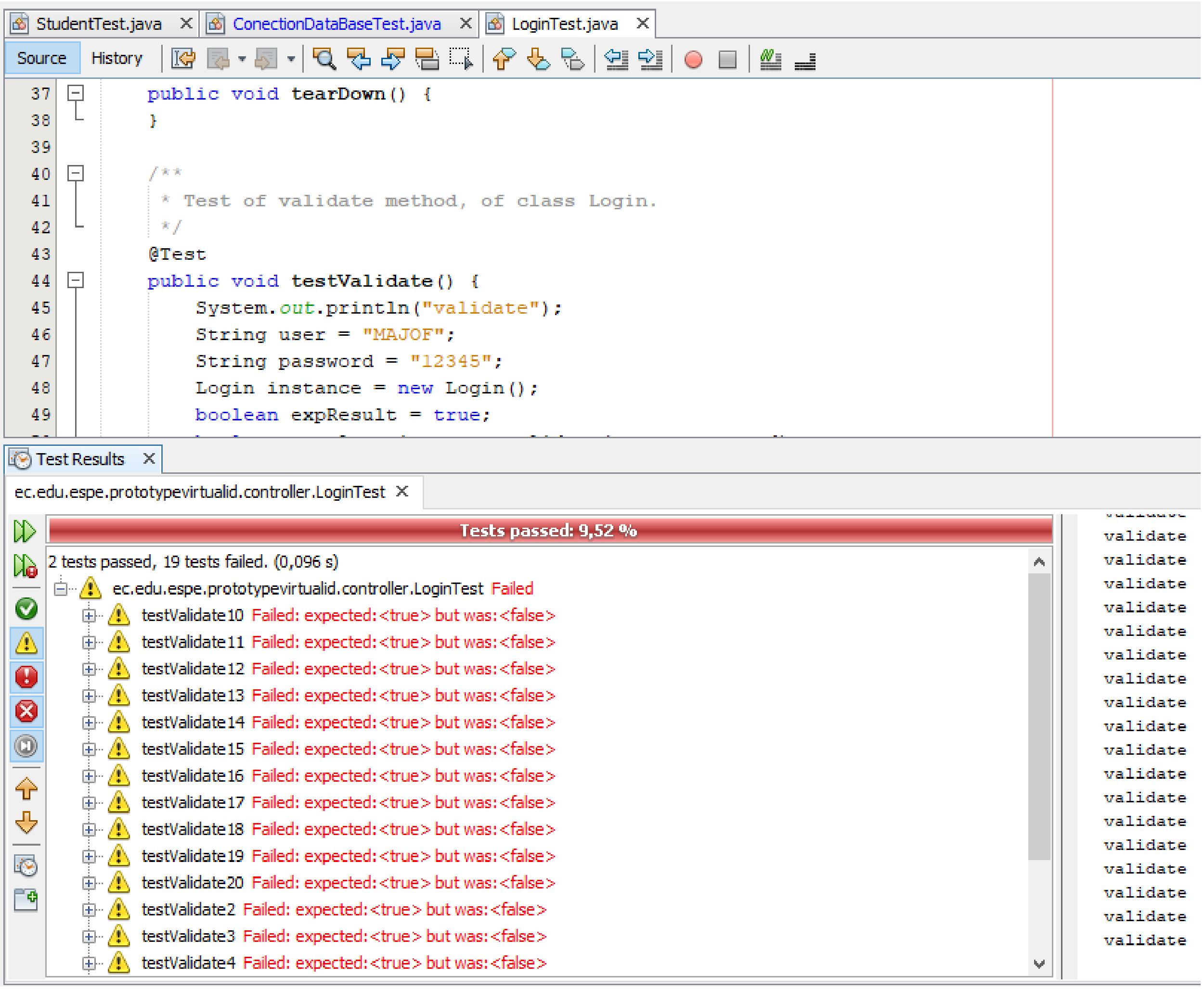
```java
    @Test
    public void testValidate12() {
        System.out.println("validate");
        String user = "Bry";
        String password = "632";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult,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

```



```
    @Test
    public void testValidate15() {
        System.out.println("validate");
        String user = "Respet";
        String password = "7531";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

    @Test
    public void testValidate16() {
        System.out.println("validate");
        String user = "PANCHO";
        String password = "268";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

    @Test
    public void testValidate17() {
        System.out.println("validate");
        String user = "Servi";
        String password = "649";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

```



```java
    @Test
    public void testValidate18() {
        System.out.println("validate");
        String user = "Banck";
        String password = "6319";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

    @Test
    public void testValidate19() {
        System.out.println("validate");
        String user = "Scatch";
        String password = "9435";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }

    @Test
    public void testValidate20() {
        System.out.println("validate");
        String user = "Lili";
        String password = "0082";
        Login instance = new Login();
        boolean expResult = true;
        boolean result = instance.validate(user, password);
        assertEquals(expResult, result);
    }
}
```


StudentTest.java
ConectionDataBaseTest.java
LoginTest.java
Source
History
Test Results
ec.edu.espe.prototypevirtualid.controller.LoginTest

```
    public void tearDown() {
    }

    /**
     * Test of validate method, of class Login.
     */
    @Test
    public void testValidate() {
        System.out.println("validate");
        String user = "MAJOF";
        String password = "12345";
        Login instance = new Login();
        boolean expResult = true;
```

Tests passed: 9,52 %

```
2 tests passed, 19 tests failed. (0,096 s)
ec.edu.espe.prototypevirtualid.controller.LoginTest Failed
    testValidate10 Failed: expected:<true> but was:<false>
    testValidate11 Failed: expected:<true> but was:<false>
    testValidate12 Failed: expected:<true> but was:<false>
    testValidate13 Failed: expected:<true> but was:<false>
    testValidate14 Failed: expected:<true> but was:<false>
    testValidate15 Failed: expected:<true> but was:<false>
    testValidate16 Failed: expected:<true> but was:<false>
    testValidate17 Failed: expected:<true> but was:<false>
    testValidate18 Failed: expected:<true> but was:<false>
    testValidate19 Failed: expected:<true> but was:<false>
    testValidate20 Failed: expected:<true> but was:<false>
    testValidate2 Failed: expected:<true> but was:<false>
    testValidate3 Failed: expected:<true> but was:<false>
    testValidate4 Failed: expected:<true> but was:<false>
```

```
validate
validate
validate
validate
validate
validate
validate
validate
validate
validate
validate
validate
validate
validate
validate
validate
validate
validate
```