**SourceCode**

**User.java**

**package** UserEntity;

**public** **class** User {

String username;

String password;

**public** User(String username, String password) {

**super**();

**this**.username = username;

**this**.password = password;

}

**public** User() {

// **TODO** Auto-generated constructor stub

}

**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;

}

@Override

**public** String toString() {

**return** "User [username=" + username + ", password=" + password + "]";

}

// checkCredentials

**public** **boolean** authentication(String username, String password) {

**if** ((username == **null**) || (password == **null**)) {

**return** **false**;

}

**if** (username.equalsIgnoreCase("Anand123")) {

**if** (password.equals("Alfa@12")) {

**return** **true**;

}

}

**return** **false**;

}

}

**Authentication.java**

package Authentication;

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

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

import org.junit.jupiter.api.Test;

import UserEntity.User;

class UserAuthentication {

@BeforeAll

public static void beforeAll() {

System.out.println("Testing going to start..");

}

@AfterAll

public static void afterAll() {

System.out.println("Testing Completed");

}

@BeforeEach

public void beforeEach() {

System.out.println("User Authentication Initiated");

}

@AfterEach

public void tearDown() {

System.out.println("User Authentication Completed");

}

@Test

public void getAndSetUsername() {

User testUser = new User();

testUser.setUsername("Anand");

assertEquals(testUser.getUsername(), "Anand");

System.out.println("Username is Valid");

}

@Test

public void getAndSetPassword() {

User testUser = new User();

testUser.setPassword("mypassword");

assertEquals(testUser.getPassword(), "mypassword");

System.out.println("Password is Valid");

}

@Test

public void checkToString() {

User testUser = new User();

assertNotNull(testUser.toString());

System.out.println("toString is null, when create Default Constructor");

}

@Test

public void checkConstructor() {

User testUser = new User("Anand", "pass@123");

User checkUser = new User();

System.out

.println("Compare Credentilas Default Constructor (with getter-setter) and Parameterized Constructor");

checkUser.setUsername("Anand");

checkUser.setPassword("pass@123");

assertEquals(testUser.getUsername(), checkUser.getUsername());

assertEquals(testUser.getPassword(), checkUser.getPassword());

System.out.println("Both Constructor are giving same result when passes same unsername and password");

}

@Test

public void testDefaultConstructor() {

User testUser = new User();

assertNotNull(testUser);

System.out.println("Test Default Constructor");

System.out.println("Default Constructor is created");

}

@Test

public void authentication() {

User user = new User();

System.out.println("Username : Anand123 , Password :Alfa@12");

assertTrue(user.authentication("Anand123", "Alfa@12"));

}

@Test

public void authentication1() {

User user = new User();

System.out.println("Username : null , Password :null");

assertFalse(user.authentication("", ""));

}

@Test

public void authentication2() {

User user = new User();

System.out.println("Username : Anand123 , Password :null");

assertFalse(user.authentication("Anand123", ""));

}

@Test

public void authentication3() {

User user = new User();

System.out.println("Username : null , Password :Alfa@12");

assertFalse(user.authentication("", "Alfa@12"));

}

@Test

public void authentication4() {

User user = new User();

System.out.println("Username : Anand20222 , Password :Anan@1223");

assertFalse(user.authentication("Anand20222", "Anan@1223"));

}

}