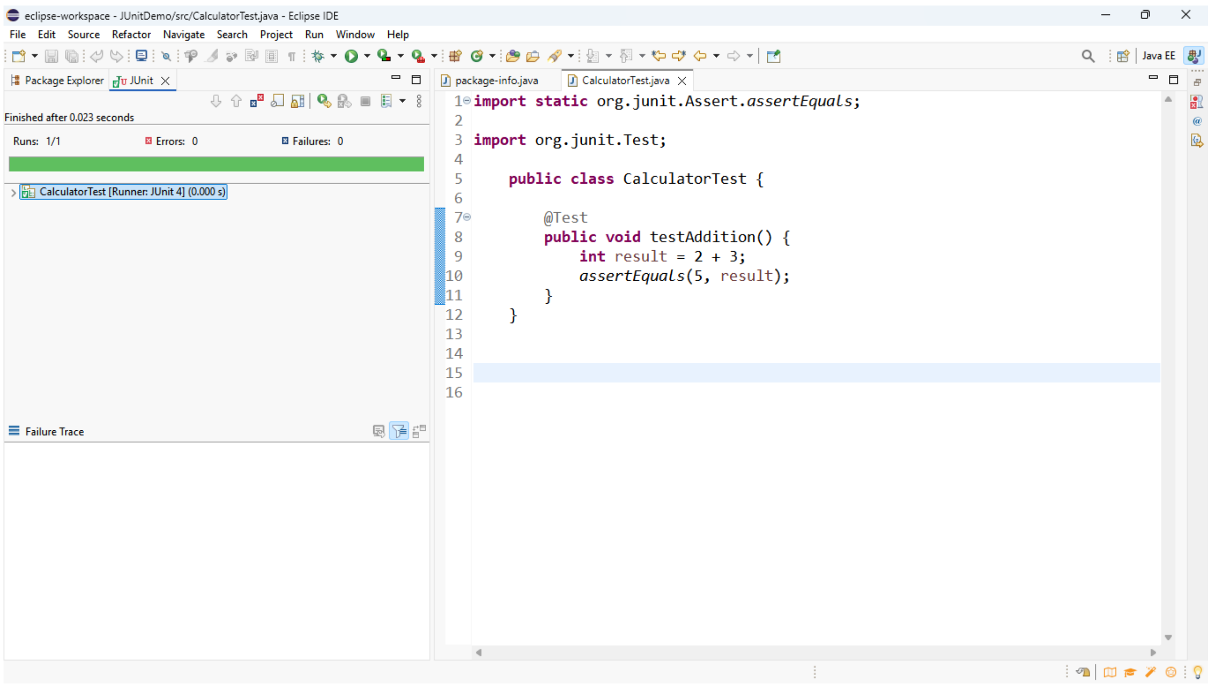
JUnit Testing Exercises

Exercise 1: Setting Up JUnit


eclipse-workspace - JUnitDemo/src/CalculatorTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
package-info.java
CalculatorTest.java
Java EE

Finished after 0.023 seconds

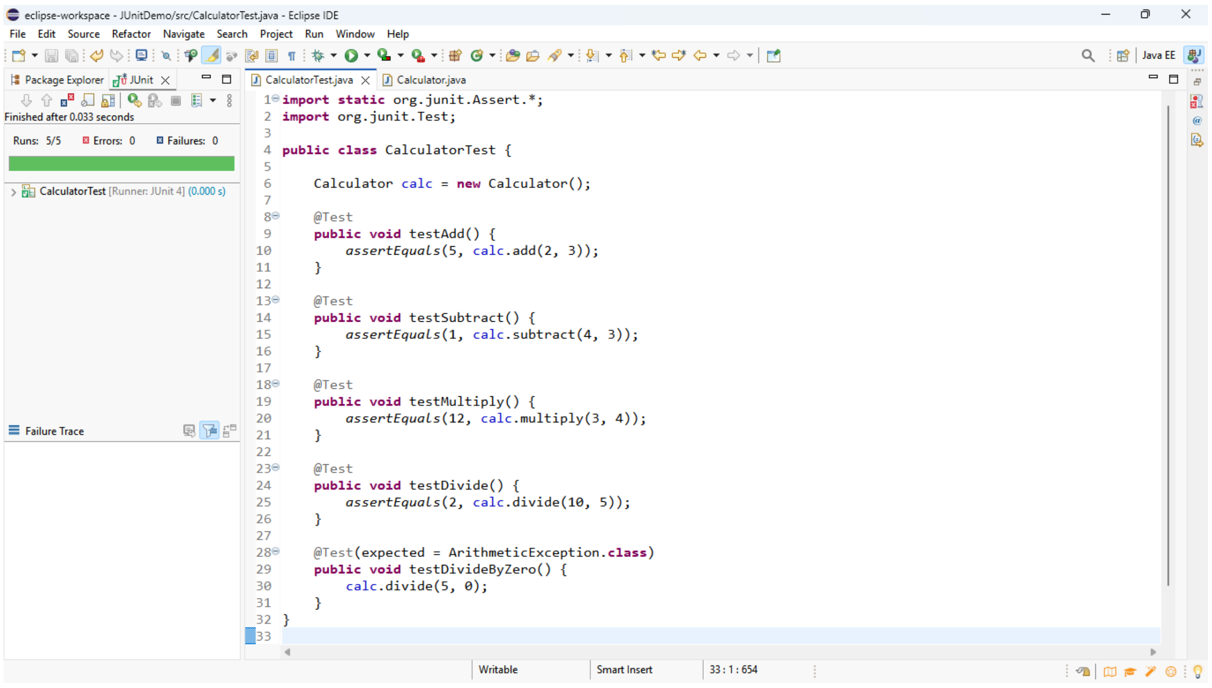
```
Runs: 1/1        Errors: 0        Failures: 0
```

CalculatorTest [Runner: JUnit 4] (0.000 s)

Failure Trace

```java
import static org.junit.Assert.assertEquals;

import org.junit.Test;

   public class CalculatorTest {

       @Test
       public void testAddition() {
           int result = 2 + 3;
           assertEquals(5, result);
       }
   }
```

Exercise 2: Writing Basic JUnit Tests


eclipse-workspace - JUnitDemo/src/CalculatorTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
CalculatorTest.java
Calculator.java
Java EE
Writable
Smart Insert
33 : 1 : 654

