

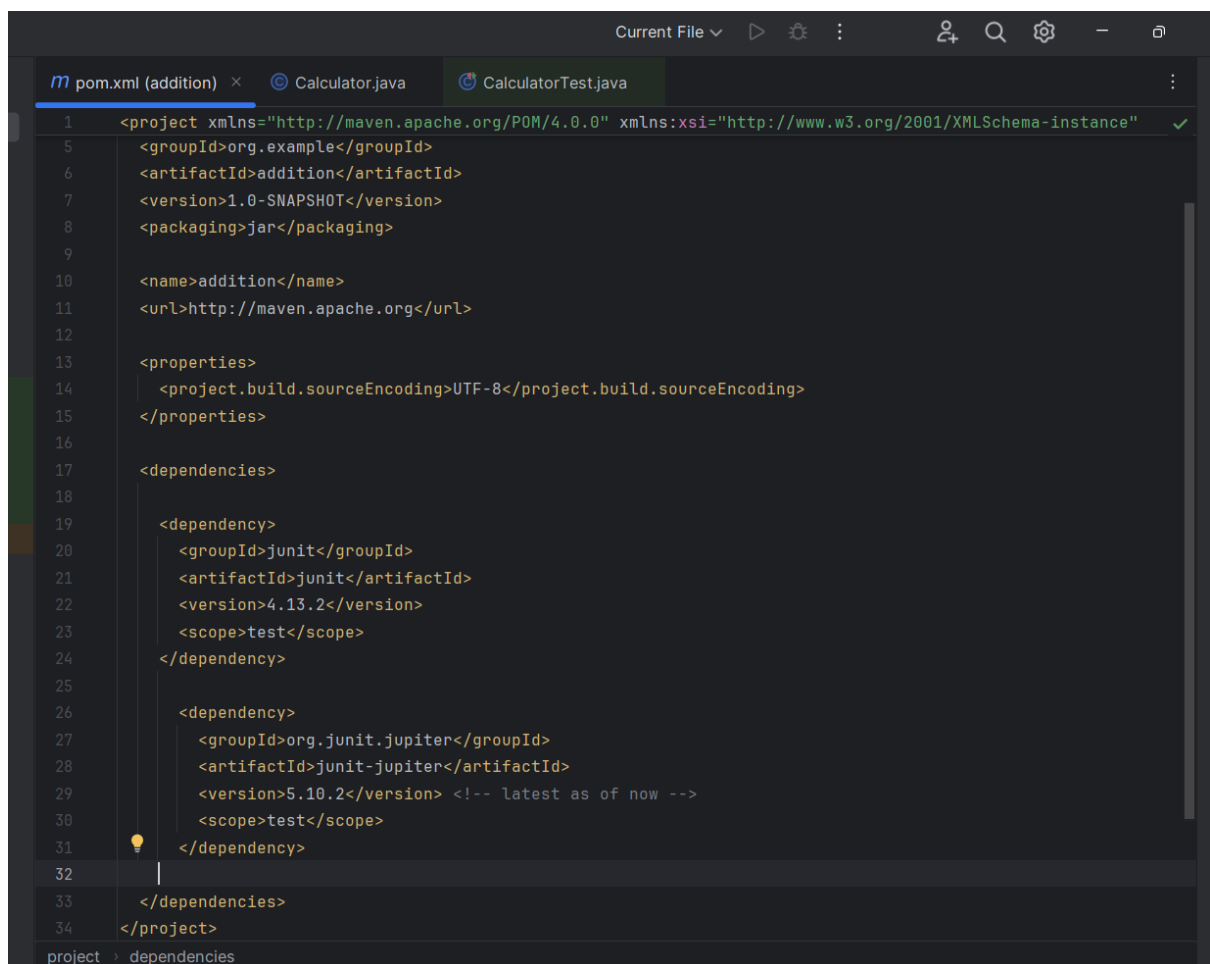
## Exercise 1: Setting Up Junit

### FROM File 1. JUnit\_Basic Testing Exercises

---

#### Dependency adding

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

A screenshot of an IDE window showing a Maven project's pom.xml file. The file is named 'pom.xml (addition)' and is open in a tab alongside 'Calculator.java' and 'CalculatorTest.java'. The XML content is as follows:

