JUnit Basic Testing

Exercise 1: Setting Up JUnit

Scenario:

You need to set up JUnit in your Java project to start writing unit tests.

Steps:

1. Create a new Java project in your IDE (e.g., IntelliJ IDEA, Eclipse).

2. Add JUnit dependency to your project. If you are using Maven, add the following to your

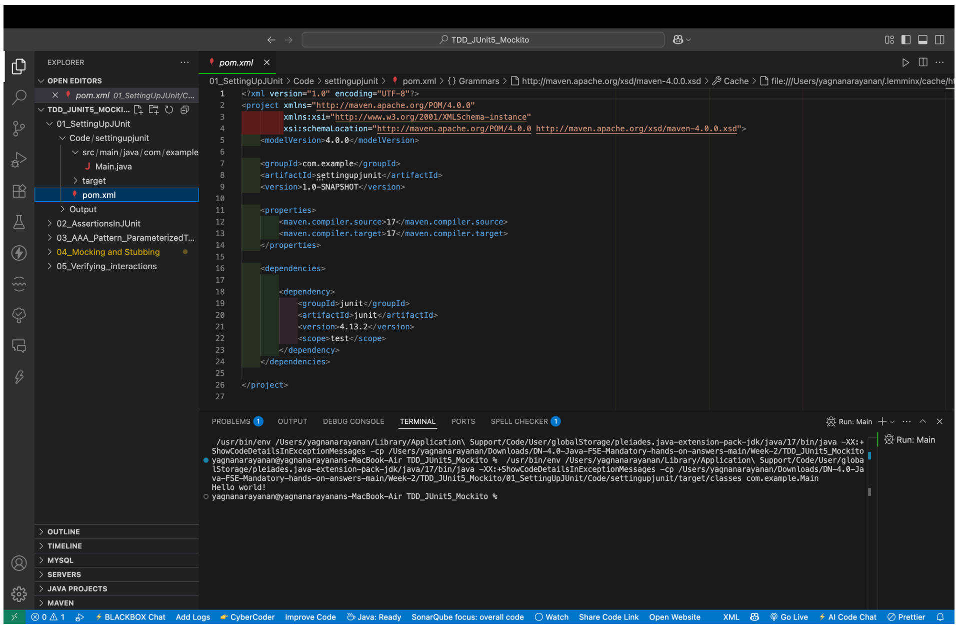
3. Create a new test class in your project

Code:

Main.java

```java
package com.example;

public class Main {
    public static void main(String[] args) {
        System.out.println("Hello world!");
    }
}
```

Output:



```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>com.example</groupId>
    <artifactId>settingupjunit</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>17</maven.compiler.source>
        <maven.compiler.target>17</maven.compiler.target>
    </properties>

    <dependencies>

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

</project>
```

```
 /usr/bin/env /Users/yagnanarayanan/Library/Application\ Support/Code/User/globalStorage/pleiades.java-extension-pack-jdk/java/17/bin/java -XX:+
ShowCodeDetailsInExceptionMessages -cp /Users/yagnanarayanan/Downloads/DN-4.0-Java-FSE-Mandatory-hands-on-answers-main/Week-2/TDD_JUnit5_Mockito
yagnanarayanan@yagnanarayanans-MacBook-Air TDD_JUnit5_Mockito %  /usr/bin/env /Users/yagnanarayanan/Library/Application\ Support/Code/User/globa
lStorage/pleiades.java-extension-pack-jdk/java/17/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/yagnanarayanan/Downloads/DN-4.0-Ja
va-FSE-Mandatory-hands-on-answers-main/Week-2/TDD_JUnit5_Mockito/01_SettingUpJUnit/Code/settingupjunit/target/classes com.example.Main
Hello world!
yagnanarayanan@yagnanarayanans-MacBook-Air TDD_JUnit5_Mockito %
```

Exercise 3: Assertions in JUnit

Scenario:

You need to use different assertions in JUnit to validate your test results.

