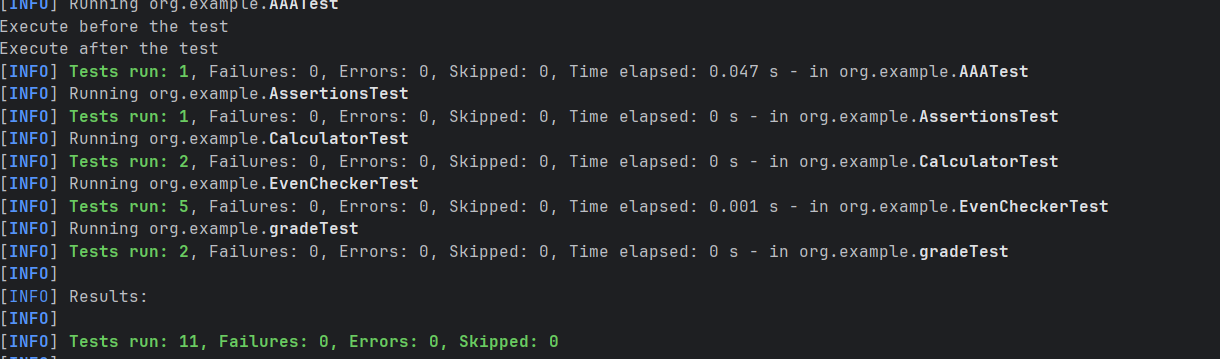
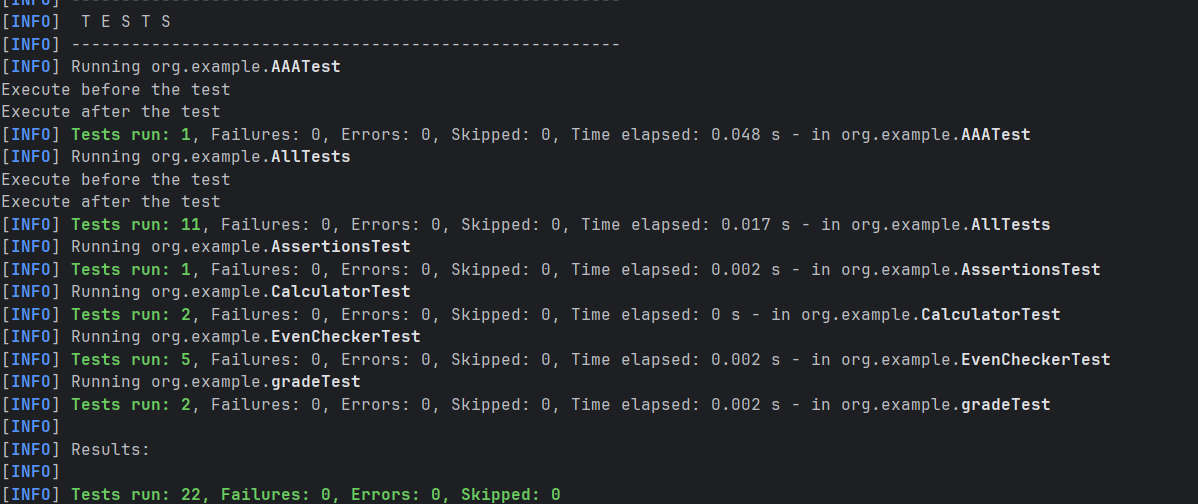
Exercise 1: Parameterized Tests

package org.example;  
  
import org.junit.Test;  
import org.junit.runner.RunWith;  
import org.junit.runners.Parameterized;  
  
import java.util.Arrays;  
import java.util.Collection;  
  
import static org.junit.Assert.*assertTrue*;  
import static org.junit.Assert.*assertFalse*;  
  
@RunWith(Parameterized.class)  
public class EvenCheckerTest {  
  
 private int input;  
 private boolean expected;  
  
 public EvenCheckerTest(int input, boolean expected) {  
 this.input = input;  
 this.expected = expected;  
 }  
  
