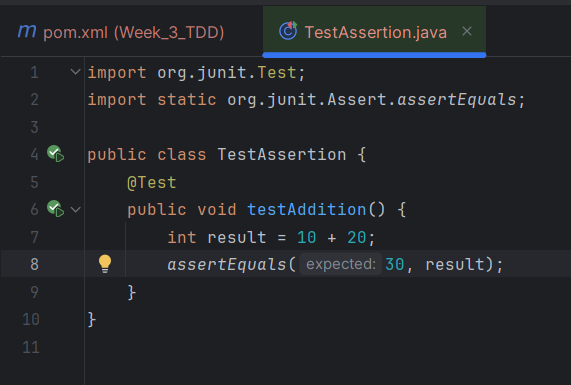
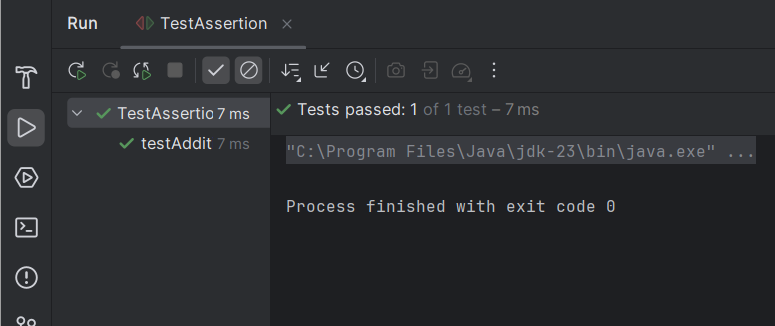
Exercise 1: Setting Up JUnit



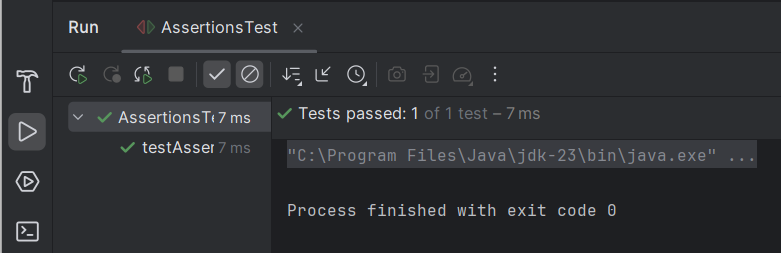
Output



Exercise 3: Assertions in JUnit



Output

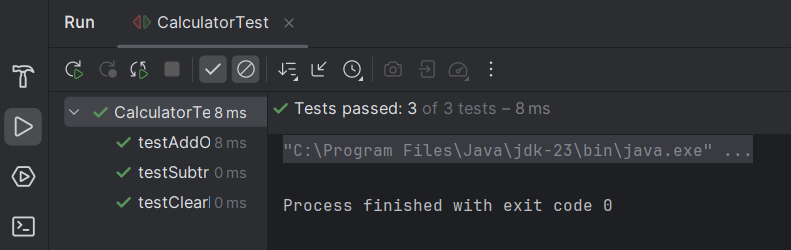


Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and

import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
import static org.junit.Assert.*assertEquals*;  
import static org.junit.Assert.*assertNotNull*;  
import static org.junit.Assert.*assertNull*;  
  
public class CalculatorTest {  
  
 private class Calculator {  
 private int result;  
  
 public Calculator() {  
 this.result = 0;  
 }  
  
 public int add(int a, int b) {  
 result = a + b;  
 return result;  
 }  
  
 public int subtract(int a, int b) {  
 result = a - b;  
 return result;  
 }  
  
 public int getResult() {  
 return result;  
 }  
  
 public void clear() {  
 result = 0;  
 }  
 }  
  
 private Calculator calculator;  
  
 @Before  
 public void setUp() {  
 calculator = new Calculator();  
 *assertNotNull*(calculator);  
 *assertEquals*(0, calculator.getResult());  
 }

@After  
 public void tearDown() {  
 calculator = null;  
 *assertNull*(calculator);  
 }  
  
 @Test  
 public void testAddOperation() {  
 int sum = calculator.add(5, 3);  
 *assertEquals*(8, sum);  
 *assertEquals*(8, calculator.getResult());  
 }  
  
 @Test  
 public void testSubtractOperation() {  
 int difference = calculator.subtract(10, 4);  
 *assertEquals*(6, difference);  
 *assertEquals*(6, calculator.getResult());  
 }  
  
 @Test  
 public void testClearMethod() {  
 calculator.add(100, 50);  
 *assertEquals*(150, calculator.getResult());  
 calculator.clear();  
 *assertEquals*(0, calculator.getResult());  
 }  
}

Output



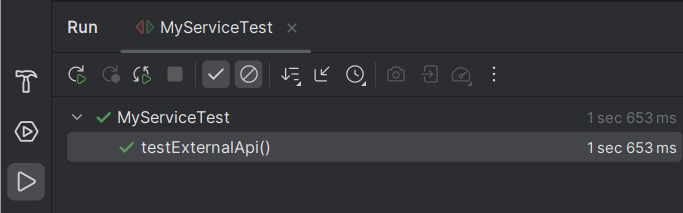
Exercise 1: Mocking and Stubbing

import static org.mockito.Mockito.\*;  
import static org.junit.jupiter.api.Assertions.\*;  
import org.junit.jupiter.api.Test;  
import org.mockito.Mockito;

public interface ExternalApi {  
 String getData();  
}

public class MyService {  
 private final ExternalApi api;  
  
 public MyService(ExternalApi api) {  
 this.api = api;  
 }  
  
 public String fetchData() {  
 return api.getData();  
 }  
}  
public class MyServiceTest {  
  
 @Test  
 public void testExternalApi() {  
   
 ExternalApi mockApi = Mockito.*mock*(ExternalApi.class);

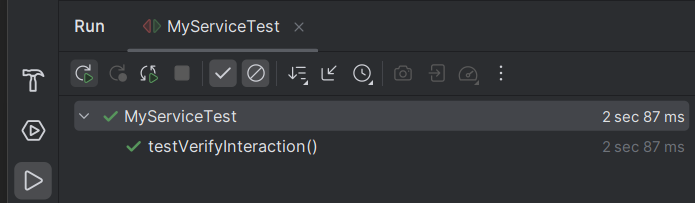
*when*(mockApi.getData()).thenReturn("Mock Data");  
  
 MyService service = new MyService(mockApi);  
  
 String result = service.fetchData();  
 *assertEquals*("Mock Data", result);  
 }  
}



Exercise 2: Verifying Interactions

import static org.mockito.Mockito.\*;  
import org.junit.jupiter.api.Test;  
  
public class MyServiceTest {  
  
 @Test  
 public void testVerifyInteraction() {  
 ExternalApi mockApi = *mock*(ExternalApi.class);  
  
 MyService service = new MyService(mockApi);  
 service.fetchData(); // Internally calls mockApi.getData()  
  
 *verify*(mockApi).getData();   
 }  
}

Output



Exercise 1: Logging Error Messages and Warning Levels