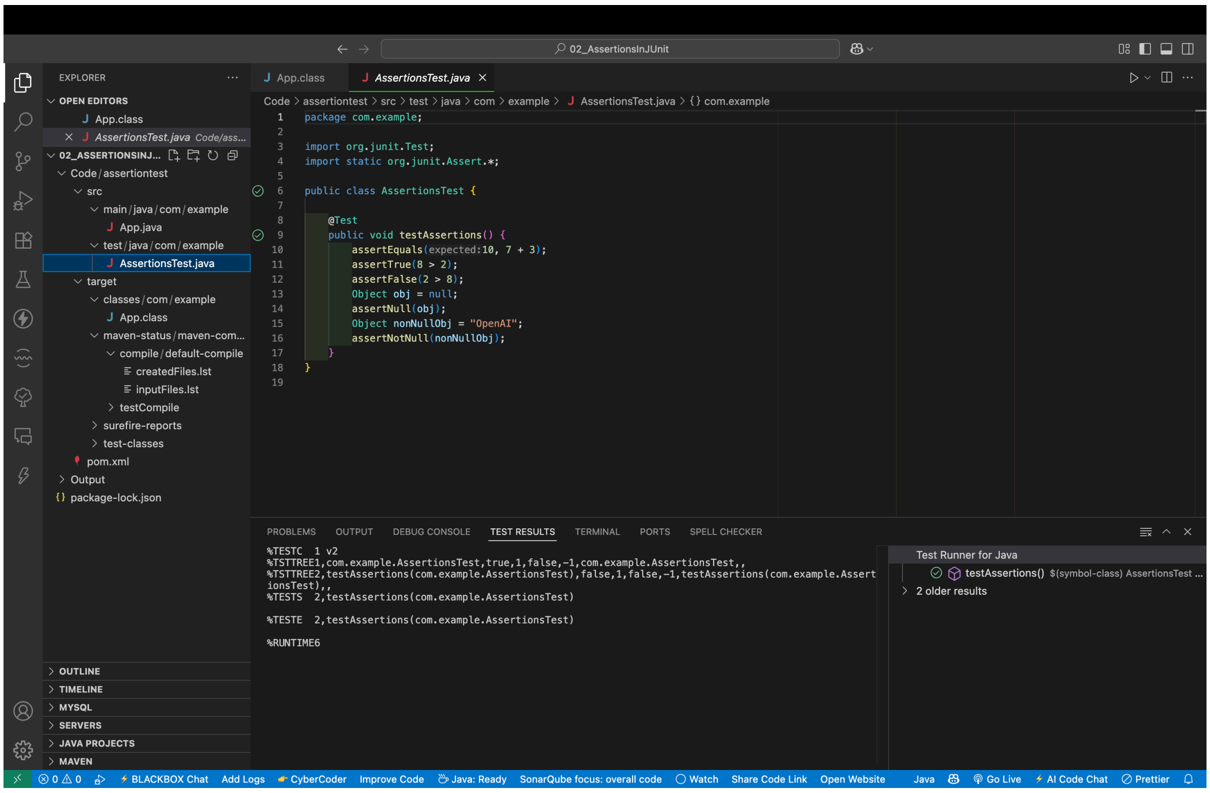
Steps: 1. Write tests using various JUnit assertions.

Code:

AssertionsTest.java

```java
package com.example;

import org.junit.Test;
import static org.junit.Assert.*;

public class AssertionsTest {

    @Test
    public void testAssertions() {
        assertEquals(10, 7 + 3);
        assertTrue(8 > 2);
        assertFalse(2 > 8);
        Object obj = null;
        assertNull(obj);
        Object nonNullObj = "OpenAI";
        assertNotNull(nonNullObj);
    }
}
```

