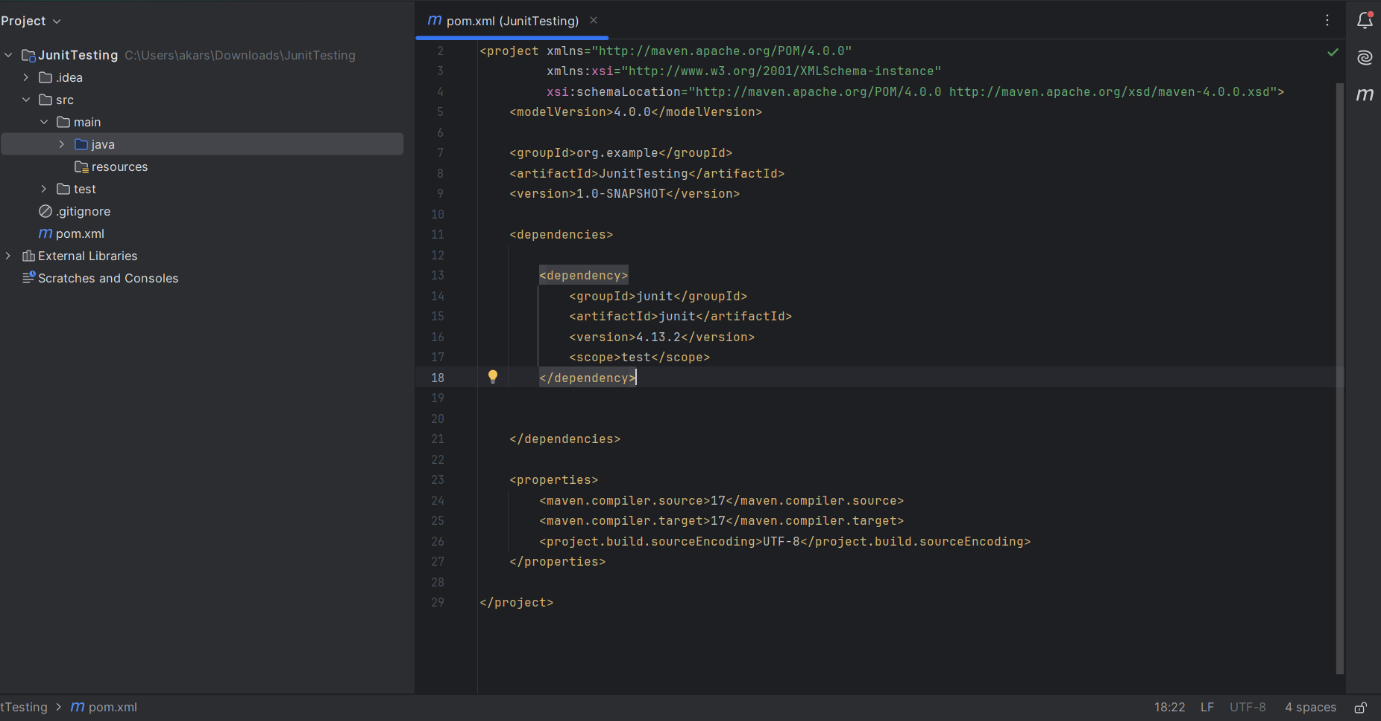
Exercise 1: Setting Up Junit

Code ->

<dependencies>  
 <dependency>  
 <groupId>junit</groupId>  
 <artifactId>junit</artifactId>  
 <version>4.13.2</version>  
 <scope>test</scope>  
 </dependency>  
</dependencies>

