**JUNIT ADVANCED TESTING**

**Exercise 1: Parameterized Tests**

**// EvenChecker.java**

package org.example;

public class EvenChecker {

public boolean isEven(int number) {

return number % 2 == 0;

}

}

**// EvenCheckerTest.java**

package org.example;

import org.junit.jupiter.params.ParameterizedTest;

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

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

public class EvenCheckerTest {

private final EvenChecker checker = new EvenChecker();

@ParameterizedTest

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

public void testIsEven\_TrueCases(int input) {

assertTrue(checker.isEven(input), input + " should be even");

}

@ParameterizedTest

@ValueSource(ints = {1, 3, 5, 7, 9})

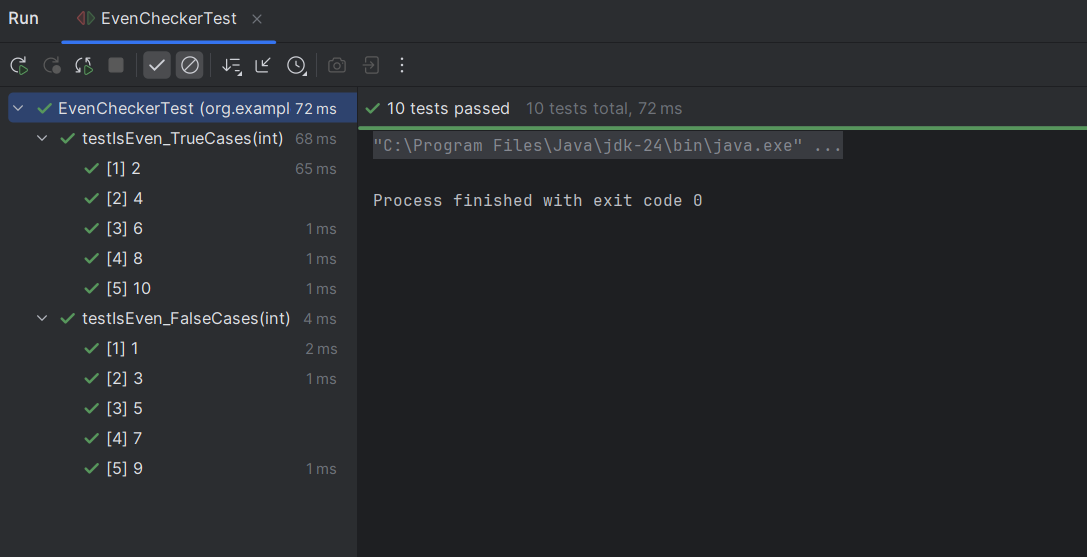
public void testIsEven\_FalseCases(int input) {

assertFalse(checker.isEven(input), input + " should be odd");

}

}

**Output:**



**Exercise 2: Test Suites and Categories**

**// Calculator.java**

package org.example;  
  
public class Calculator {  
 public int add(int a, int b) {  
 return a + b;  
 }  
  
 public int subtract(int a, int b) {  
 return a - b;  
 }  
}

**// CalculatorTest.java**

package org.example;  
  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class CalculatorTest {  
  
 @Test  
 void testAdd() {  
 *assertEquals*(5, 2 + 3);  
 }  
  
 @Test  
 void testSubtract() {  
 *assertEquals*(1, 4 - 3);  
 }  
}

**// EvenChecker.java**

package org.example;  
  