Output:



```java
package com.example;

import org.junit.Test;
import static org.junit.Assert.*;

public class AssertionsTest {

    @Test
    public void testAssertions() {
        assertEquals(expected:10, 7 + 3);
        assertTrue(8 > 2);
        assertFalse(2 > 8);
        Object obj = null;
        assertNull(obj);
        Object nonNullObj = "OpenAI";
        assertNotNull(nonNullObj);
    }
}
```

```
%TESTC  1 v2
%TSTTREE1,com.example.AssertionsTest,true,1,false,-1,com.example.AssertionsTest,,
%TSTTREE2,testAssertions(com.example.AssertionsTest),false,1,false,-1,testAssertions(com.example.Assert
ionsTest),,
%TESTS  2,testAssertions(com.example.AssertionsTest)

%TESTE  2,testAssertions(com.example.AssertionsTest)

%RUNTIME6
```

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

Scenario:

You need to organize your tests using the Arrange-Act-Assert (AAA) pattern and use setup and teardown methods.

Steps:

1. Write tests using the AAA pattern.

2. Use @Before and @After annotations for setup and teardown methods.

Code:

Calculator.java

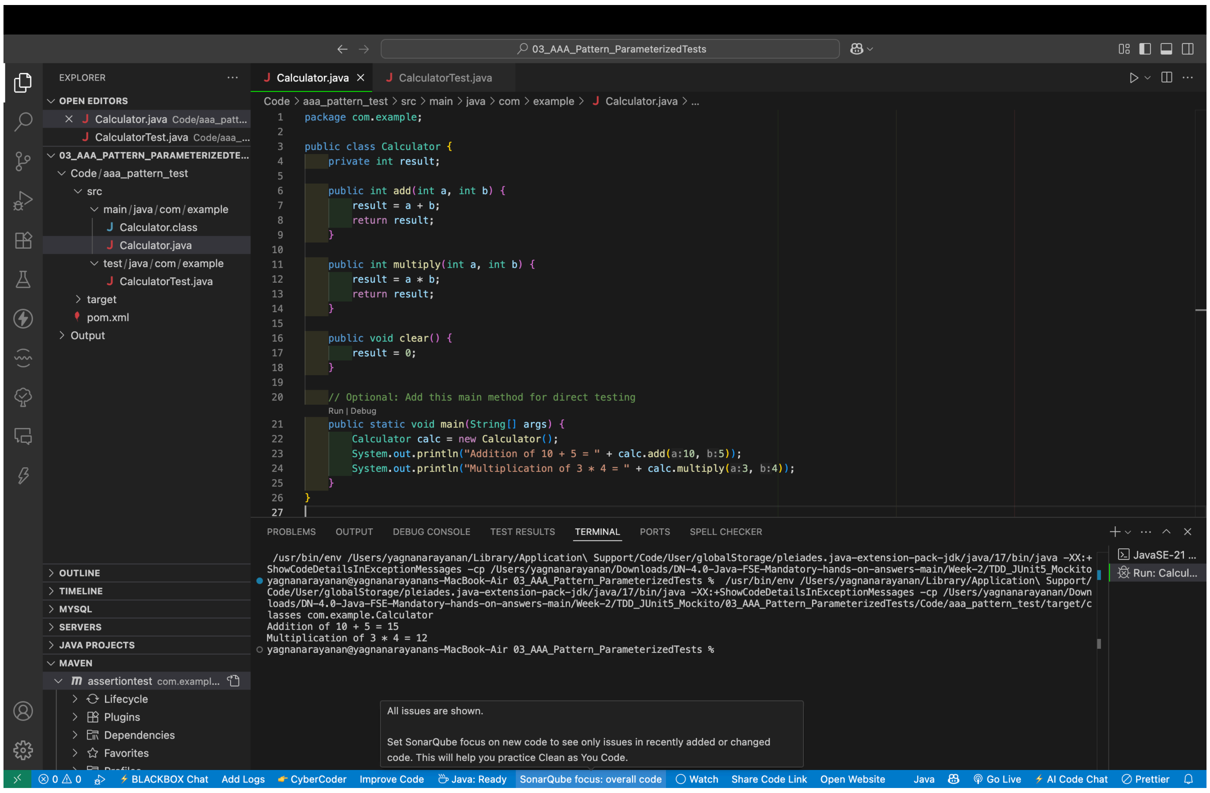
```java
package com.example;

public class Calculator {
    private int result;

    public int add(int a, int b) {
        result = a + b;
        return result;
    }

    public int multiply(int a, int b) {
        result = a * b;
        return result;
    }

    public void clear() {
        result = 0;
    }

    // Optional: Add this main method for direct testing
    public static void main(String[] args) {
        Calculator calc = new Calculator();
        System.out.println("Addition of 10 + 5 = " +
calc.add(10, 5));
        System.out.println("Multiplication of 3 * 4 = " +
calc.multiply(3, 4));
```

```
    }
}
```