Output ->

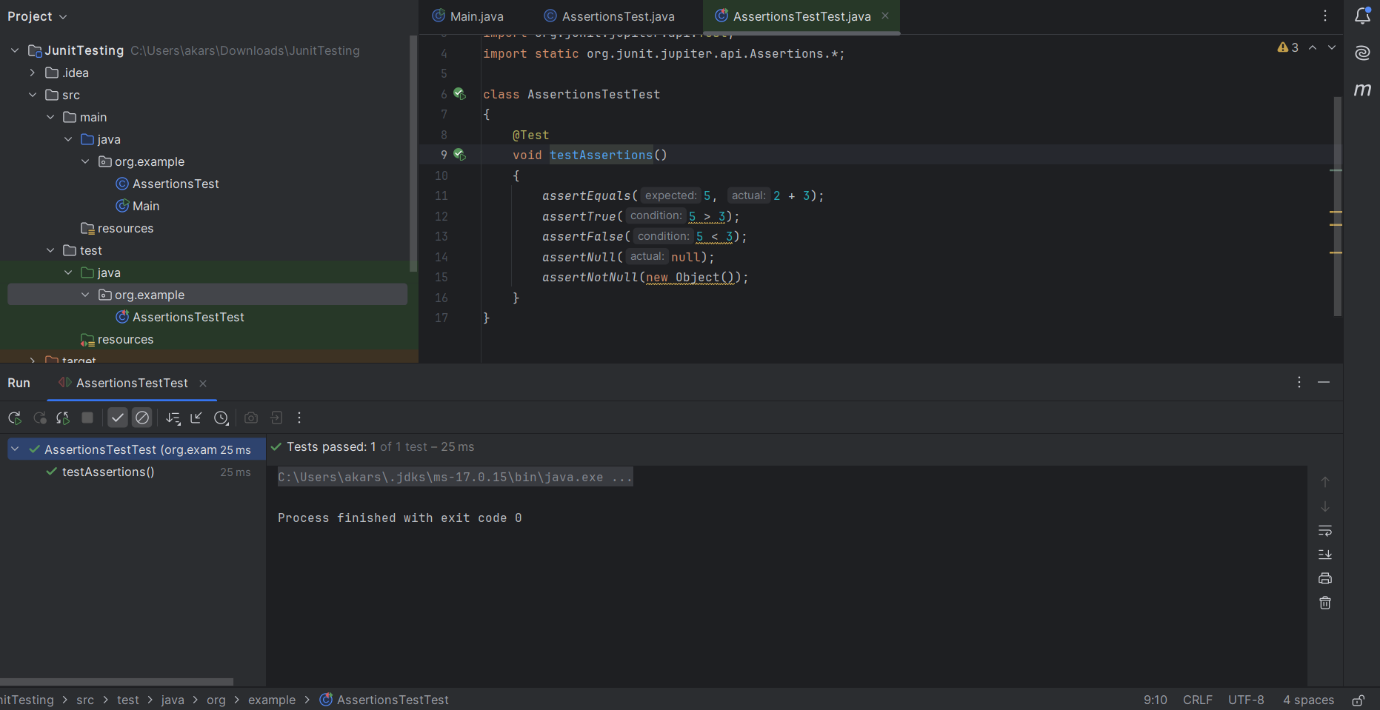


Exercise 3: Assertions in Junit

Code ->

package org.example;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class AssertionsTestTest  
{  
 @Test  
 void testAssertions()  
 {  
 *assertEquals*(5, 2 + 3);  
 *assertTrue*(5 > 3);  
 *assertFalse*(5 < 3);  
 *assertNull*(null);  
 *assertNotNull*(new Object());  
 }  
}