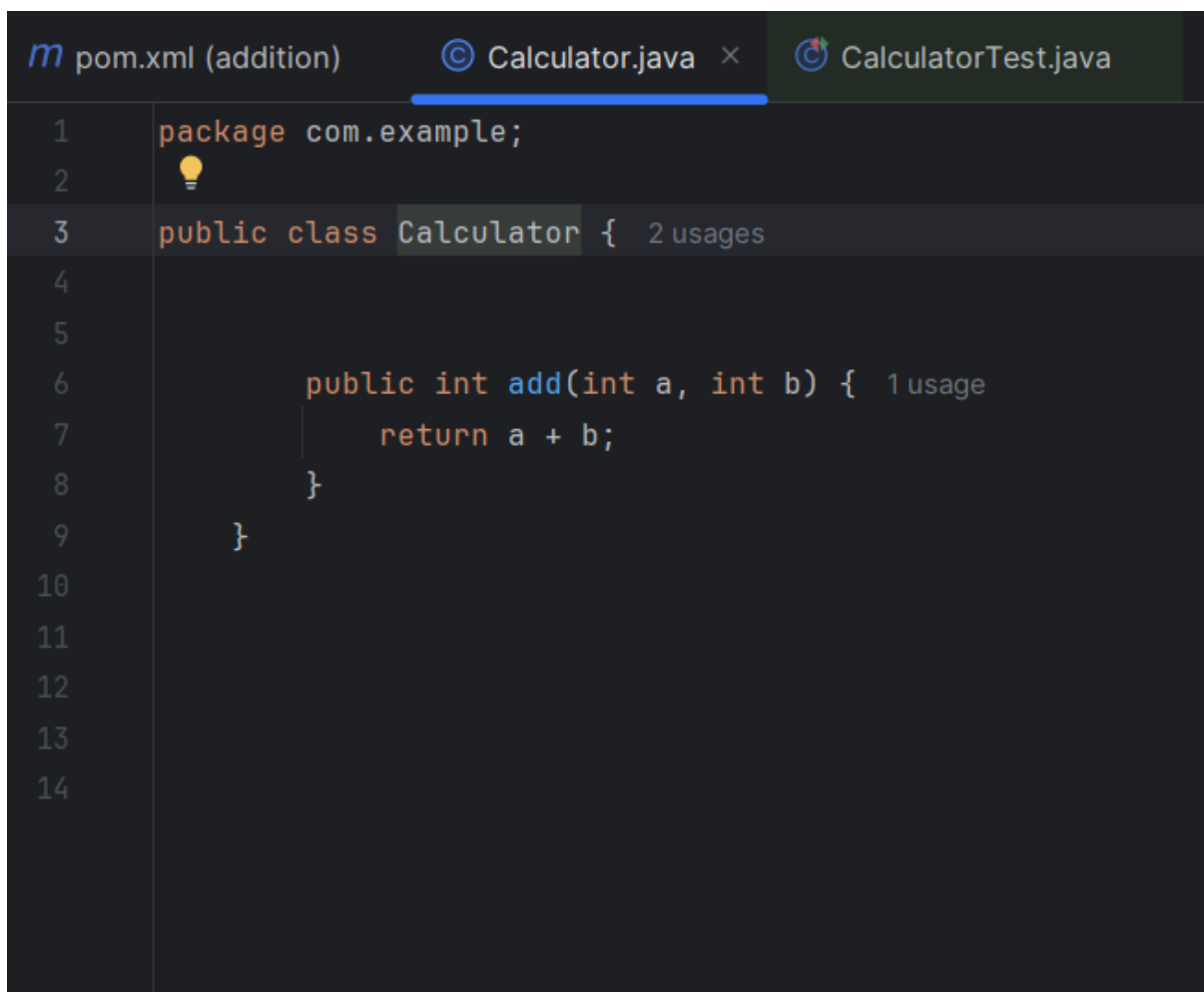
```
1 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ✓
5   <groupId>org.example</groupId>
6   <artifactId>addition</artifactId>
7   <version>1.0-SNAPSHOT</version>
8   <packaging>jar</packaging>
9
10  <name>addition</name>
11  <url>http://maven.apache.org</url>
12
13  <properties>
14    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
15  </properties>
16
17  <dependencies>
18
19    <dependency>
20      <groupId>junit</groupId>
21      <artifactId>junit</artifactId>
22      <version>4.13.2</version>
23      <scope>test</scope>
24    </dependency>
25
26    <dependency>
27      <groupId>org.junit.jupiter</groupId>
28      <artifactId>junit-jupiter</artifactId>
29      <version>5.10.2</version> <!-- latest as of now -->
30      <scope>test</scope>
31    </dependency>
32
33  </dependencies>
34 </project>
```

The IDE interface includes a top bar with 'Current File' and various icons. The bottom status bar shows the current path as 'project > dependencies'.

Calculator Class:

```
package com.example;
```

```
public class Calculator {  
    public int add(int a, int b) {  
        return a + b;  
    }  
}
```



The screenshot shows an IDE window with three tabs: 'pom.xml (addition)', 'Calculator.java', and 'CalculatorTest.java'. The 'Calculator.java' tab is active and shows the following code with syntax highlighting and usage counts:

