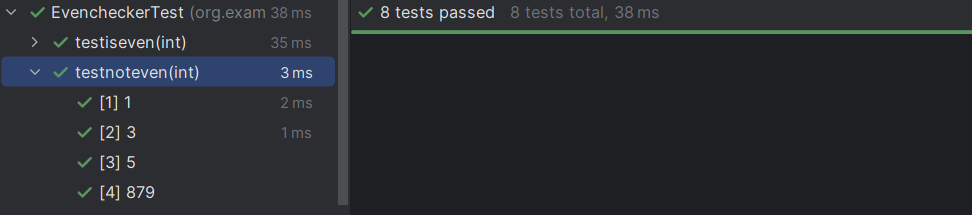
EXERCISE 1(Parameterized Tests)

package org.example;  
  
public class Evenchecker {  
 public boolean even(int n){  
 return n%2==0;  
 }  
}

TEST

package org.example;  
  
import org.junit.jupiter.params.ParameterizedTest;  
import org.junit.jupiter.params.provider.ValueSource;  
  
import static org.junit.jupiter.api.Assertions.\*;  
  
class EvencheckerTest {  
 Evenchecker even =new Evenchecker();  
 @ParameterizedTest  
 @ValueSource(ints ={ 2,4,6,8})  
 void testiseven(int n){  
 *assertTrue*(even.even(n));  
 }  
 @ParameterizedTest  
 @ValueSource(ints = {1,3,5,879})  
 void testnoteven(int n){  
 *assertFalse*(even.even(n));  
 }  
}

OUTPUT



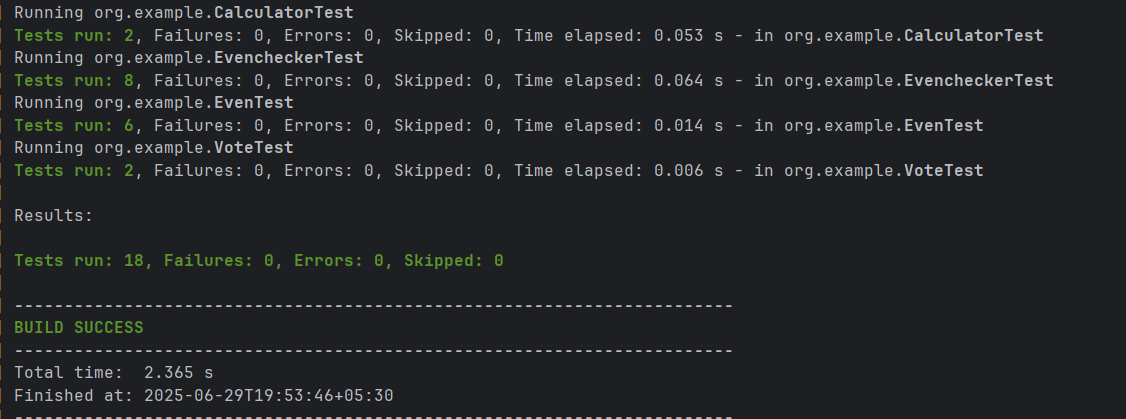
EXERCISE 2(Test Suites and Categories)

package org.example;  
  
import org.junit.platform.suite.api.SelectClasses;  
import org.junit.platform.suite.api.Suite;  
  
@Suite  
@SelectClasses({  
 EvenTest.class,  
 VoteTest.class,  
 CalculatorTest.class,  
 Evenchecker.class  
  
})  
public class AllTest {  
  
}

package org.example;  
  
import org.junit.jupiter.params.ParameterizedTest;  
import org.junit.jupiter.params.provider.ValueSource;  
  
import static org.junit.jupiter.api.Assertions.*assertTrue*;  
import static org.junit.jupiter.api.Assertions.*assertFalse*;  
  
public class EvenTest {  
  
 @ParameterizedTest  
 @ValueSource(ints = {2, 4, 6})  
 void testEven(int number) {  
 *assertTrue*(number % 2 == 0);  
 }  
  
 @ParameterizedTest  
 @ValueSource(ints = {1, 3, 5})  
 void testOdd(int number) {  
 *assertFalse*(number % 2 == 0);  
 }  
}

package org.example;  
  
import org.junit.jupiter.api.Test;  
  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class VoteTest {  
 @Test  
 public void testeligiblevoter(){  
 Vote vote=new Vote();  
 *assertEquals*("you are eligible for voting",vote.voter(19));  
 }  
 @Test  
 public void testnoteligiblevoter(){  
 Vote vote=new Vote();  
 *assertEquals*("you are not eligible for voting",vote.voter(9));  
 }  
}

