**Day2 Lab1: JUnit 5 parameterized tests**

**JUnit 5 parameterized tests** in **IntelliJ IDEA** using @ParameterizedTest, @ValueSource, and @CsvSource.

**Setup**

Make sure you have these dependencies in your **Maven** pom.xml:

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter</artifactId>

<version>5.10.0</version>

<scope>test</scope>

</dependency>

**Exercise 1: Example with @ValueSource**

**Use case**: Check if numbers are even.

import org.junit.jupiter.params.ParameterizedTest;

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

import static org.junit.jupiter.api.Assertions.assertTrue;

public class EvenNumberTest {

@ParameterizedTest

@ValueSource(ints = {2, 4, 6, 8, 10})

void testEvenNumbers(int number) {

assertTrue(number % 2 == 0, number + " should be even");

}

}

**Runs the same test for values 2, 4, 6, 8, 10.**

**Exercise 2: Example with @CsvSource**

**Use case**: Check string length.

import org.junit.jupiter.params.ParameterizedTest;

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

import static org.junit.jupiter.api.Assertions.assertEquals;

public class StringLengthTest {

@ParameterizedTest

@CsvSource({

"hello, 5",

"JUnit, 5",

"Java, 4"

})

void testStringLength(String word, int expectedLength) {

assertEquals(expectedLength, word.length());

}

}

**Runs test with different string–length pairs.**

**Exercise 3: Example with @CsvSource (Multiple Arguments)**

**Use case**: Check sum of two numbers.

import org.junit.jupiter.params.ParameterizedTest;

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

import static org.junit.jupiter.api.Assertions.assertEquals;

public class CalculatorTest {

@ParameterizedTest

@CsvSource({

"2, 3, 5",

"10, 20, 30",

"-5, 5, 0"

})

void testAddition(int a, int b, int expectedSum) {

assertEquals(expectedSum, a + b);

}

}

**Runs test for (2+3=5), (10+20=30), (-5+5=0).**

**Exercise 4: Example with @ValueSource (Strings)**

**Use case**: Validate palindrome words.

import org.junit.jupiter.params.ParameterizedTest;

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

import static org.junit.jupiter.api.Assertions.assertTrue;

public class PalindromeTest {

boolean isPalindrome(String word) {

return new StringBuilder(word).reverse().toString().equalsIgnoreCase(word);

}

@ParameterizedTest

@ValueSource(strings = {"madam", "racecar", "level", "radar"})

void testPalindromeWords(String candidate) {

assertTrue(isPalindrome(candidate), candidate + " should be a palindrome");

}

}

**Tests all words as palindromes.**

IntelliJ will run all parameterized cases separately (you’ll see them listed in the test runner).