**CA OOSE | Part B**

Sérgio Vinício da Silva Oliveira

13 Dec 2023

The link also allows access to this content

<https://github.com/sergio-oliveira-br/OOSE>

Note

First, I would like to mention that Nickolas Franco gave up this course. For this reason, the development of this was carried out only by me.

1 / 17

1. Development of the class in Java .............................................................................. 1

2. Full test of the class ........................................................................................................ 9

3. Details of the methodology followed..................................................................... 16

Extra

4. Codes ……………………………………………………………………………………………………..17

1 / 17

**1. Development of the class in Java.**

The classes chosen for development are related to the Login attribute, executed by the garden centre cashier.

A screenshot of a computer screen

Description automatically generated

As a main flow, we have the user carrying his username and password in the system registration. In possession of these details, the system would allow the user to access the sales system.

When contingency mode is enabled, an alternate flow is enabled when an administrator key is present.

In the system's exception flow, any attempt not registered by the users is denied access.

The flowchart below can be used to represent the reasoning behind the program in question.

A diagram of a diagram

Description automatically generated

The class diagram, description of input, main process and output are part of the programming that has been performed.

A diagram of a computer

Description automatically generated

Communication Diagram

A diagram of a user flow

Description automatically generated

Verify system functionality by examining evidence.

Keep in mind that the link holds all the evidence found in this, including the codes created.

<https://github.com/sergio-oliveira-br/OOSE>

Class – Login.java

A screenshot of a computer

Description automatically generated

Class Signup.java

A screenshot of a computer

Description automatically generated

Class Users.java

A screenshot of a computer

Description automatically generated

Login.java

Program executed with incorrect credentials.

A computer screen shot of a computer screen

Description automatically generated

Login.java

Program executed with right credentials.

A screenshot of a computer

Description automatically generated

**2. Full test of the class.**

The classes chosen for development are related to the Login attribute, executed by the garden centre cashier.

The contract below contains the necessary details.

A screenshot of a login form

Description automatically generated

9 / 17

The creation of the login method requires each user to have a username and password to ensure their roles are only accessed.

This project will not address the process of creating users.

The proposal here outlines the system's capabilities, testing only the individual components of the program. In other words, the approach is to verify the login functionality, which will determine whether or not access to the system is allowed.

The policy adopted addresses testing all possible combinations, including as guidelines correct and incorrect entries, being user input fundamental. Moreover, two users have been created to verify the extension of registered users.

To put it simply, the tests were:

|  |  |  |
| --- | --- | --- |
| Name: | | testLoginBlank |
| Input | username: | " " |
| password: | " " |
| Output | ExpResult: | 0 |
| Result: | 0 |
| Status: | | Passed |
| Comments: | To validate null entries | |
|  |

|  |  |  |
| --- | --- | --- |
| Name: | | testLoginWrongPassword |
| Input | username: | oliveira-sergio |
| password: | BraziL123 |
| Output | ExpResult: | 0 |
| Result: | 0 |
| Status: | | Passed |
| Comments: | To validate incorrect entries.Note that the 'L' is written in capital letters, but it's incorrect. | |
|  |

|  |  |  |
| --- | --- | --- |
| Name: | | testLoginAllRightSergio |
| Input | username: | oliveira-sergio |
| password: | Brazil123 |
| Output | ExpResult: | 1 |
| Result: | 1 |
| Status: | | Passed |
| Comments: | To validate correct entries | |
|  |

|  |  |  |
| --- | --- | --- |
| Name: | | testLoginAllRightAdmin |
| Input | username: | Admin123 |
| password: | Admin123 |
| Output | ExpResult: | 1 |
| Result: | 1 |
| Status: | | Passed |
| Comments: | To assess the length of the list that was created. | |
|  |

|  |  |  |
| --- | --- | --- |
| Name: | | testLoginAllRightAdminButExpResultWrong |
| Input | username: | Admin123 |
| password: | Admin123 |
| Output | ExpResult: | 0 |
| Result: | 1 |
| Status: | | Failed |
| Comments: | To confirm the accuracy of the tests carried out. ExpResult should be 1. | |
|  |

The evidence of the tests performed can be found below.

Note that one of the results did not pass the test. This was due to the test validating the accuracy of the test.