Output ->



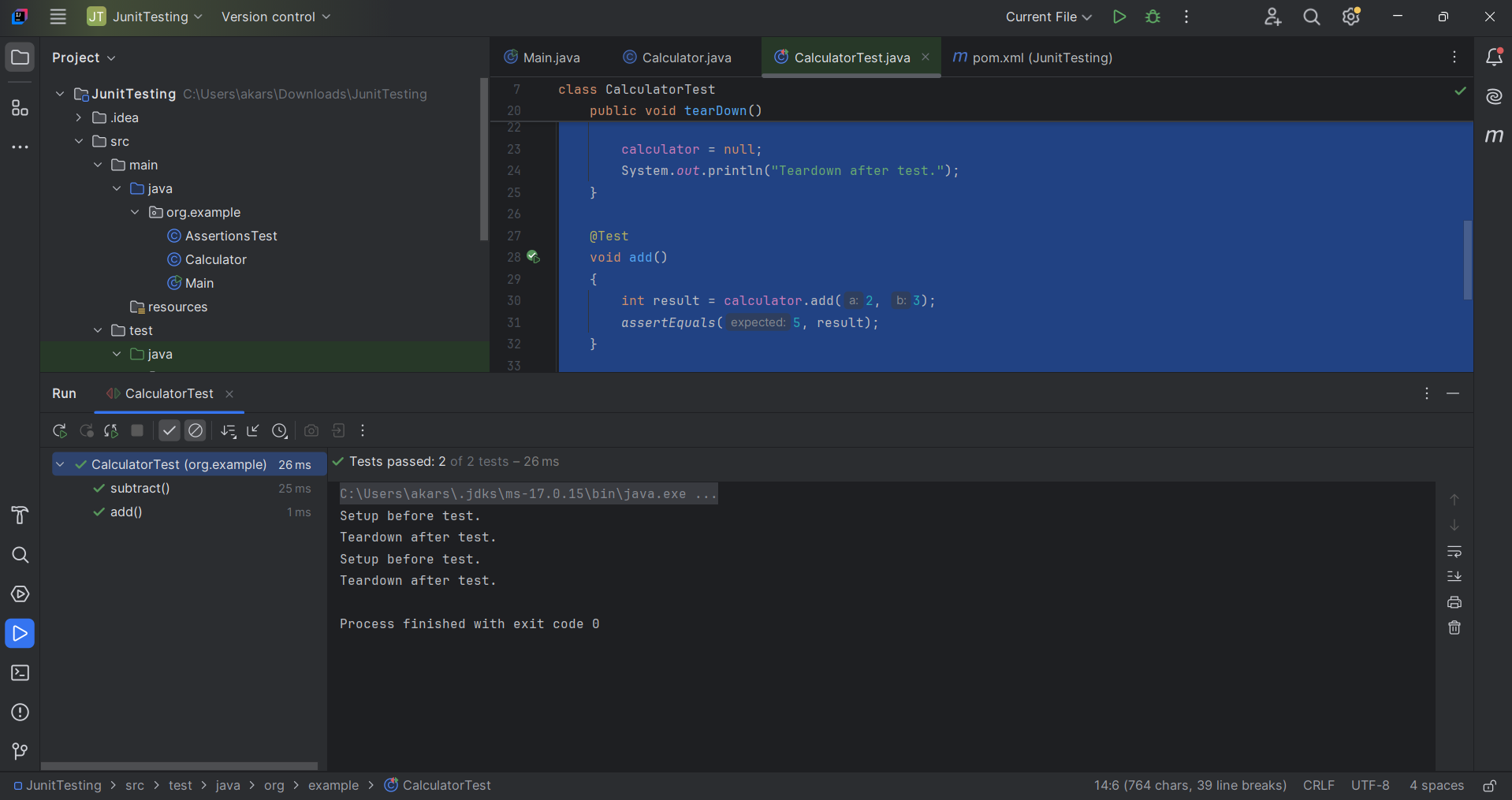
Exercise 4: Arrange-Act-Assert (AAA) Pattern, Test Fixtures, Setup and Teardown Methods in Junit

Code ->

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;  
 }  
}

package org.example;  
import org.junit.jupiter.api.AfterEach;  
import org.junit.jupiter.api.BeforeEach;  
import org.junit.jupiter.api.Test;  
import static org.junit.jupiter.api.Assertions.\*;  
  
class CalculatorTest  
{  
  
 private Calculator calculator;  
  
 @BeforeEach  
 public void setUp()  
 {  
 calculator = new Calculator();  
 System.*out*.println("Setup before test.");  
 }  
  
 @AfterEach  
 public void tearDown()  
 {  
  
 calculator = null;  
 System.*out*.println("Teardown after test.");  
 }  
  
 @Test  
 void add()  
 {  
 int result = calculator.add(2, 3);  
 *assertEquals*(5, result);  
 }  
  
 @Test  
 void subtract()  
 {  
 int result = calculator.subtract(10, 4);  
 *assertEquals*(6, result);  
 }  
}