CalculatorTest.java

```java
package com.example;

import org.junit.Before;
import org.junit.After;
import org.junit.Test;
import static org.junit.Assert.*;

public class CalculatorTest {

    private Calculator calculator;

    @Before
    public void setUp() {
        calculator = new Calculator();
        System.out.println("Setup complete.");
    }

    @After
    public void tearDown() {
        calculator.clear();
        System.out.println("Teardown complete.");
    }

    @Test
    public void testAdd() {
        int a = 12;
        int b = 8;
        int result = calculator.add(a, b);
        assertEquals(20, result);
    }

    @Test
    public void testMultiply() {
        int a = 7;
        int b = 4;
        int result = calculator.multiply(a, b);
        assertEquals(28, result);
    }
}
```

Output:



```java
package com.example;

public class Calculator {
    private int result;

    public int add(int a, int b) {
        result = a + b;
        return result;
    }

    public int multiply(int a, int b) {
        result = a * b;
        return result;
    }

    public void clear() {
        result = 0;
    }

    // Optional: Add this main method for direct testing
    public static void main(String[] args) {
        Calculator calc = new Calculator();
        System.out.println("Addition of 10 + 5 = " + calc.add(a:10, b:5));
        System.out.println("Multiplication of 3 * 4 = " + calc.multiply(a:3, b:4));
    }
}
```

```
 /usr/bin/env /Users/yagnanarayanan/Library/Application\ Support/Code/User/globalStorage/pleiades.java-extension-pack-jdk/java/17/bin/java -XX:+
ShowCodeDetailsInExceptionMessages -cp /Users/yagnanarayanan/Downloads/DN-4.0-Java-FSE-Mandatory-hands-on-answers-main/Week-2/TDD_JUnit5_Mockito
yagnanarayanan@yagnanarayanans-MacBook-Air 03_AAA_Pattern_ParameterizedTests %  /usr/bin/env /Users/yagnanarayanan/Library/Application\ Support/
Code/User/globalStorage/pleiades.java-extension-pack-jdk/java/17/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/yagnanarayanan/Down
loads/DN-4.0-Java-FSE-Mandatory-hands-on-answers-main/Week-2/TDD_JUnit5_Mockito/03_AAA_Pattern_ParameterizedTests/Code/aaa_pattern_test/target/c
lasses com.example.Calculator
Addition of 10 + 5 = 15
Multiplication of 3 * 4 = 12
yagnanarayanan@yagnanarayanans-MacBook-Air 03_AAA_Pattern_ParameterizedTests %
```

All issues are shown.

Set SonarQube focus on new code to see only issues in recently added or changed code. This will help you practice Clean as You Code.

Mockito Hands-On Exercises

Exercise 1: Mocking and Stubbing

Scenario:

You need to test a service that depends on an external API. Use Mockito to mock the external API and stub its methods.

Steps:

1. Create a mock object for the external API.

2. Stub the methods to return predefined values.

3. Write a test case that uses the mock object.

Code:

ExternalApi.java

```java
package com.example;

public interface ExternalApi {
    String getData();
    void sendData(String data);
}
```

