**Exercise 1: Setting Up JUnit**

* Pom.xml file
* <dependency>  
   <groupId>junit</groupId>  
   <artifactId>junit</artifactId>  
   <version>4.13.2</version>  
   <scope>test</scope>  
  </dependency>
* MainTest.java

package org.example

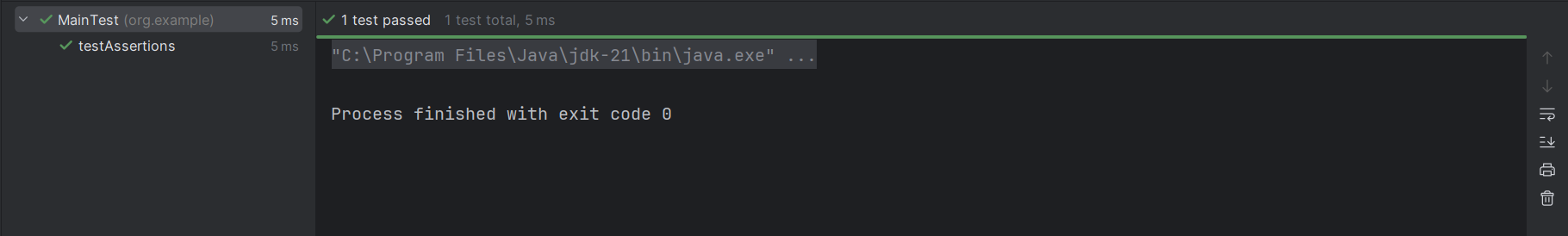
public class MainTest {

}

**Exercise 3: Assertions in JUnit**

public class AssertionsTest {  
 @Test  
 public 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**

* Main.java

package org.example;  
  
public class Main {  
  
 public int add(int n1, int n2) {  
 return n1 + n2;  
 }  
  
 public int subtract(int n1, int n2) {  
 return n1 - n2;  
 }  
  
 public static void main(String[] args) {  
  
 System.*out*.println("Hey");  
  
 }  
}

* MainTest.java

package org.example;  
  
import org.junit.After;  
import org.junit.Before;  
import org.junit.Test;  
  
import static org.junit.Assert.\*;  
  
public class MainTest {  
  
 private Main calculator;  
  
 @Before  
 public void setUp() {   
 calculator = new Main();  
 System.*out*.println("Setup complete");  
 }  
  
 @After  
 public void tearDown() {   
 calculator = null;  
 System.*out*.println("Teardown complete");  
 }  
  
 @Test  
 public void testAddition() {   
 int a = 5;  
 int b = 3;  
  
 int result = calculator.add(a, b);  
  
 *assertEquals*(8, result);  
 }  
  
 @Test  
 public void testSubtraction() {   
 int a = 10;  
 int b = 4;  
  
 int result = calculator.subtract(a, b);  
  
 *assertEquals*(6, result);  
 }  
  
  
}

* **Output**

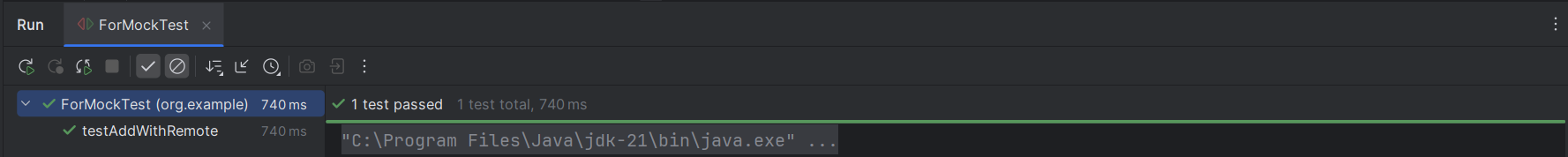
****

**Exercise 1: Mocking and Stubbing**

* **ForMock.java**
* package org.example;  
    
  public class ForMock {  
    
   private final ExternalAPI api;  
    
   public ForMock(ExternalAPI api) {  
   this.api = api;  
   }  
    
   public int add(int n1, int n2) {  
   return n1 + n2;  
   }  
    
   public int addWithRemote(int n1) {  
   int remote = api.fetchRemoteValue();   
   return n1 + remote;  
   }  
    
   public int subtract(int n1, int n2) {  
   return n1 - n2;  
   }  
    
   public static void main(String[] args) {  
   System.*out*.println("Hey");  
   }  
  }
* **ForMockTest.java**

package org.example;  
  
import org.junit.Test;  
import org.junit.Before;  
import static org.junit.Assert.*assertEquals*;  
import static org.mockito.Mockito.\*;  
  
public class ForMockTest {  
  
 private ExternalAPI apiMock;  
 private ForMock main;  
  
 @Before  
 public void setUp() {  
 apiMock = *mock*(ExternalAPI.class);   
 *when*(apiMock.fetchRemoteValue()).thenReturn(5);   
 main = new ForMock(apiMock);  
 }  
  
 @Test  
 public void testAddWithRemote() {   
 int result = main.addWithRemote(10);  
 *assertEquals*(15, result); // 10 + stubbed 5  
 }  
}

* **Output**



**Exercise 2: Verifying Interactions**

* ExternalApi2.java

package org.example;  
  
public interface ExternalApi2 {  
 String getData();  
}

* MyService.java

package org.example;  
  
public class MyService {  
 private final ExternalApi2 api;  
  
 public MyService(ExternalApi2 api) {  
 this.api = api;  
 }  
  
 public void fetchData() {  
 api.getData();  
 }  
}

**Output**