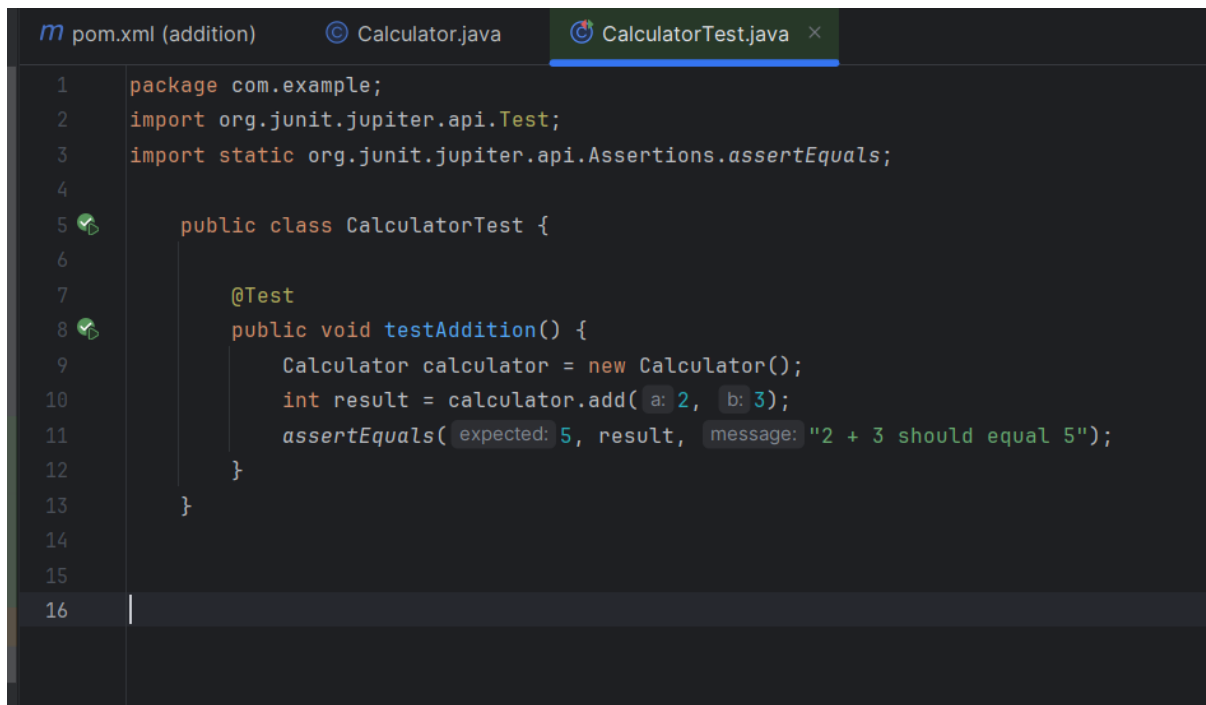
```
1 package com.example;  
2  
3 public class Calculator { 2 usages  
4  
5  
6     public int add(int a, int b) { 1 usage  
7         return a + b;  
8     }  
9 }  
10  
11  
12  
13  
14
```

## CalculatorTest:

```
package com.example;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.assertEquals;
```

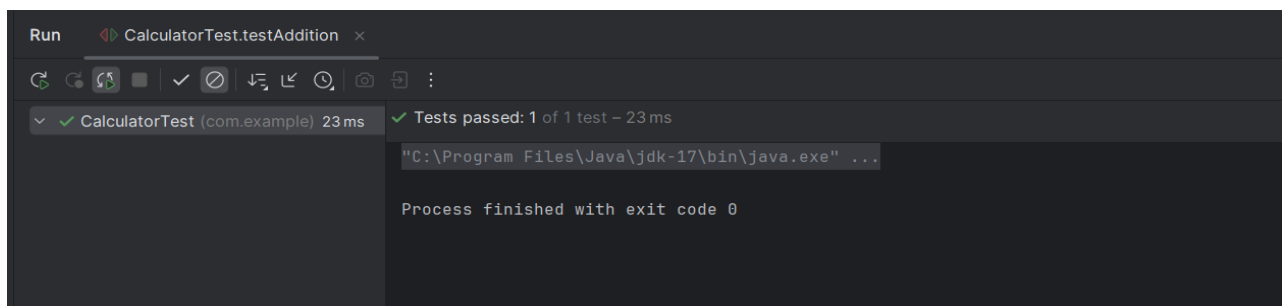
```
public class CalculatorTest {

    @Test
    public void testAddition() {
        Calculator calculator = new Calculator();
        int result = calculator.add(2, 3);
        assertEquals(5, result, "2 + 3 should equal 5");
    }
}
```

A screenshot of an IDE window showing the code for CalculatorTest.java. The window has three tabs: 'pom.xml (addition)', 'Calculator.java', and 'CalculatorTest.java' (which is active). The code is as follows:

```
1 package com.example;
2 import org.junit.jupiter.api.Test;
3 import static org.junit.jupiter.api.Assertions.assertEquals;
4
5 public class CalculatorTest {
6
7     @Test
8     public void testAddition() {
9         Calculator calculator = new Calculator();
10        int result = calculator.add( a: 2, b: 3);
11        assertEquals( expected: 5, result, message: "2 + 3 should equal 5");
12    }
13 }
14
15
16
```

## Output:

A screenshot of the IDE's Run output window. The title bar says 'Run' and the active tab is 'CalculatorTest.testAddition'. The output shows a successful test run:

```
✓ CalculatorTest (com.example) 23 ms    ✓ Tests passed: 1 of 1 test - 23 ms
"C:\Program Files\Java\jdk-17\bin\java.exe" ...
Process finished with exit code 0
```