Finished after 0.033 seconds

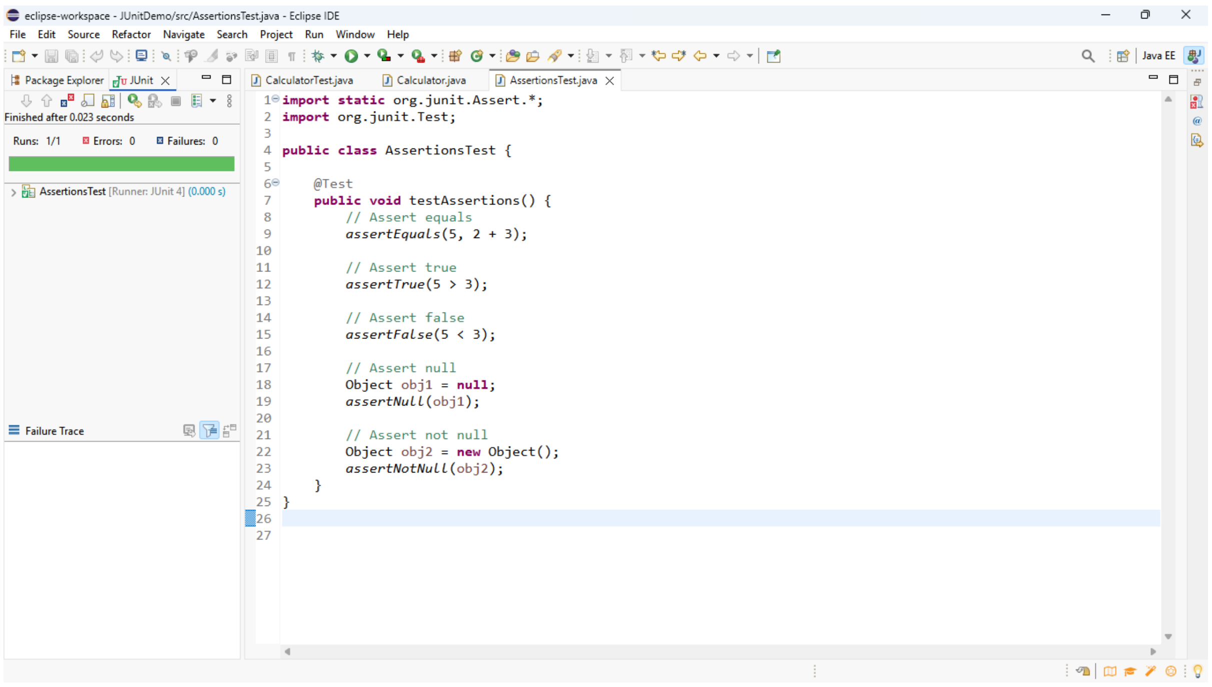
```
Runs: 5/5    Errors: 0    Failures: 0
```

CalculatorTest [Runner: JUnit 4] (0.000 s)

Failure Trace

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

public class CalculatorTest {

    Calculator calc = new Calculator();

    @Test
    public void testAdd() {
        assertEquals(5, calc.add(2, 3));
    }

    @Test
    public void testSubtract() {
        assertEquals(1, calc.subtract(4, 3));
    }

    @Test
    public void testMultiply() {
        assertEquals(12, calc.multiply(3, 4));
    }

    @Test
    public void testDivide() {
        assertEquals(2, calc.divide(10, 5));
    }

    @Test(expected = ArithmeticException.class)
    public void testDivideByZero() {
        calc.divide(5, 0);
    }
}
```

Exercise 3: Assertions in JUnit



eclipse-workspace - JUnitDemo/src/AssertionsTest.java - Eclipse IDE

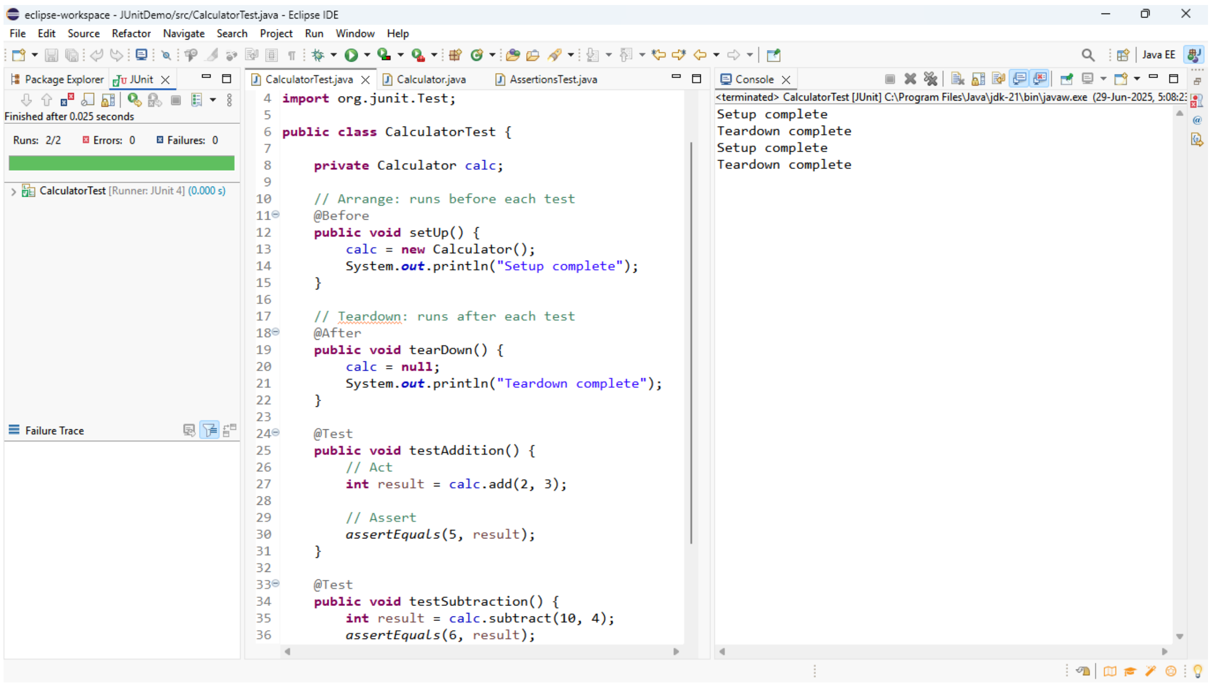
Finished after 0.023 seconds

Runs: 1/1 Errors: 0 Failures: 0

AssertionsTest [Runner: JUnit 4] (0.000 s)

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

public class AssertionsTest {

    @Test
    public void testAssertions() {
        // Assert equals
        assertEquals(5, 2 + 3);

        // Assert true
        assertTrue(5 > 3);

        // Assert false
        assertFalse(5 < 3);

        // Assert null
        Object obj1 = null;
        assertNull(obj1);

        // Assert not null
        Object obj2 = new Object();
        assertNotNull(obj2);
    }
}
```