 @Parameterized.Parameters  
 public static Collection<Object[]> testData() {  
 return Arrays.*asList*(new Object[][] {  
 {2, true},  
 {4, true},  
 {7, false},  
 {9, false},  
 {10, true}  
 });  
 }  
  
 @Test  
 public void testIsEven() {  
 EvenChecker checker = new EvenChecker();  
 if (expected) {  
 *assertTrue*(checker.isEven(input));  
 } else {  
 *assertFalse*(checker.isEven(input));  
 }  
 }  
}



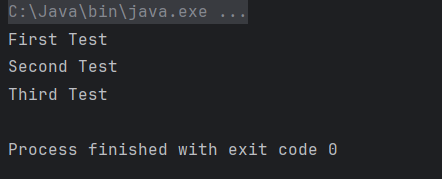
Exercise 2: Test Suites and Categories

package org.example;  
  
import org.junit.runner.RunWith;  
import org.junit.runners.Suite;  
  
@RunWith(Suite.class)  
@Suite.SuiteClasses({  
 EvenCheckerTest.class,  
 AAATest.class,  
 AssertionsTest.class,  
 CalculatorTest.class,  
 gradeTest.class  
})  
public class AllTests {  
  
}



Exercise 3: Test Execution Order

package org.example;  
  
import org.junit.FixMethodOrder;  
import org.junit.Test;  
import org.junit.runners.MethodSorters;  
  
import static org.junit.Assert.*assertTrue*;  
  
@FixMethodOrder(MethodSorters.*NAME\_ASCENDING*)  
public class OrderedTests {  
  
 @Test  
 public void testA\_First() {  
 System.*out*.println("First Test");  
 *assertTrue*(true);  
 }  
  
 @Test  
 public void testB\_Second() {  
 System.*out*.println("Second Test");  
 *assertTrue*(true);  
 }  
  
 @Test  
 public void testC\_Third() {  
 System.*out*.println("Third Test");  
 *assertTrue*(true);  
 }  
}



Exercise 4: Exception Testing

@class  
package org.example;

public class ExceptionThrower {

public void throwException(String message) {

if (message == null) {

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

}

}

}

@test

package org.example;

import org.junit.Test;

public class ExceptionThrowerTest {

@Test(expected = IllegalArgumentException.class)

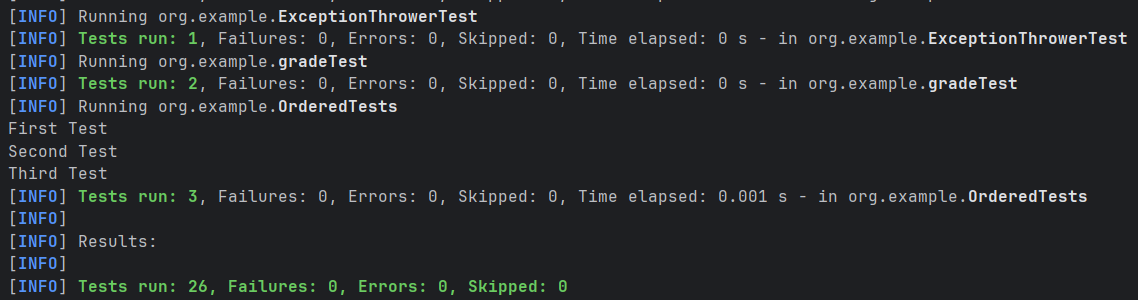
public void testThrowException() {

ExceptionThrower thrower = new ExceptionThrower();

thrower.throwException(null);

}

}



Exercise 5: Timeout and Performance Testing

@class

package org.example;  
  
public class PerformanceTester {  
  
 public void performTask() throws InterruptedException {  
  
 Thread.*sleep*(500);  
 }  
}

@test

package org.example;  
  
import org.junit.Test;  
  
public class PerformanceTesterTest {  
  
 @Test(timeout = 1000)   
 public void testPerformTaskWithinTimeout() throws InterruptedException {  
 PerformanceTester tester = new PerformanceTester();  
 tester.performTask();  
 }  
}