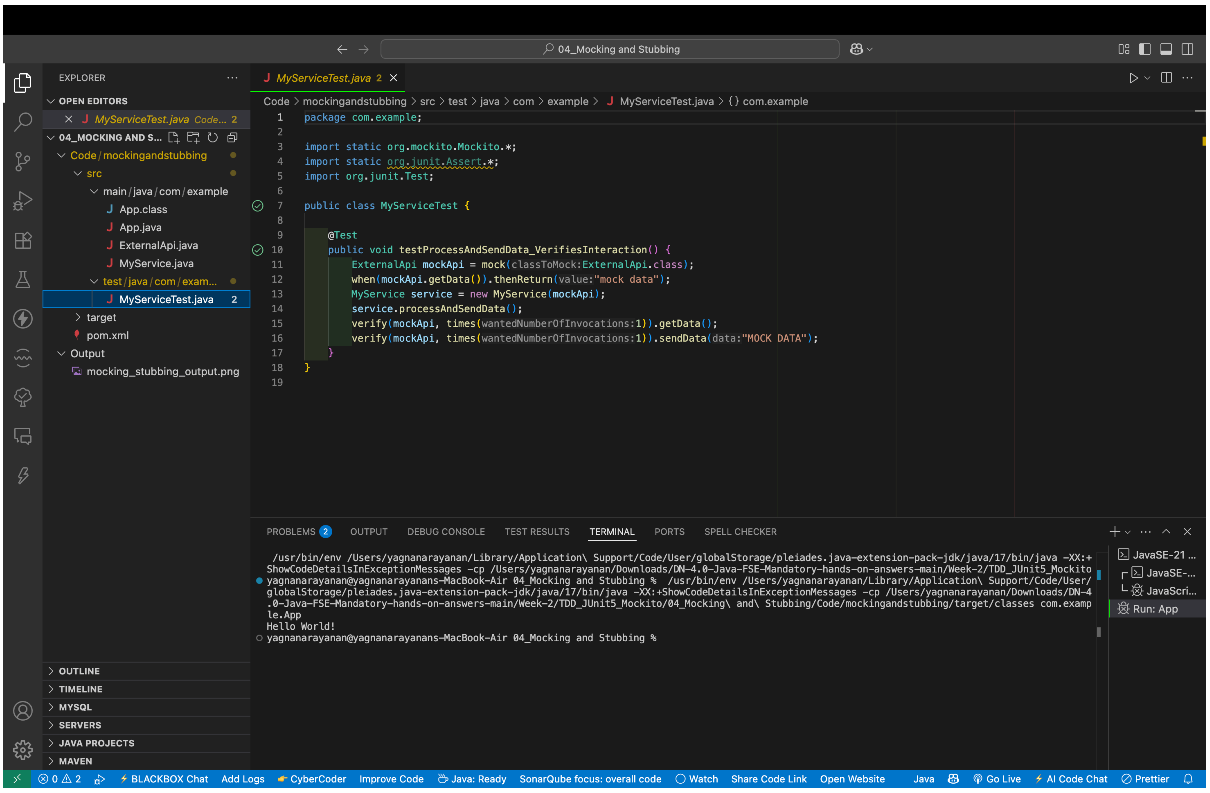
MyService.java

```java
package com.example;

public class MyService {

    private ExternalApi api;

    public MyService(ExternalApi api) {
        this.api = api;
    }

    public void processAndSendData() {
        String data = api.getData();
        String processed = data.toUpperCase();
        api.sendData(processed);
    }
}
```

MyServiceTest.java

```java
package com.example;

import static org.mockito.Mockito.*;
import static org.junit.Assert.*;
import org.junit.Test;

public class MyServiceTest {

    @Test
    public void testProcessAndSendData_VerifiesInteraction() {
        ExternalApi mockApi = mock(ExternalApi.class);
        when(mockApi.getData()).thenReturn("mock data");
        MyService service = new MyService(mockApi);
        service.processAndSendData();
        verify(mockApi, times(1)).getData();
        verify(mockApi, times(1)).sendData("MOCK DATA");
    }
}
```