Junit Test – Test Results

A screenshot of a computer

Description automatically generated

Junit Test – Test Results

A screenshot of a computer

Description automatically generated

LoginTest.java

A screenshot of a computer

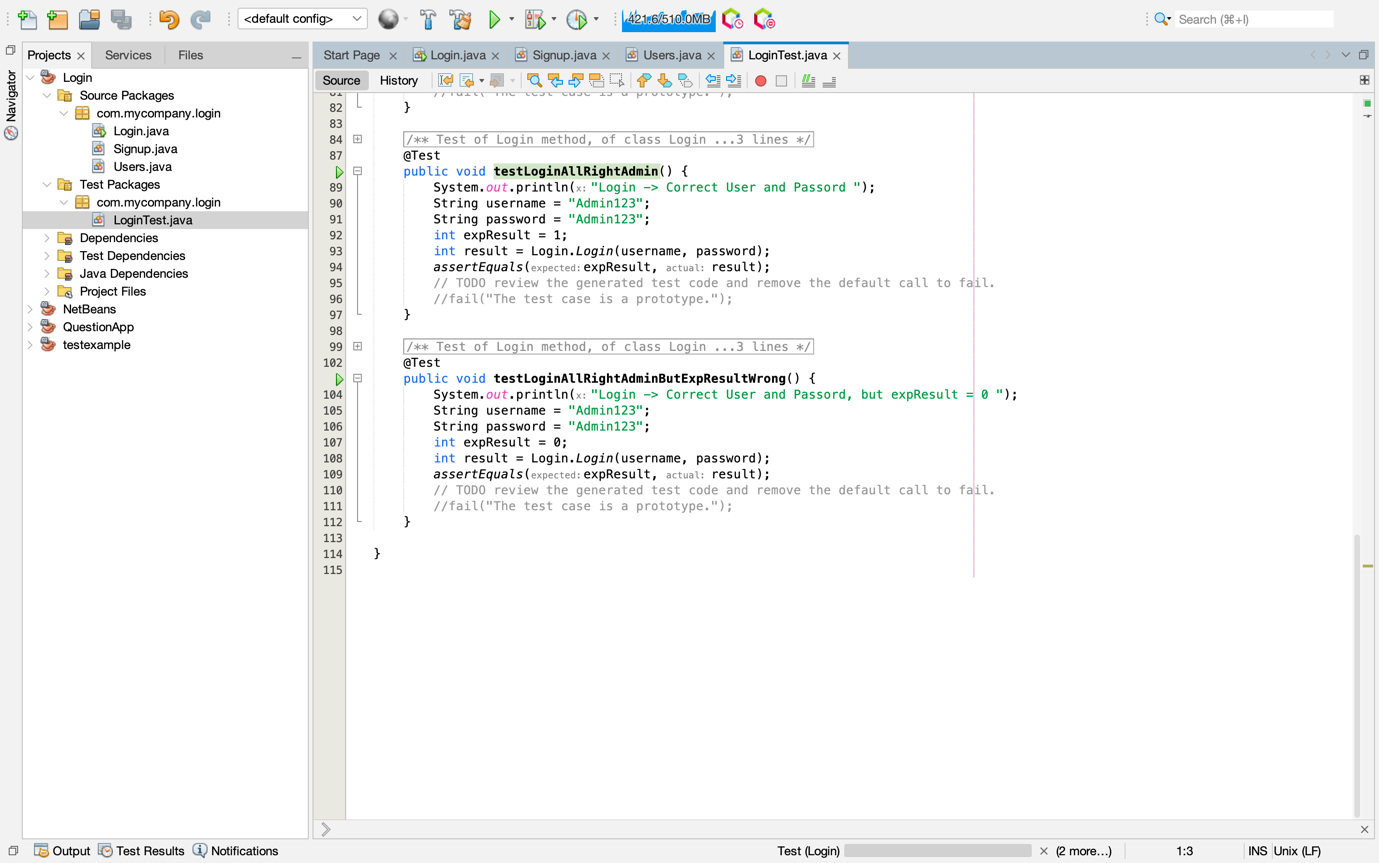
Description automatically generated

LoginTest.java

A screenshot of a computer

Description automatically generated

LoginTest.java



**3. Details of the methodology followed.**

The methodology used in the tests followed the principles of Test Oriented Development hereinafter TDD.

The reason for this choice is due to the rapid test cycle, coding and revaluation.

JUnit was the tool chosen to test the frameworks and is justified by the fact that the test is focused only on the classes and methods at hand. Moreover, for have the ability to be fast.

Based on the choices that I have made and the test results, we can conclude that the login method is free from faults. And it was guaranteed thanks to the TDD applied in JUnit.