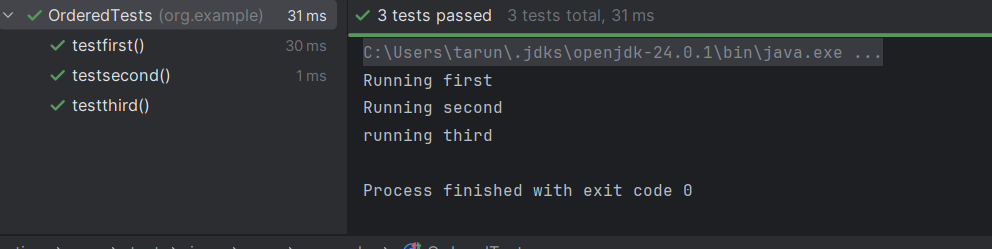
OUTPUT



EXERCISE 3(Test Execution Order)

package org.example;  
import org.junit.jupiter.api.\*;  
  
@TestMethodOrder(MethodOrderer.OrderAnnotation.class)  
public class OrderedTests {  
 @Test  
 @Order(1)  
 void testfirst(){  
 System.*out*.println("Running first");  
  
 }  
 @Test  
 @Order(3)  
 void testthird(){  
 System.*out*.println("running third");  
 }  
 @Test  
 @Order(2)  
 void testsecond(){  
 System.*out*.println("Running second");  
 }  
  
  
}

OUTPUT



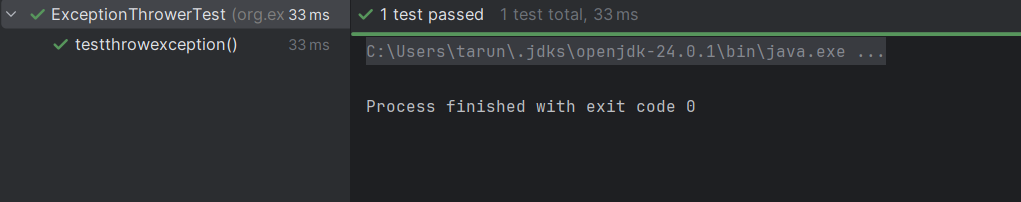
EXERCISE 4(Exception Testing)

package org.example;  
  
public class ExceptionThrower {  
 public void thorowException(boolean exceptionthrow){  
 if(exceptionthrow){  
 throw new IllegalArgumentException("Exception was thrown");  
 }  
 }  
}

TEST

package org.example;  
  
import org.junit.jupiter.api.Test;  
  
import static org.junit.jupiter.api.Assertions.\*;  
  
class ExceptionThrowerTest {  
 @Test  
 void testthrowexception(){  
 ExceptionThrower thrower=new ExceptionThrower();  
 *assertThrows*(IllegalArgumentException.class,()->{  
 thrower.thorowException(true);  
 });  
 }  
  
}

OUTPUT

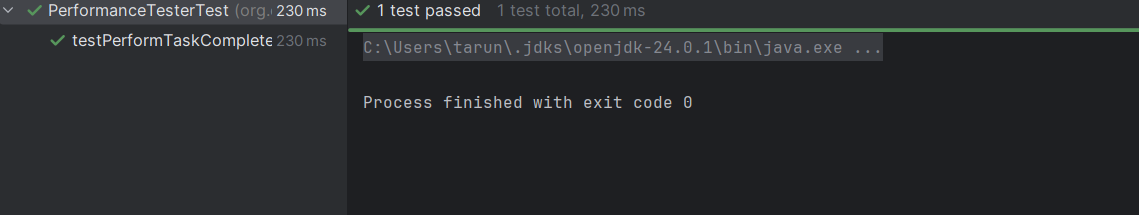


EXERCISE 5(Timeout and Performance Testing)

package org.example;  
  
public class PerformanceTester {  
 public void performTask() {  
 try {  
 Thread.*sleep*(200);  
 } catch (InterruptedException e) {  
 Thread.*currentThread*().interrupt();  
 }  
 }  
}

TEST

package org.example;  
import org.junit.jupiter.api.Test;  
import java.time.Duration;  
import static org.junit.jupiter.api.Assertions.*assertTimeout*;  
public class PerformanceTesterTest {  
 @Test  
 void testPerformTaskCompletesWithinTime() {  
 PerformanceTester tester = new PerformanceTester();  
 *assertTimeout*(Duration.*ofSeconds*(1), tester::performTask);  
 }  
}

OUTPUT