Output:



```java
package com.example;

import static org.mockito.Mockito.*;
import static org.junit.Assert.*;
import org.junit.Test;

public class MyServiceTest {

    @Test
    public void testProcessAndSendData_VerifiesInteraction() {
        ExternalApi mockApi = mock(classToMock:ExternalApi.class);
        when(mockApi.getData()).thenReturn(value:"mock data");
        MyService service = new MyService(mockApi);
        service.processAndSendData();
        verify(mockApi, times(wantedNumberOfInvocations:1)).getData();
        verify(mockApi, times(wantedNumberOfInvocations:1)).sendData(data:"MOCK DATA");
    }
}
```

```
 /usr/bin/env /Users/yagnanarayanan/Library/Application\ Support/Code/User/globalStorage/pleiades.java-extension-pack-jdk/java/17/bin/java -XX:+
ShowCodeDetailsInExceptionMessages -cp /Users/yagnanarayanan/Downloads/DN-4.0-Java-FSE-Mandatory-hands-on-answers-main/Week-2/TDD_JUnit5_Mockito
yagnanarayanan@yagnanarayanans-MacBook-Air 04_Mocking and Stubbing %  /usr/bin/env /Users/yagnanarayanan/Library/Application\ Support/Code/User/
globalStorage/pleiades.java-extension-pack-jdk/java/17/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/yagnanarayanan/Downloads/DN-4
.0-Java-FSE-Mandatory-hands-on-answers-main/Week-2/TDD_JUnit5_Mockito/04_Mocking\ and\ Stubbing/Code/mockingandstubbing/target/classes com.examp
le.App
Hello World!
yagnanarayanan@yagnanarayanans-MacBook-Air 04_Mocking and Stubbing %
```

Exercise 2: Verifying Interactions

Scenario:

You need to ensure that a method is called with specific arguments.

Steps:

1. Create a mock object.

2. Call the method with specific arguments.

3. Verify the interaction.

Code:

ExternalApi.java

```java
package com.example;

public interface ExternalApi {
    String getData();
    void sendData(String data);
}
```

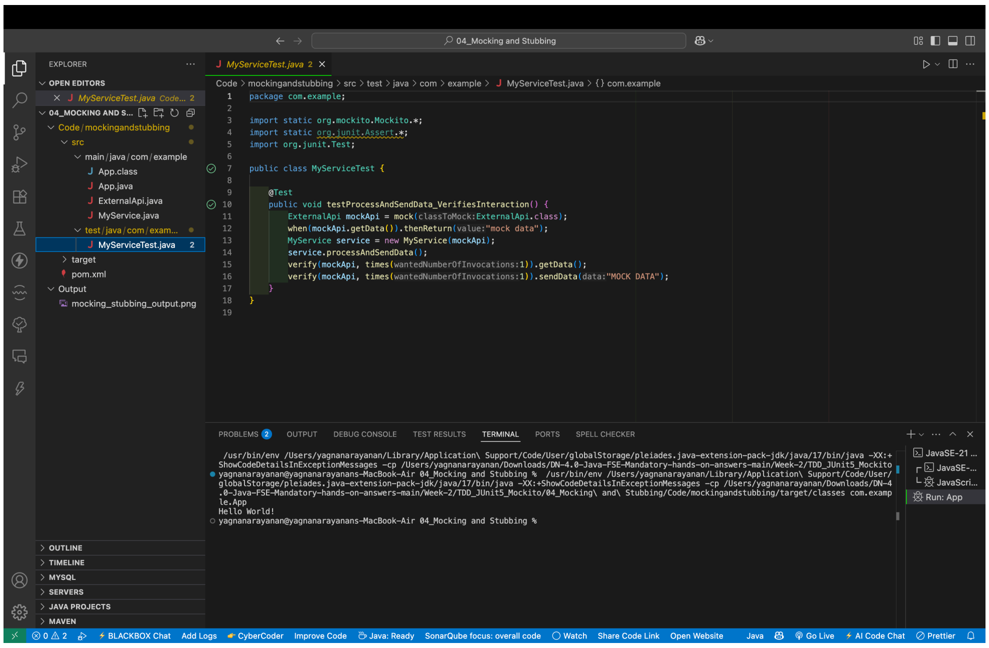
MyService.java

```java
package com.example;

public class MyService {
    private ExternalApi externalApi;

    public MyService(ExternalApi externalApi) {
        this.externalApi = externalApi;
    }

    public void sendProcessedData(String data) {
        String processed = data.toUpperCase();
        externalApi.sendData(processed);
    }
}
```