The burndown chart below illustrates the way I worked during this stage.

Activities like functional class development, test development, and reports were considered for the mapped tests.



16 / 17

User Stories

A diagram of a computer program

Description automatically generated

Keep in mind that the link holds all the evidence found in this, including the codes created.

<https://github.com/sergio-oliveira-br/OOSE>

CODE

Main Class - Login

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\*/

package com.mycompany.login;

import java.util.Scanner;

/\*\*

\*

\* @author sergiovinicio

\*/

public class Login

{

public static void main(String[] args)

{

System.out.println("CA | OOSE Part B");

System.out.println("Welcome to the garden center."

+ " Accessing the point of sale requires a login.");

//declare variables

String username;

String password;

//declare object

Scanner sc = new Scanner(System.in);

//input

System.out.println("Please enter your Username");

username = sc.nextLine();

System.out.println("Please enter your Password");

password = sc.nextLine();

// Create an instance of the Users class

Users myU = new Users();

// Process

Signup signedUpUser = myU.signUp(username);

//output

if (signedUpUser != null && signedUpUser.getPassword().equals(password))

{

System.out.println("Login successful");

}

else

{

System.out.println("Login failed. Invalid username or password.");

}

}

Login created to test

public static int Login (String username, String password)

{

// Create an instance

Users myU = new Users();

// Process

Signup signedUpUser = myU.signUp(username);

//output

if (signedUpUser != null && signedUpUser.getPassword().equals(password))

{

System.out.println("Login successful");

return 1;

}

else

{

System.out.println("Login failed. Invalid username or password.");

return 0;

}

}

Class Users

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package com.mycompany.login;

import java.util.ArrayList;

import java.util.List;

/\*\*

\*

\* @author sergiovinicio

\*/

public class Users

{

private final List<Signup> user = new ArrayList<>();

public Users()

{

user.add(new Signup("oliveira-sergio", "Brazil123"));

user.add(new Signup("Admin123", "Admin123"));

}

public Signup signUp (String username)

{

for (Signup i : user)

{

if (i.getUsername().equals(username))

{

return i;

}

}

return null;

}

}

Class SingUp

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template

\*/

package com.mycompany.login;

/\*\*

\*

\* @author sergiovinicio

\*/

public class Signup

{

private String username;

private String password;

public Signup(String username, String password)

{

this.username = username;

this.password = password;

}

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

}

Class Login Test – By Junit Test

/\*

\* Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license

\* Click nbfs://nbhost/SystemFileSystem/Templates/UnitTests/JUnit5TestClass.java to edit this template

\*/

package com.mycompany.login;

import org.junit.jupiter.api.AfterEach;

import org.junit.jupiter.api.AfterAll;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.BeforeAll;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.\*;

/\*\*

\*

\* @author sergiovinicio

\*/

public class LoginTest {

public LoginTest() {

}

@BeforeAll

public static void setUpClass() {

}

@AfterAll

public static void tearDownClass() {

}

@BeforeEach

public void setUp() {

}

@AfterEach

public void tearDown() {

}

/\*\*

\* Test of Login method, of class Login.

\*/

@Test

public void testLoginBlank() {

System.out.println("Login -> All Blank");

String username = "";

String password = "";

int expResult = 0;

int result = Login.Login(username, password);

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 Login method, of class Login.

\*/

@Test

public void testLoginAllRightSergio() {

System.out.println("Login -> Correct User and Password");

String username = "oliveira-sergio";

String password = "Brazil123";

int expResult = 1;

int result = Login.Login(username, password);

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 Login method, of class Login.

\*/

@Test

public void testLoginWrongPassword() {

System.out.println("Login -> Correct User and Wrong Passord ");

String username = "oliveira-sergio";

String password = "BraziL123";

int expResult = 0;

int result = Login.Login(username, password);

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 Login method, of class Login.

\*/

@Test

public void testLoginAllRightAdmin() {

System.out.println("Login -> Correct User and Passord ");

String username = "Admin123";

String password = "Admin123";

int expResult = 1;

int result = Login.Login(username, password);

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 Login method, of class Login.

\*/

@Test

public void testLoginAllRightAdminButExpResultWrong() {

System.out.println("Login -> Correct User and Passord, but expResult = 0 ");

String username = "Admin123";

String password = "Admin123";

int expResult = 0;

int result = Login.Login(username, password);

assertEquals(expResult, result);

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

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

}

}