public class EvenChecker {  
  
 public boolean isEven(int number) {  
 return number % 2 == 0;  
 }  
}

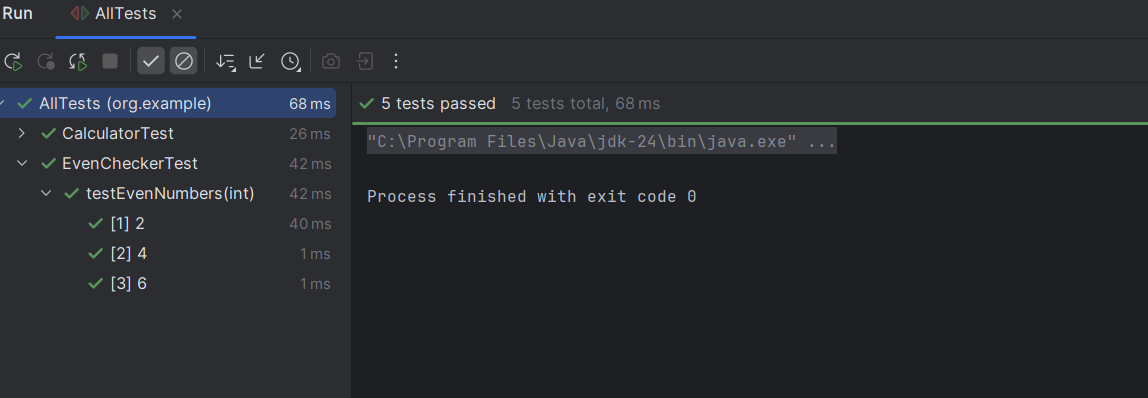
**// EvenCheckerTest.java**

package org.example;  
  
import org.junit.jupiter.params.ParameterizedTest;  
import org.junit.jupiter.params.provider.ValueSource;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class EvenCheckerTest {  
  
 @ParameterizedTest  
 @ValueSource(ints = {2, 4, 6})  
 void testEvenNumbers(int input) {  
 *assertTrue*(input % 2 == 0);  
 }  
}

**// AllTests.java**

package org.example;  
  
import org.junit.platform.suite.api.SelectClasses;  
import org.junit.platform.suite.api.Suite;  
  
import org.example.CalculatorTest;  
import org.example.EvenCheckerTest;  
  
@Suite  
@SelectClasses({  
 CalculatorTest.class,  
 EvenCheckerTest.class  
})  
public class AllTests {  
 // No methods needed; this class just runs the suite  
}

**Output:**



**Exercise 3: Test Execution Order**

**// OrderedTests.java**

package org.example;

import org.junit.jupiter.api.Test;

import org.junit.jupiter.api.Order;

import org.junit.jupiter.api.TestMethodOrder;

import org.junit.jupiter.api.MethodOrderer.OrderAnnotation;

@TestMethodOrder(OrderAnnotation.class) // Enable ordered execution

public class OrderedTests {

@Test

@Order(3)

void testC() {

System.out.println("Test C - Order 3");

}

@Test

@Order(1)

void testA() {

System.out.println("Test A - Order 1");

}

@Test

@Order(2)

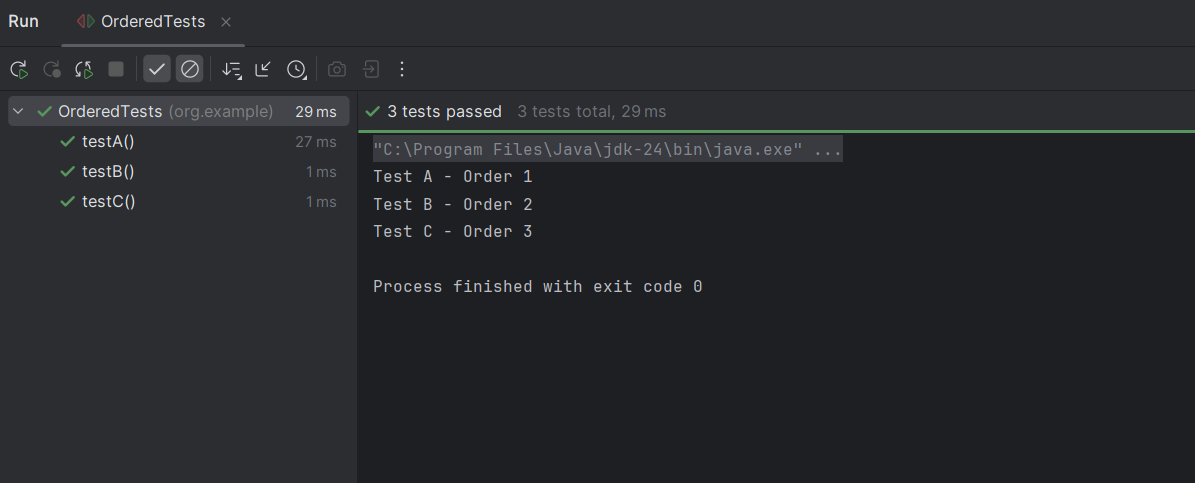
void testB() {

System.out.println("Test B - Order 2");