MyServiceTest.java

```java
package com.example;

import static org.mockito.Mockito.*;
import org.junit.Test;

public class MyServiceTest {

    @Test
    public void testVerifyInteractionWithSpecificArguments() {
        ExternalApi mockApi = mock(ExternalApi.class);
        MyService service = new MyService(mockApi);

        service.sendProcessedData("hello world");

        verify(mockApi).sendData("HELLO WORLD");
    }
}
```

Output:



```java
package com.example;

import static org.mockito.Mockito.*;
import static org.junit.Assert.*;
import org.junit.Test;

public class MyServiceTest {

    @Test
    public void testProcessAndSendData_VerifiesInteraction() {
        ExternalApi mockApi = mock(classToMock:ExternalApi.class);
        when(mockApi.getData()).thenReturn(value:"mock data");
        MyService service = new MyService(mockApi);
        service.processAndSendData();
        verify(mockApi, times(wantedNumberOfInvocations:1)).getData();
        verify(mockApi, times(wantedNumberOfInvocations:1)).sendData(data:"MOCK DATA");
    }
}
```

```
 /usr/bin/env /Users/yagnanarayanan/Library/Application\ Support/Code/User/globalStorage/pleiades.java-extension-pack-jdk/java/17/bin/java -XX:+
ShowCodeDetailsInExceptionMessages -cp /Users/yagnanarayanan/Downloads/DN-4.0-Java-FSE-Mandatory-hands-on-answers-main/Week-2/TDD_JUnit5_Mockito
yagnanarayanan@yagnanarayanans-MacBook-Air 04_Mocking and Stubbing %  /usr/bin/env /Users/yagnanarayanan/Library/Application\ Support/Code/User/
globalStorage/pleiades.java-extension-pack-jdk/java/17/bin/java -XX:+ShowCodeDetailsInExceptionMessages -cp /Users/yagnanarayanan/Downloads/DN-4
.0-Java-FSE-Mandatory-hands-on-answers-main/Week-2/TDD_JUnit5_Mockito/04_Mocking\ and\ Stubbing/Code/mockingandstubbing/target/classes com.examp
le.App
Hello World!
yagnanarayanan@yagnanarayanans-MacBook-Air 04_Mocking and Stubbing %
```

Logging using SLF4J

Exercise 1: Logging Error Messages and Warning Levels

Task: Write a Java application that demonstrates logging error messages and warning levels

using SLF4J.