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



eclipse-workspace - JUnitDemo/src/CalculatorTest.java - Eclipse IDE

Finished after 0.025 seconds

Runs: 2/2 Errors: 0 Failures: 0

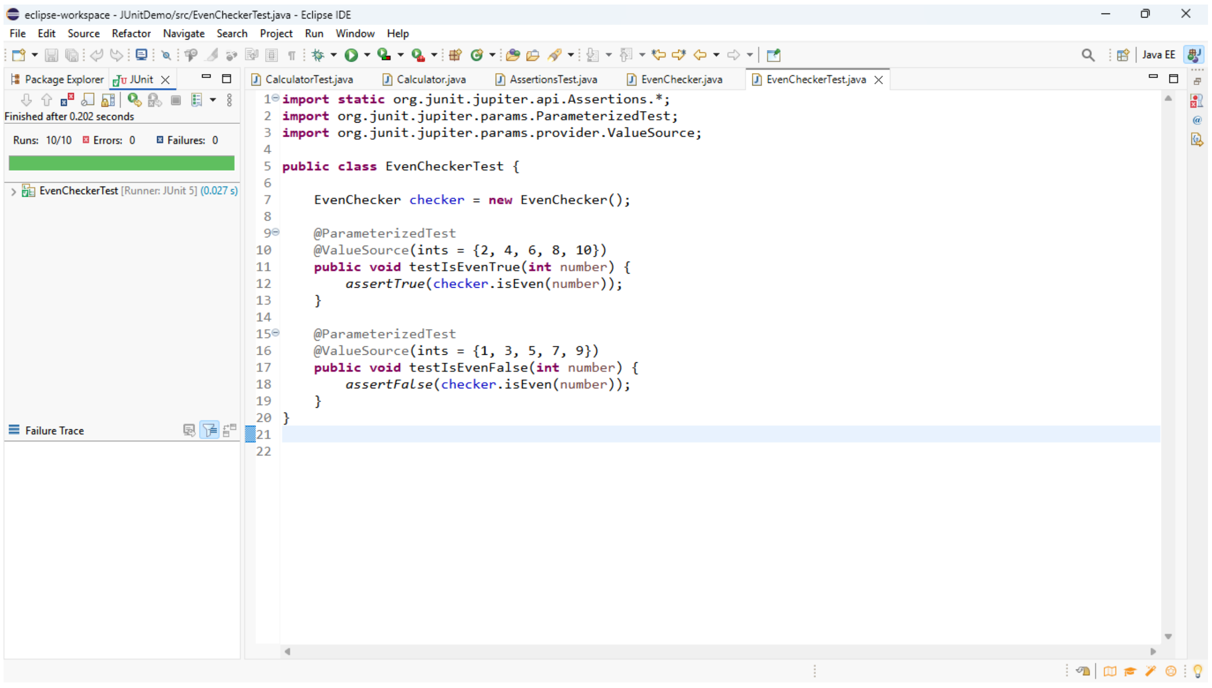
CalculatorTest [Runner: JUnit 4] (0.000 s)

```java
import org.junit.Test;

public class CalculatorTest {

    private Calculator calc;

    // Arrange: runs before each test
    @Before
    public void setUp() {
        calc = new Calculator();
        System.out.println("Setup complete");
    }

    // Teardown: runs after each test
    @After
    public void tearDown() {
        calc = null;
        System.out.println("Teardown complete");
    }

    @Test
    public void testAddition() {
        // Act
        int result = calc.add(2, 3);

        // Assert
        assertEquals(5, result);
    }

    @Test
    public void testSubtraction() {
        int result = calc.subtract(10, 4);
        assertEquals(6, result);
```

```
<terminated> CalculatorTest [JUnit] C:\Program Files\Java\jdk-21\bin\javaw.exe (29-Jun-2025, 5:08:2
Setup complete
Teardown complete
Setup complete
Teardown complete
```