Output ->



Exercise 1: Mocking and Stubbing

Code ->

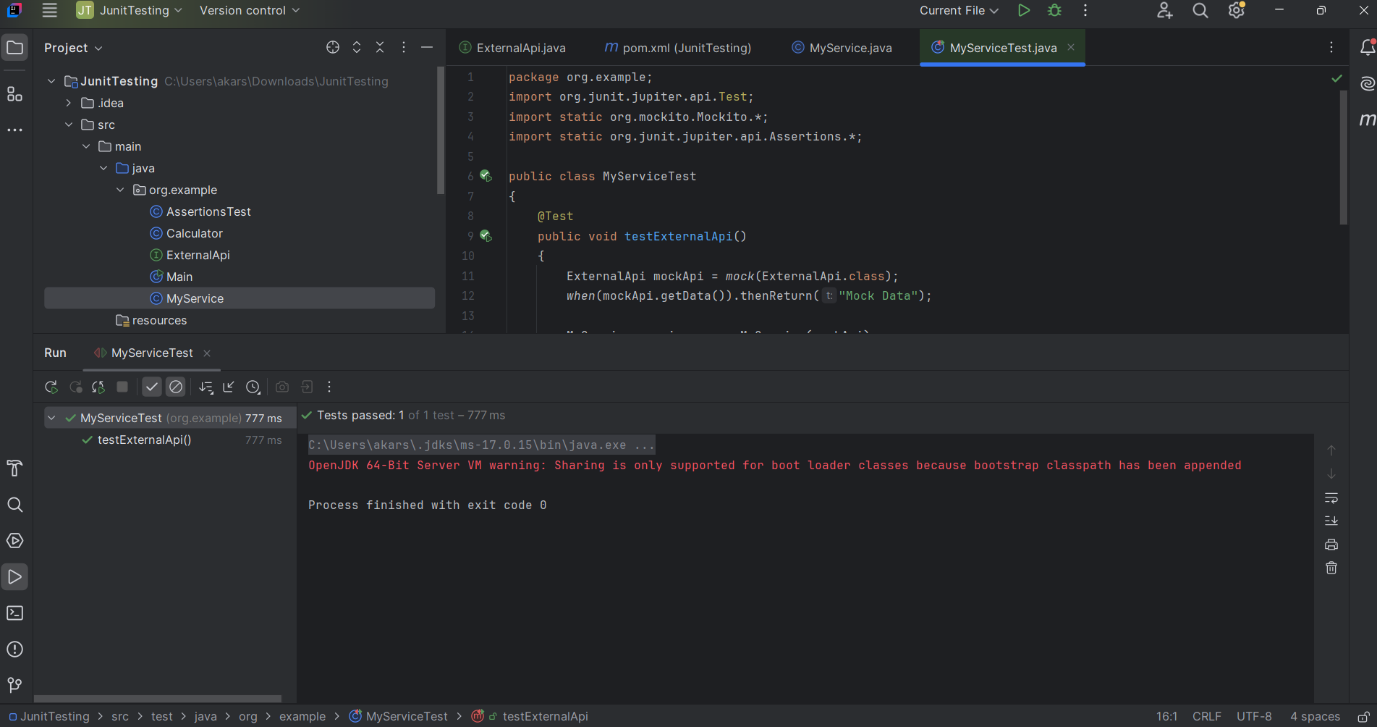
package org.example;

public interface ExternalApi  
{  
 String getData();  
}

package org.example;  
  
public class MyService  
{  
 private final ExternalApi api;  
  
 public MyService(ExternalApi api)  
 {  
 this.api = api;  
 }  
  
 public String fetchData()  
 {  
 return api.getData();  
 }  
}

package org.example;  
import org.junit.jupiter.api.Test;  
import static org.mockito.Mockito.\*;  
import static org.junit.jupiter.api.Assertions.\*;  
  
public class MyServiceTest  
{  
 @Test  
 public void testExternalApi()  
 {  
 ExternalApi mockApi = *mock*(ExternalApi.class);  
 *when*(mockApi.getData()).thenReturn("Mock Data");  
  
 MyService service = new MyService(mockApi);  
 String result = service.fetchData();  
  
 *assertEquals*("Mock Data", result);  
 }  
}

Output ->



Exercise 2: Verifying Interactions

Code ->

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

Output ->