1. Add SLF4J and Logback dependencies to your `pom.xml` file:

2. Create a Java class that uses SLF4J for logging:

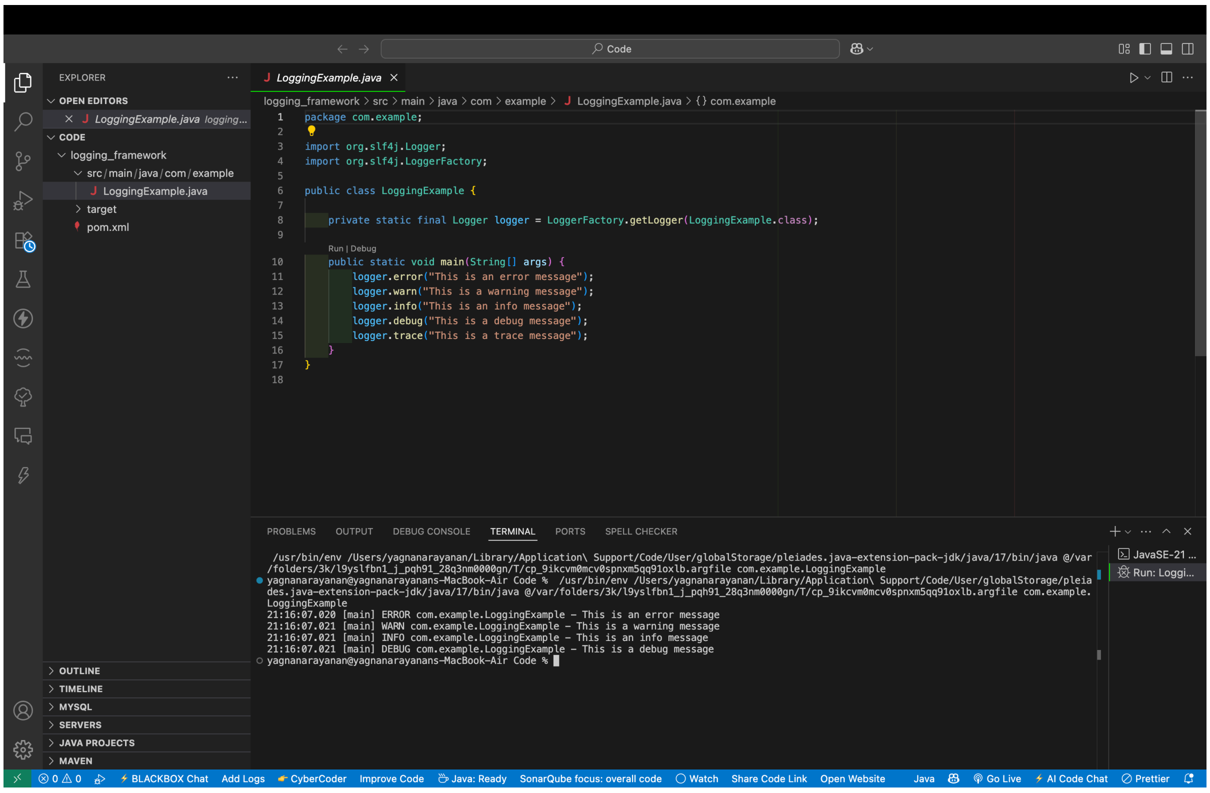
Code:

LoggingExample.java

```java
package com.example;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class LoggingExample {

    private static final Logger logger =
LoggerFactory.getLogger(LoggingExample.class);
```

```java
    public static void main(String[] args) {
        logger.error("This is an error message");
        logger.warn("This is a warning message");
        logger.info("This is an info message");
        logger.debug("This is a debug message");
        logger.trace("This is a trace message");
    }
}
```

Output:



```java
package com.example;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class LoggingExample {

    private static final Logger logger = LoggerFactory.getLogger(LoggingExample.class);

    public static void main(String[] args) {
        logger.error("This is an error message");
        logger.warn("This is a warning message");
        logger.info("This is an info message");
        logger.debug("This is a debug message");
        logger.trace("This is a trace message");
    }
}
```

```
 /usr/bin/env /Users/yagnanarayanan/Library/Application\ Support/Code/User/globalStorage/pleiades.java-extension-pack-jdk/java/17/bin/java @/var
/folders/3k/l9yslfbn1_j_pqh91_28q3nm0000gn/T/cp_9ikcvm0mcv0spnxm5qq91oxlb.argfile com.example.LoggingExample
yagnanarayanan@yagnanarayanans-MacBook-Air Code %  /usr/bin/env /Users/yagnanarayanan/Library/Application\ Support/Code/User/globalStorage/pleia
des.java-extension-pack-jdk/java/17/bin/java @/var/folders/3k/l9yslfbn1_j_pqh91_28q3nm0000gn/T/cp_9ikcvm0mcv0spnxm5qq91oxlb.argfile com.example.
LoggingExample
21:16:07.020 [main] ERROR com.example.LoggingExample - This is an error message
21:16:07.021 [main] WARN com.example.LoggingExample - This is a warning message
21:16:07.021 [main] INFO com.example.LoggingExample - This is an info message
21:16:07.021 [main] DEBUG com.example.LoggingExample - This is a debug message
yagnanarayanan@yagnanarayanans-MacBook-Air Code % 
```