Advanced JUnit Testing Exercises

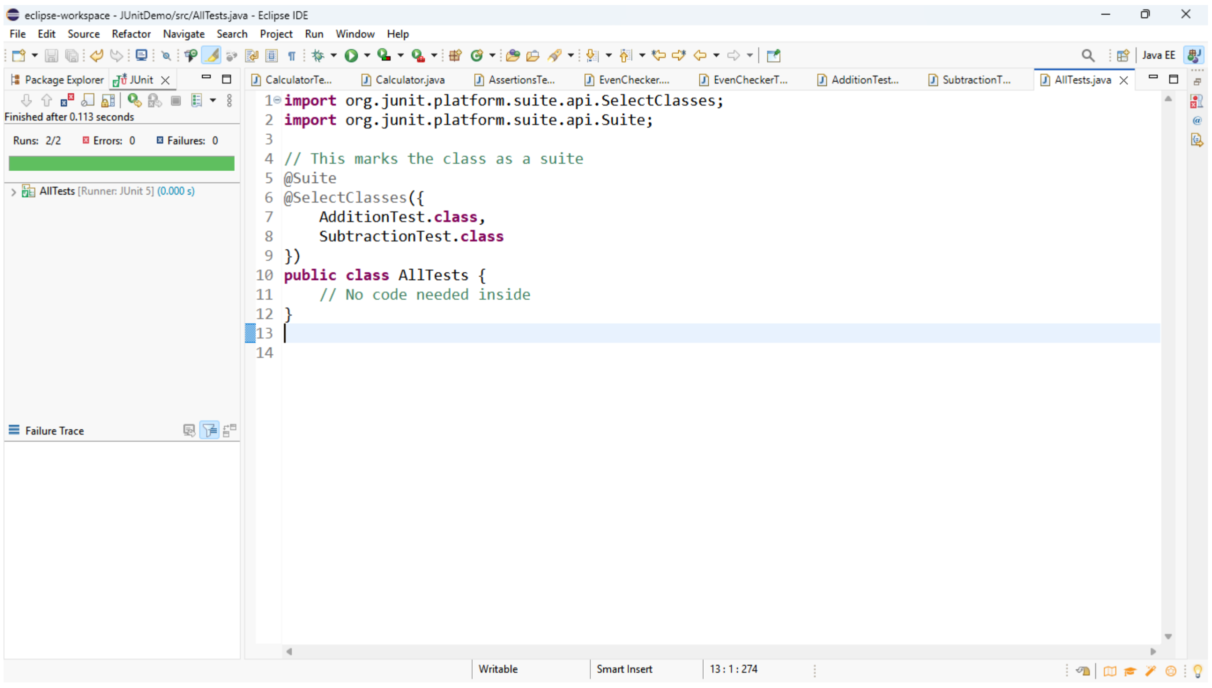
Exercise 1: Parameterized Tests


eclipse-workspace - JUnitDemo/src/EvenCheckerTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
CalculatorTest.java
Calculator.java
AssertionsTest.java
EvenChecker.java
EvenCheckerTest.java
Failure Trace
Java EE

```
Finished after 0.202 seconds
Runs: 10/10   Errors: 0   Failures: 0
EvenCheckerTest [Runner: JUnit 5] (0.027 s)
```

```java
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.params.ParameterizedTest;
import org.junit.jupiter.params.provider.ValueSource;

public class EvenCheckerTest {

    EvenChecker checker = new EvenChecker();

    @ParameterizedTest
    @ValueSource(ints = {2, 4, 6, 8, 10})
    public void testIsEvenTrue(int number) {
        assertTrue(checker.isEven(number));
    }

    @ParameterizedTest
    @ValueSource(ints = {1, 3, 5, 7, 9})
    public void testIsEvenFalse(int number) {
        assertFalse(checker.isEven(number));
    }
}
```

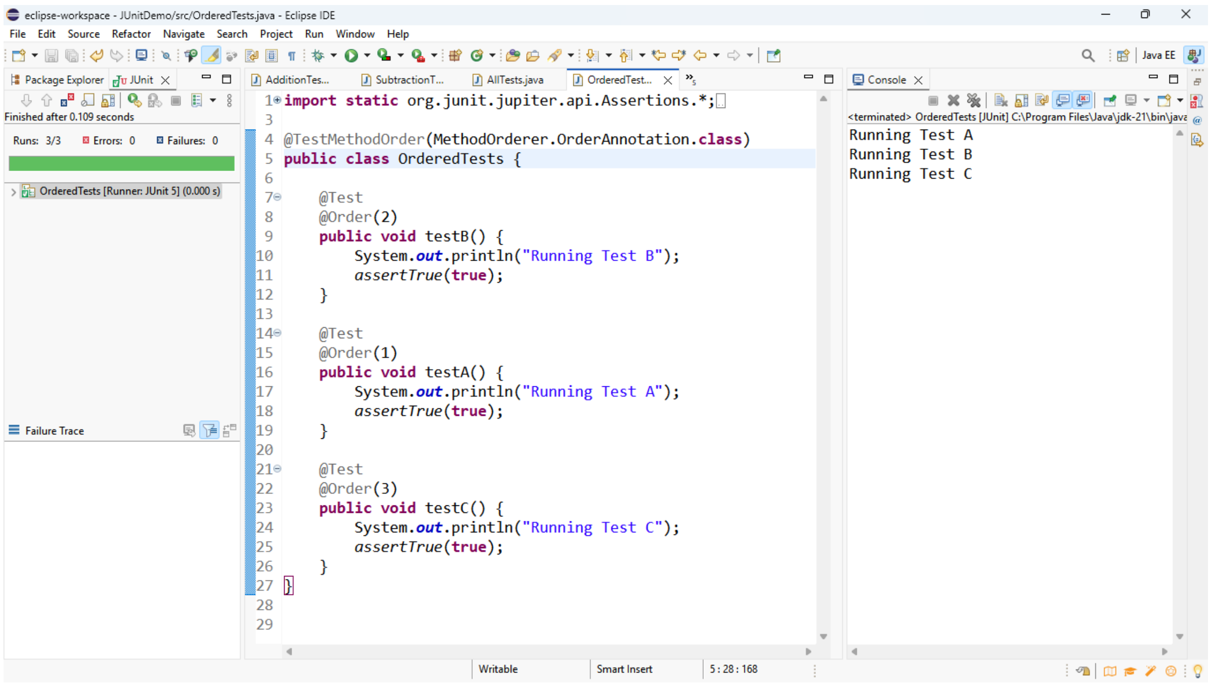
Exercise 2: Test Suites and Categories



```
Finished after 0.113 seconds
Runs: 2/2   Errors: 0   Failures: 0
AllTests [Runner: JUnit 5] (0.000 s)
```

```java
import org.junit.platform.suite.api.SelectClasses;
import org.junit.platform.suite.api.Suite;

// This marks the class as a suite
@Suite
@SelectClasses({
    AdditionTest.class,
    SubtractionTest.class
})
public class AllTests {
    // No code needed inside
}
```

Exercise 3: Test Execution Order


eclipse-workspace - JUnitDemo/src/OrderedTests.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
AdditionTes...
SubtractionT...
AllTests.java
OrderedTest...
Console
Failure Trace
Writable
Smart Insert
5 : 28 : 168

