**Exercise 1: Setting Up JUnit**

import org.junit.Test;import static org.junit.Assert.assertEquals;public class CalculatorTest { public void testAdd() { Calculator calc = new Calculator(); int result = calc.add(3, 4); assertEquals(7, result); }}

**Exercise 3: Assertions in J**

import org.junit.Test;

import static org.junit.Assert.\*;

public class AssertionsTest {

public void testAssertions() {

assertEquals(5, 2 + 3);

assertTrue(5 > 3);

assertFalse(5 < 3);

assertNull(null);

assertNotNull(new Object());

}

}

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

**Teardown Methods in JUnit**

import org.junit.After;

import org.junit.Before;

import org.junit.Test;

import static org.junit.Assert.\*;

public class AAAPatternTest {

private Calculator calculator;

public void setUp() {

calculator = new Calculator();

System.out.println("Setup complete.");

} public void testAdd() {

int result = calculator.add(10, 5);

assertEquals(15, result);

}

public void tearDown() {

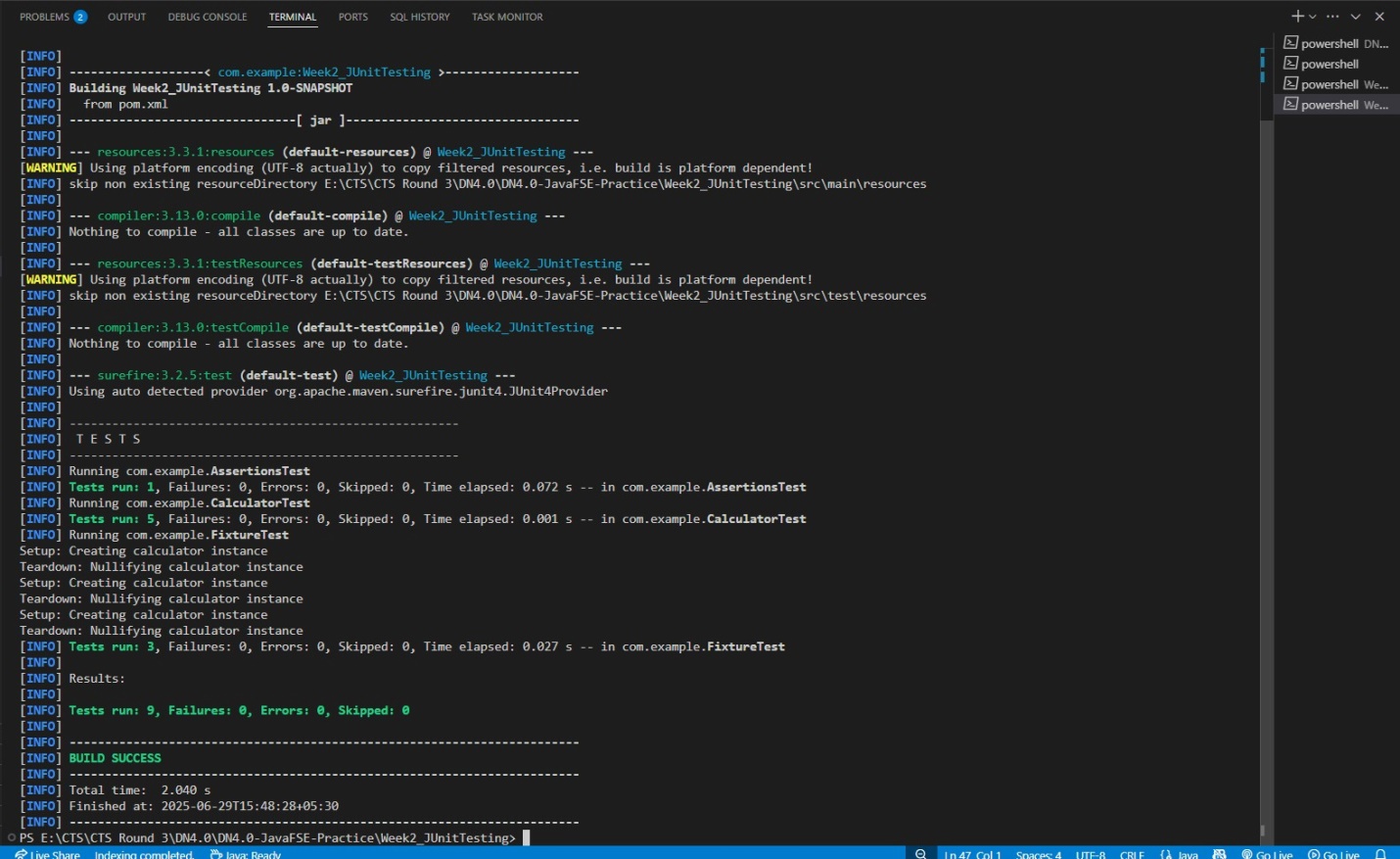
calculator = null;

System.out.println("Teardown complete.");

}

}

**OUTPUT**



**Mockito exercises**

**Exercise 1: Mocking and Stubbi**

**1.ExternalApi.java**

public interface ExternalApi {

String getData();

