**Exercise 9: Parameterized Test with JUnit**

**Task**: Use @ParameterizedTest to test multiple inputs. Test: Write code for this.

**UserService.java :-**

public class UserService {

public boolean isValidEmail(String email) {

return email != null && email.contains("@") && email.endsWith(".com");

}

}

**UserServiceParamTest.java :-**

package com.example.demo;

import com.example.demo.service.UserService;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.function.Executable;

import org.junit.jupiter.params.ParameterizedTest;

import org.junit.jupiter.params.provider.ValueSource;

import java.util.ArrayList;

import java.util.List;

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

public class UserServiceParamTest {

private final UserService userService = new UserService();

@Test

void testAllValidEmails() {

String[] validEmails = {

"naresh@example.com",

"another@sample.com",

"user123@domain.com"

};

List<Executable> assertions = new ArrayList<>();

for (String email : validEmails) {

assertions.add(() -> assertTrue(userService.isValidEmail(email), "Expected valid: " + email));

}

assertAll("Valid emails", assertions);

}

@Test

void testAllInvalidEmails() {

String[] invalidEmails = {

"invalid-email",

"test@domain.org",

"",

"null"

};

List<Executable> assertions = new ArrayList<>();

for (String email : invalidEmails) {

String input = "null".equals(email) ? null : email;

assertions.add(() -> assertFalse(userService.isValidEmail(input), "Expected invalid: " + email));

}

assertAll("Invalid emails", assertions);

}

}

Output :-