Finished after 0.109 seconds

Runs: 3/3 Errors: 0 Failures: 0

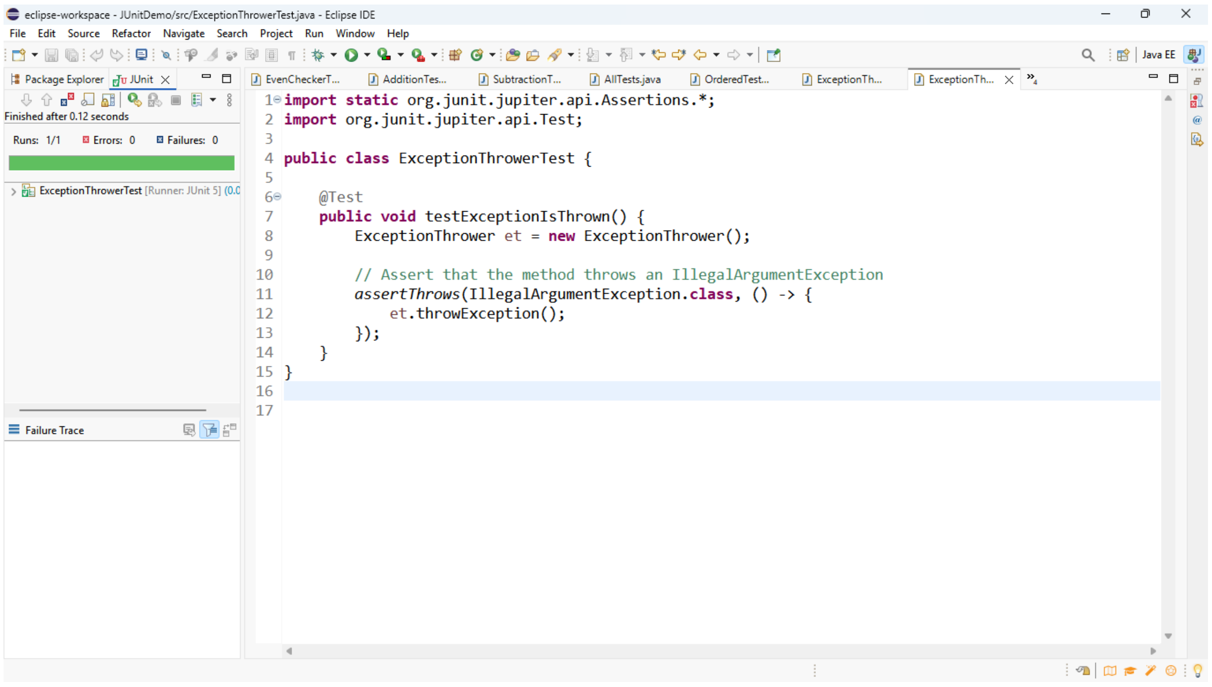
OrderedTests [Runner: JUnit 5] (0.000 s)

```java
import static org.junit.jupiter.api.Assertions.*;

@TestMethodOrder(MethodOrderer.OrderAnnotation.class)
public class OrderedTests {

    @Test
    @Order(2)
    public void testB() {
        System.out.println("Running Test B");
        assertTrue(true);
    }

    @Test
    @Order(1)
    public void testA() {
        System.out.println("Running Test A");
        assertTrue(true);
    }

    @Test
    @Order(3)
    public void testC() {
        System.out.println("Running Test C");
        assertTrue(true);
    }
}
```

<terminated> OrderedTests [JUnit] C:\Program Files\Java\jdk-21\bin\java

```
Running Test A
Running Test B
Running Test C
```

Exercise 4: Exception Testing


eclipse-workspace - JUnitDemo/src/ExceptionThrowerTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
EvenCheckerT...
AdditionTes...
SubtractionT...
AllTests.java
OrderedTest...
ExceptionTh...
ExceptionTh...
Failure Trace

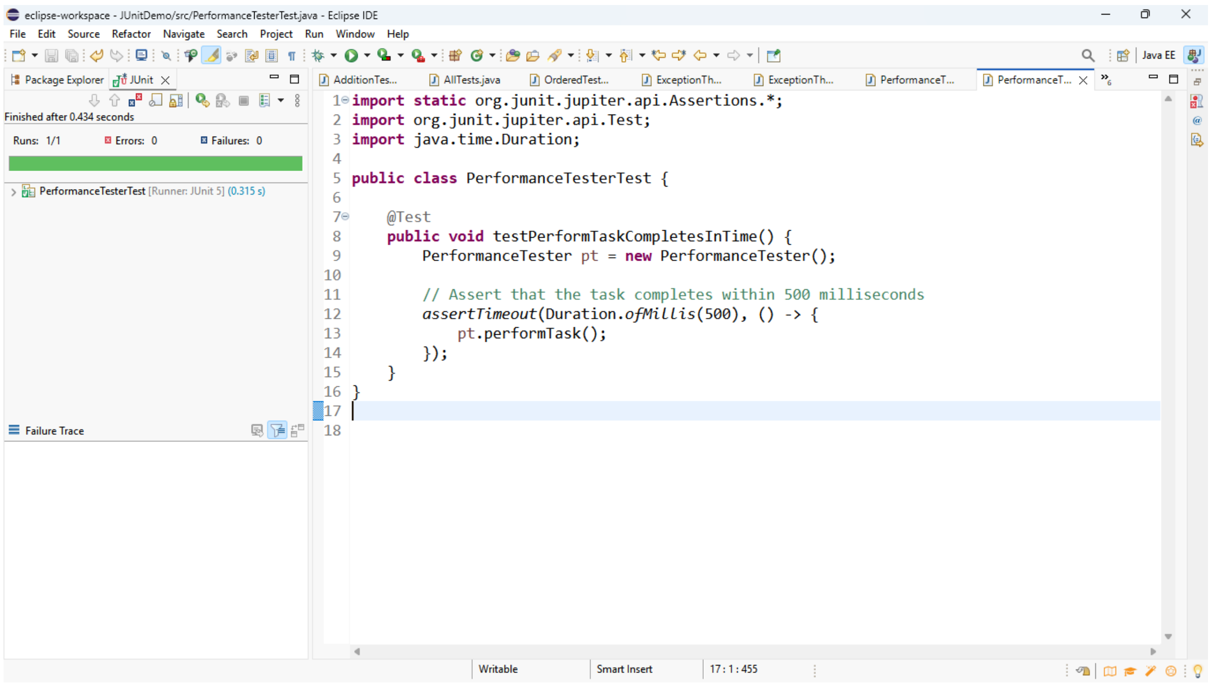
Finished after 0.12 seconds

Runs: 1/1 Errors: 0 Failures: 0

ExceptionThrowerTest [Runner: JUnit 5] (0.0

```java
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;

public class ExceptionThrowerTest {

    @Test
    public void testExceptionIsThrown() {
        ExceptionThrower et = new ExceptionThrower();

        // Assert that the method throws an IllegalArgumentException
        assertThrows(IllegalArgumentException.class, () -> {
            et.throwException();
        });
    }
}
```