}

**2.MyService.java**

public class MyService {

private ExternalApi api;

public MyService(ExternalApi api) {

this.api = api;

}

public String fetchData() {

return api.getData();

}

}

**3.MyServiceTest.java (with Mockito)**

import static org.mockito.Mockito.\*;

import static org.junit.jupiter.api.Assertions.assertEquals;

import org.junit.jupiter.api.Test;

import org.mockito.Mockito;

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);

}

}

**4.Updated pom.xml to include Mockito and JUnit 5**

<dependencies>

<!-- JUnit 5 -->

<dependency>

<groupId>org.junit.jupiter</groupId>

<artifactId>junit-jupiter</artifactId>

<version>5.7.0</version>

<scope>test</scope>

</dependency>

<!-- Mockito -->

<dependency>

<groupId>org.mockito</groupId>

<artifactId>mockito-core</artifactId>

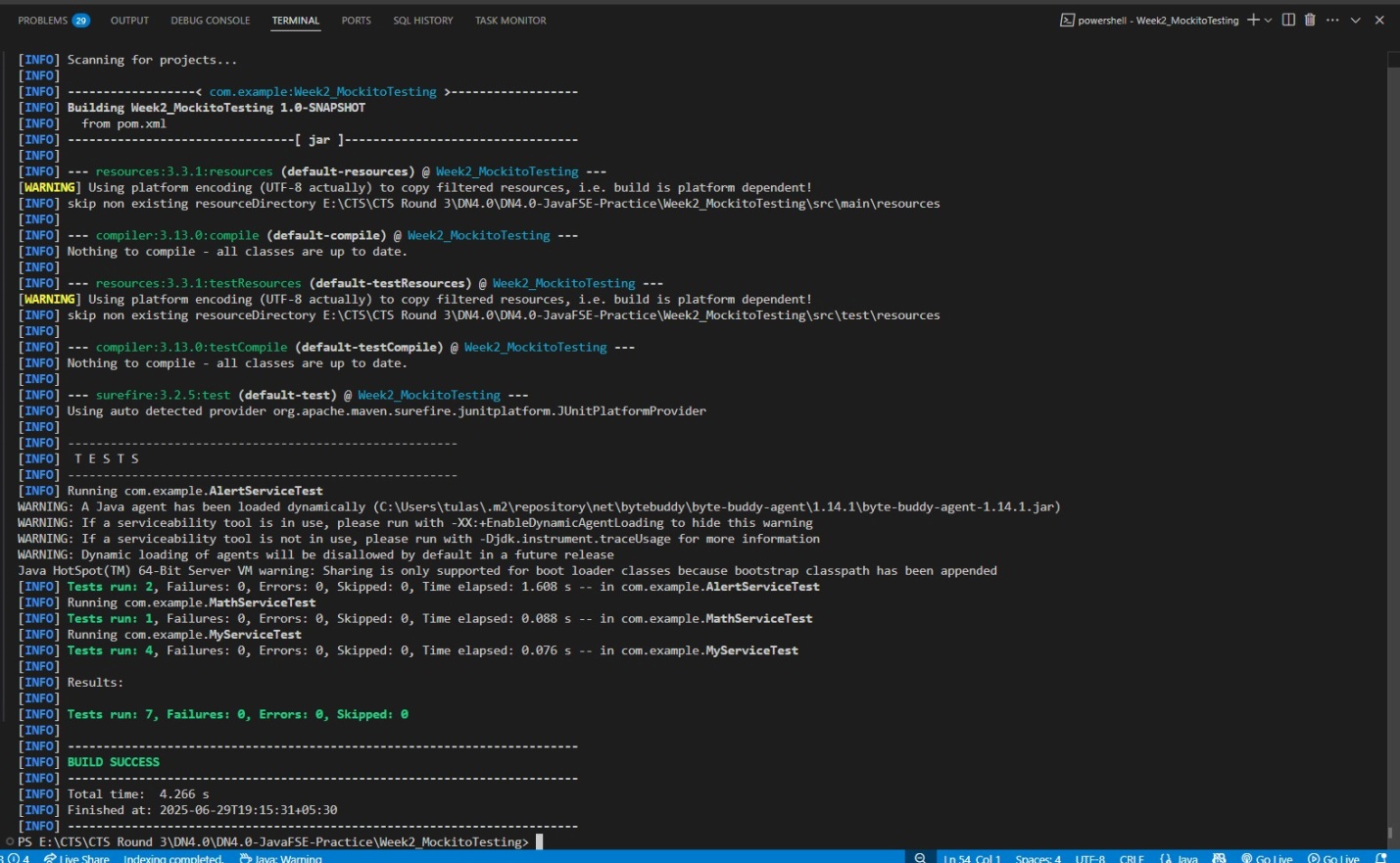
<version>3.9.0</version>

<scope>test</scope>

</dependency>

</dependencies>

**OUTPUT**



**Exercise 2: Verifying Interactions**

**ExternalApi.java**

public interface ExternalApi { void getData();}

**MyService.java**

public class MyService { private ExternalApi api; public MyService(ExternalApi api) { this.api = api; } public void fetchData() { api.getData(); }}

**MyServiceTest.java**

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

**OUTPUT**

