Advanced JUnit Testing Exercises

Exercise 1: Parameterized Tests

**File: EvenChecker.java**

public class EvenChecker {

public boolean isEven(int number) {

return number % 2 == 0;

}

}

**File: EvenCheckerTest.java**

import org.junit.jupiter.params.ParameterizedTest;

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

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

class EvenCheckerTest {

@ParameterizedTest

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

void testIsEven(int number) {

EvenChecker checker = new EvenChecker();

assertTrue(checker.isEven(number));

}

}

**OUTPUT:**



Exercise 2: Test Suites and Categories

**File: Calculator.java**

public class Calculator {

public int add(int a, int b) {

return a + b;

}

public int multiply(int a, int b) {

return a \* b;

}

}

**File: AdditionTest.java**

import org.junit.jupiter.api.Test;

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

class AdditionTest {

@Test

void testAdd() {

Calculator calc = new Calculator();

assertEquals(15, calc.add(10, 5));

}

}

**File: MultiplicationTest.java**

import org.junit.jupiter.api.Test;

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

class MultiplicationTest {

@Test

void testMultiply() {

Calculator calc = new Calculator();

assertEquals(20, calc.multiply(4, 5));

}

}

**File: AllTests.java**

import org.junit.platform.suite.api.SelectClasses;

import org.junit.platform.suite.api.Suite;

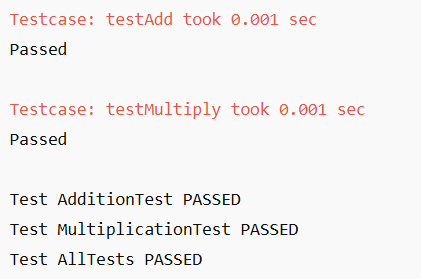
@Suite

@SelectClasses({AdditionTest.class, MultiplicationTest.class})

public class AllTests {

}

**OUTPUT:**

****

Exercise 3: Test Execution Order

**File: OrderedTests.java**

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

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

@TestMethodOrder(MethodOrderer.OrderAnnotation.class)

class OrderedTests {

@Test

@Order(3)

void testC() {

assertTrue(true);

}

@Test

@Order(1)

void testA() {

assertEquals(2, 1 + 1);

}

@Test

@Order(2)

void testB() {

assertFalse(5 < 3);

}

}

**OUTPUT:**

****

Exercise 4: Exception Testing

**File: ExceptionThrower.java**

public class ExceptionThrower {

public void throwException() {

throw new IllegalArgumentException("Invalid input");

}

}

**File: ExceptionThrowerTest.java**

import org.junit.jupiter.api.Test;

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

class ExceptionThrowerTest {

@Test

void testThrowException() {

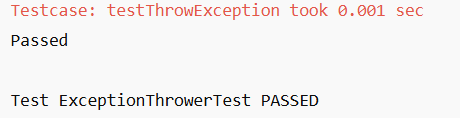
ExceptionThrower et = new ExceptionThrower();

assertThrows(IllegalArgumentException.class, et::throwException);

}

}

**OUTPUT:**



Exercise 5: Timeout and Performance Testing

**File: PerformanceTester.java**

public class PerformanceTester {

public void performTask() {

for (int i = 0; i < 1000000; i++) {

Math.sqrt(i);

}

}

}

**File: PerformanceTesterTest.java**

import org.junit.jupiter.api.Test;

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

import java.time.Duration;

class PerformanceTesterTest {

@Test

void testPerformTaskCompletesOnTime() {

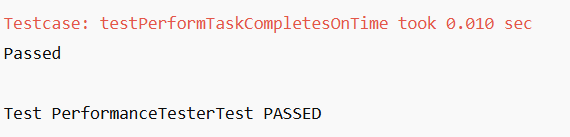
PerformanceTester tester = new PerformanceTester();

assertTimeout(Duration.ofMillis(100), tester::performTask);

}

}

**OUTPUT:**

****