Exercise 5: Timeout and Performance Testing

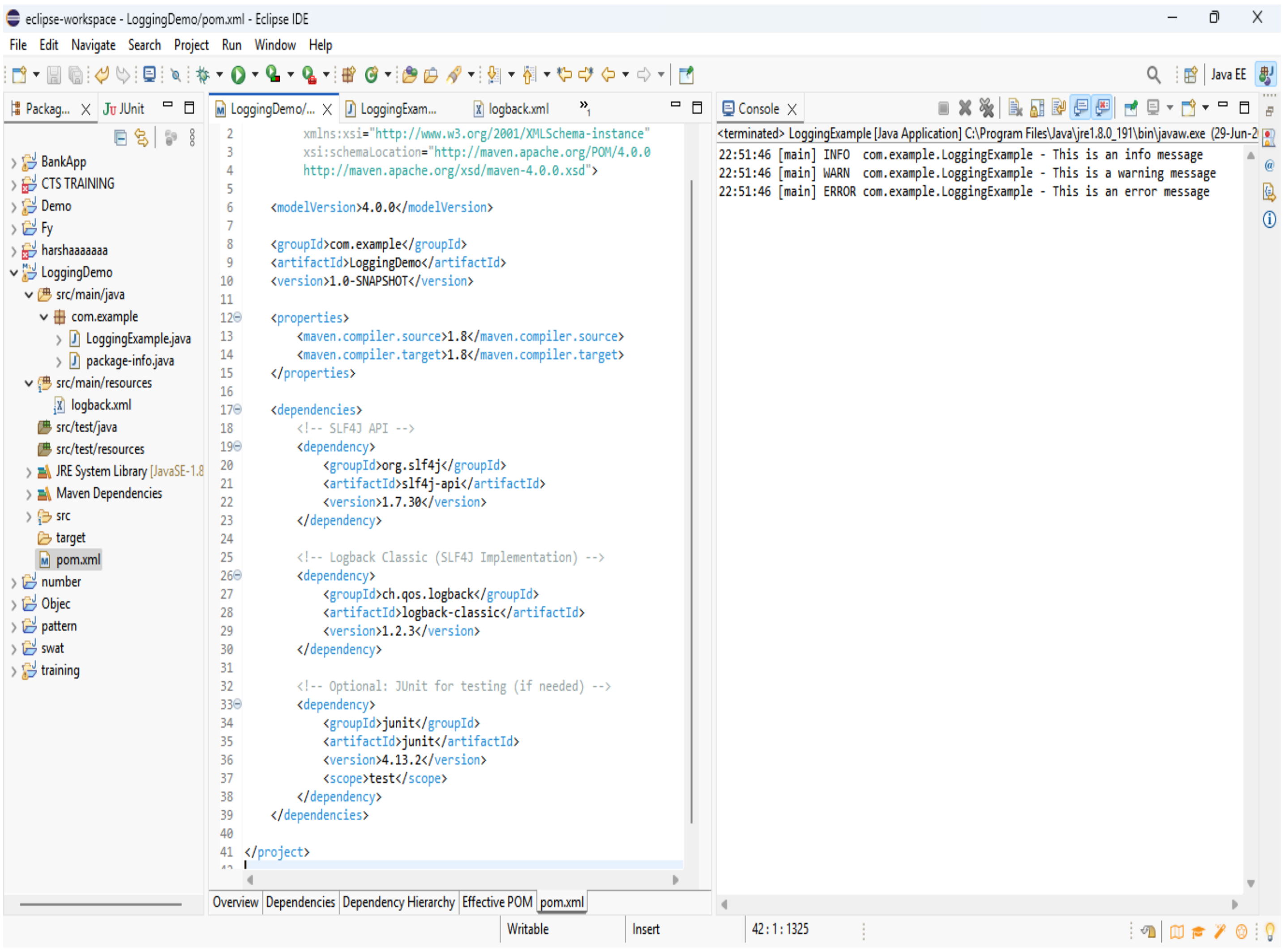


```
Finished after 0.434 seconds
Runs: 1/1    Errors: 0    Failures: 0
PerformanceTesterTest [Runner: JUnit 5] (0.315 s)
```

```java
import static org.junit.jupiter.api.Assertions.*;
import org.junit.jupiter.api.Test;
import java.time.Duration;

public class PerformanceTesterTest {

    @Test
    public void testPerformTaskCompletesInTime() {
        PerformanceTester pt = new PerformanceTester();

        // Assert that the task completes within 500 milliseconds
        assertTimeout(Duration.ofMillis(500), () -> {
            pt.performTask();
        });
    }
}
```

Logging using SLF4J

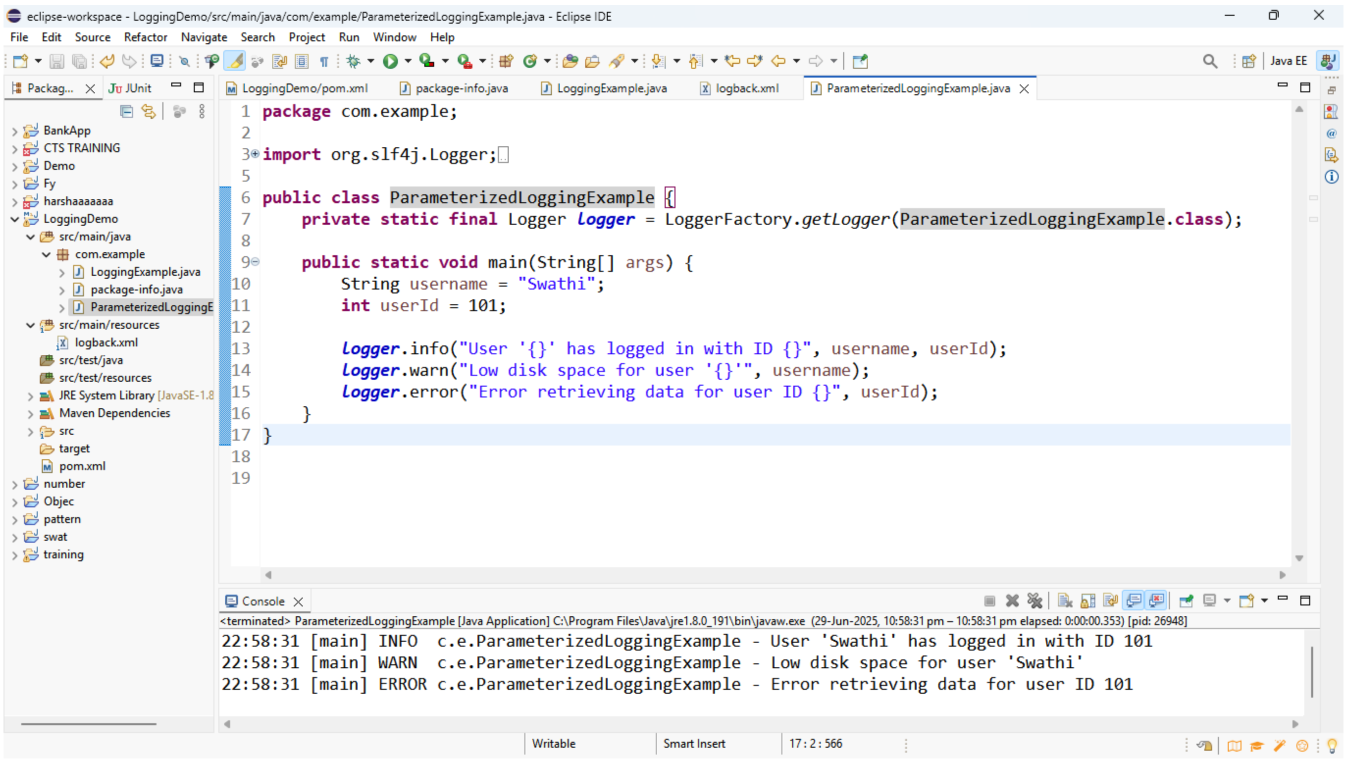
Exercise 1: Logging Error Messages and Warning Levels



```xml
         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>LoggingDemo</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>

    <dependencies>
        <!-- SLF4J API -->
        <dependency>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-api</artifactId>
            <version>1.7.30</version>
        </dependency>

        <!-- Logback Classic (SLF4J Implementation) -->
        <dependency>
            <groupId>ch.qos.logback</groupId>
            <artifactId>logback-classic</artifactId>
            <version>1.2.3</version>
        </dependency>

        <!-- Optional: JUnit for testing (if needed) -->
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>4.13.2</version>
            <scope>test</scope>
        </dependency>
    </dependencies>

</project>
```

```
<terminated> LoggingExample [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe  (29-Jun-2
22:51:46 [main] INFO  com.example.LoggingExample - This is an info message
22:51:46 [main] WARN  com.example.LoggingExample - This is a warning message
22:51:46 [main] ERROR com.example.LoggingExample - This is an error message
```