}

}

**Output:**



**Exercise 4: Exception Testing**

**// ExceptionThrower.java**

package org.example;

public class ExceptionThrower {

public void throwException(String input) {

if (input == null) {

throw new IllegalArgumentException("Input cannot be null");

}

}

}

**// ExceptionThrowerTest.java**

package org.example;

import org.junit.jupiter.api.Test;

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

public class ExceptionThrowerTest {

@Test

void testThrowsException() {

ExceptionThrower et = new ExceptionThrower();

IllegalArgumentException thrown = assertThrows(

IllegalArgumentException.class,

() -> et.throwException(null),

"Expected throwException(null) to throw IllegalArgumentException"

);

assertEquals("Input cannot be null", thrown.getMessage());

}

@Test

void testNoException() {

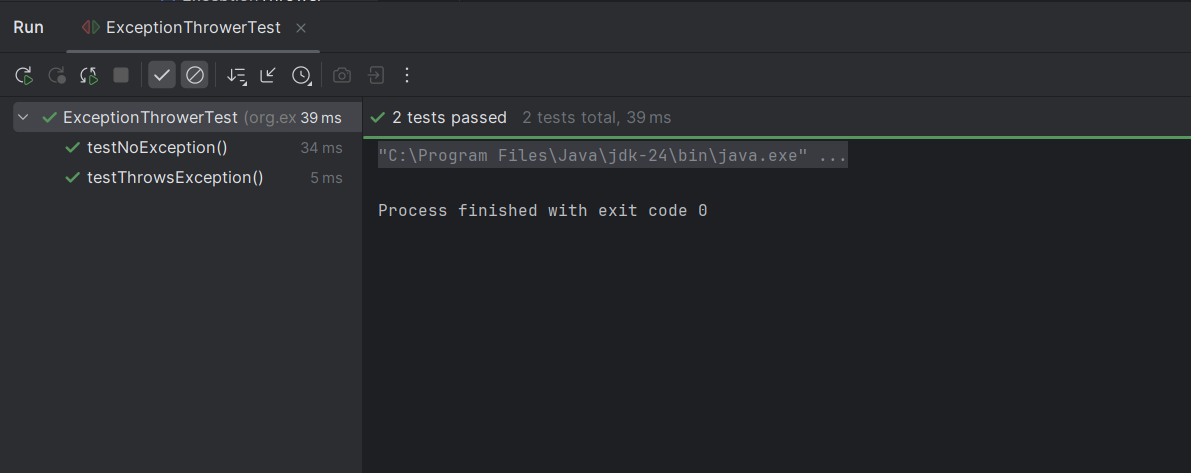
ExceptionThrower et = new ExceptionThrower();

assertDoesNotThrow(() -> et.throwException("Hello"));

}

}

**Output:**



**Exercise 5: Timeout and Performance Testing**

**// PerformanceTester.java**

package org.example;

public class PerformanceTester {

public void performTask() {

try {

// Simulate some delay (e.g., long computation or I/O)

Thread.sleep(300); // 300 milliseconds

} catch (InterruptedException e) {

Thread.currentThread().interrupt();

}

}

}

**// PerformanceTesterTest.java**

package org.example;

import org.junit.jupiter.api.Test;

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

import java.time.Duration;

public class PerformanceTesterTest {

@Test

void testPerformTaskCompletesWithinTime() {

PerformanceTester tester = new PerformanceTester();

assertTimeout(Duration.ofMillis(500), tester::performTask);

}

@Test

void testFailsIfTaskTakesTooLong() {

PerformanceTester tester = new PerformanceTester();

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

}

}

**Output:**