Exercise 2: Parameterized Logging

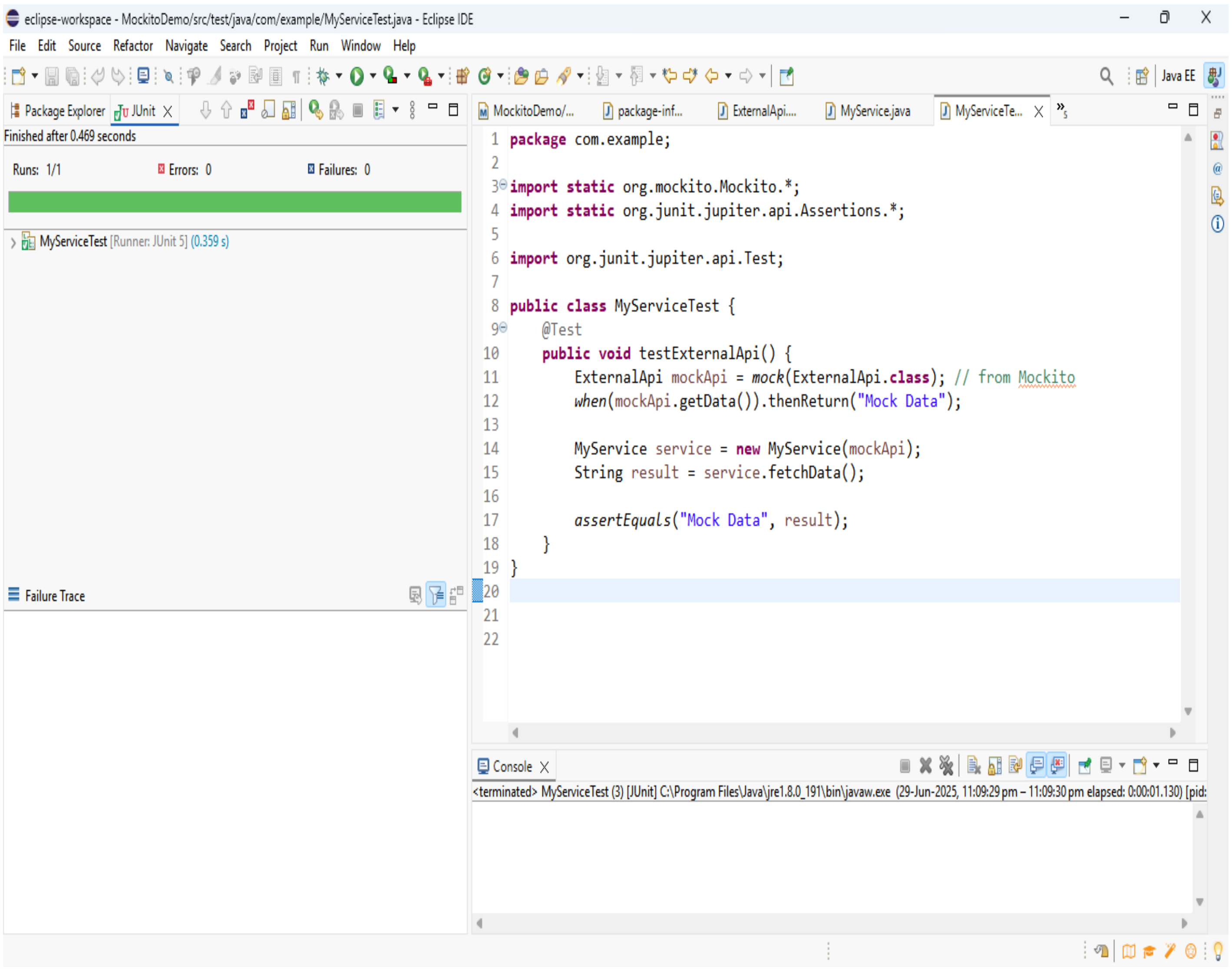


```java
package com.example;

import org.slf4j.Logger;

public class ParameterizedLoggingExample {
    private static final Logger logger = LoggerFactory.getLogger(ParameterizedLoggingExample.class);

    public static void main(String[] args) {
        String username = "Swathi";
        int userId = 101;

        logger.info("User '{}' has logged in with ID {}", username, userId);
        logger.warn("Low disk space for user '{}'", username);
        logger.error("Error retrieving data for user ID {}", userId);
    }
}
```

```
<terminated> ParameterizedLoggingExample [Java Application] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (29-Jun-2025, 10:58:31 pm – 10:58:31 pm elapsed: 0:00:00.353) [pid: 26948]
22:58:31 [main] INFO  c.e.ParameterizedLoggingExample - User 'Swathi' has logged in with ID 101
22:58:31 [main] WARN  c.e.ParameterizedLoggingExample - Low disk space for user 'Swathi'
22:58:31 [main] ERROR c.e.ParameterizedLoggingExample - Error retrieving data for user ID 101
```

Mockito Hands-On Exercises

Exercise 1: Mocking and Stubbing


eclipse-workspace - MockitoDemo/src/test/java/com/example/MyServiceTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer
JUnit
MockitoDemo/...
package-inf...
ExternalApi...
MyService.java
MyServiceTe...
Java EE
Failure Trace
Console

Finished after 0.469 seconds

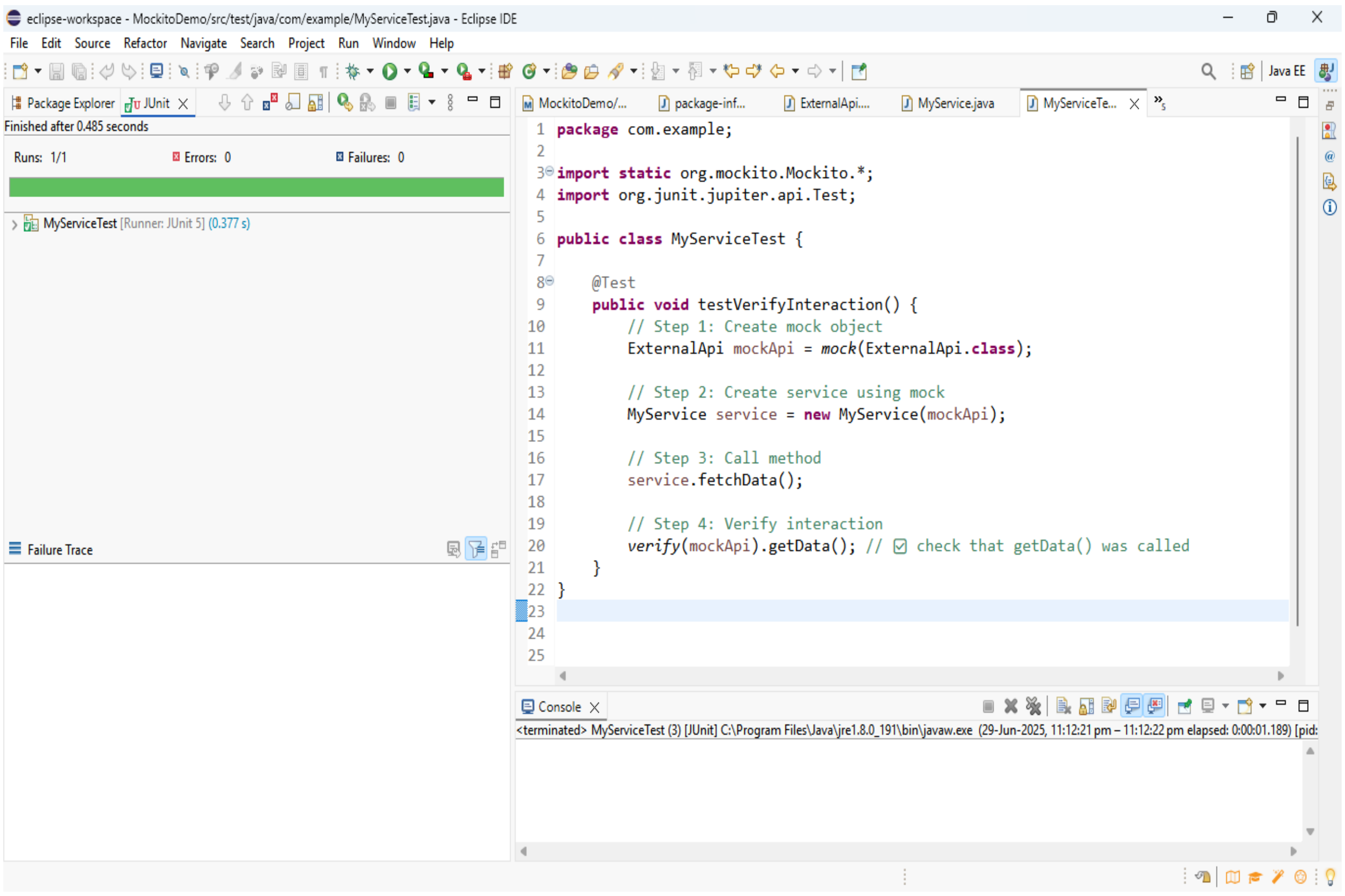
Runs: 1/1 Errors: 0 Failures: 0

MyServiceTest [Runner: JUnit 5] (0.359 s)

```java
package com.example;

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

import org.junit.jupiter.api.Test;

public class MyServiceTest {
    @Test
    public void testExternalApi() {
        ExternalApi mockApi = mock(ExternalApi.class); // from Mockito
        when(mockApi.getData()).thenReturn("Mock Data");

        MyService service = new MyService(mockApi);
        String result = service.fetchData();

        assertEquals("Mock Data", result);
    }
}
```

```
<terminated> MyServiceTest (3) [JUnit] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (29-Jun-2025, 11:09:29 pm – 11:09:30 pm elapsed: 0:00:01.130) [pid:
```

Exercise 2: Verifying Interactions


eclipse-workspace - MockitoDemo/src/test/java/com/example/MyServiceTest.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Java EE
Package Explorer
JUnit
MockitoDemo/...
package-inf...
ExternalApi....
MyService.java
MyServiceTe...
Failure Trace
Console

Finished after 0.485 seconds

Runs: 1/1 Errors: 0 Failures: 0

MyServiceTest [Runner: JUnit 5] (0.377 s)

```java
package com.example;

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

public class MyServiceTest {

    @Test
    public void testVerifyInteraction() {
        // Step 1: Create mock object
        ExternalApi mockApi = mock(ExternalApi.class);

        // Step 2: Create service using mock
        MyService service = new MyService(mockApi);

        // Step 3: Call method
        service.fetchData();

        // Step 4: Verify interaction
        verify(mockApi).getData(); // ☑ check that getData() was called
    }
}
```

```
<terminated> MyServiceTest (3) [JUnit] C:\Program Files\Java\jre1.8.0_191\bin\javaw.exe (29-Jun-2025, 11:12:21 pm – 11:12:22 pm elapsed: 0:00:01.189) [pid:
```