**Superset ID:**  6125132  
**Name:** Podugu Vinay Kumar

**Week : 2**

**Junit Basic Testing Exercises**

We create a basic calculator using the junit

**Pom.xml**  
We add dependencies in the pom.xml to let Maven automatically download and manage required libraries for our project. This ensures consistent builds, handles transitive dependencies, and simplifies project setup.  
  
<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

https://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>Junit</groupId>

<artifactId>Junit</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.13.2</version>

<scope>test</scope>

</dependency>

</dependencies>

<build>

<sourceDirectory>src/main/java</sourceDirectory>

<testSourceDirectory>src/test/java</testSourceDirectory>

<plugins>

<plugin>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.8.1</version>

<configuration>

<release>22</release>

</configuration>

</plugin>

</plugins>

</build>

</project>

**Main Caliculator.java:**

**public** **class** Calculator {

**public** **int** add(**int** a, **int** b) {

**return** a + b;

}

**public** **int** subtract(**int** a, **int** b) {

**return** a - b;

}

**public** **int** multiply(**int** a, **int** b) {

**return** a \* b;

}

**public** **int** divide(**int** a, **int** b) {

**if** (b == 0) {

**throw** **new** IllegalArgumentException("Division by zero is not allowed.");

}

**return** a / b;

}

}

**CalculatorTest.java:**

import static org.junit.Assert.\*;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

public class CalculatorTest {

private Calculator calculator;

@Before

public void setUp() {

calculator = new Calculator();

}

@After

public void tearDown() {

calculator = null;

}

@Test

public void testAdd() {

*assertEquals*(5, calculator.add(2, 3));

}

@Test

public void testSubtract() {

*assertEquals*(1, calculator.subtract(3, 2));

}

@Test

public void testMultiply() {

*assertEquals*(6, calculator.multiply(2, 3));

}

@Test

public void testDivide() {

*assertEquals*(2, calculator.divide(6, 3));

}

@Test(expected = IllegalArgumentException.class)

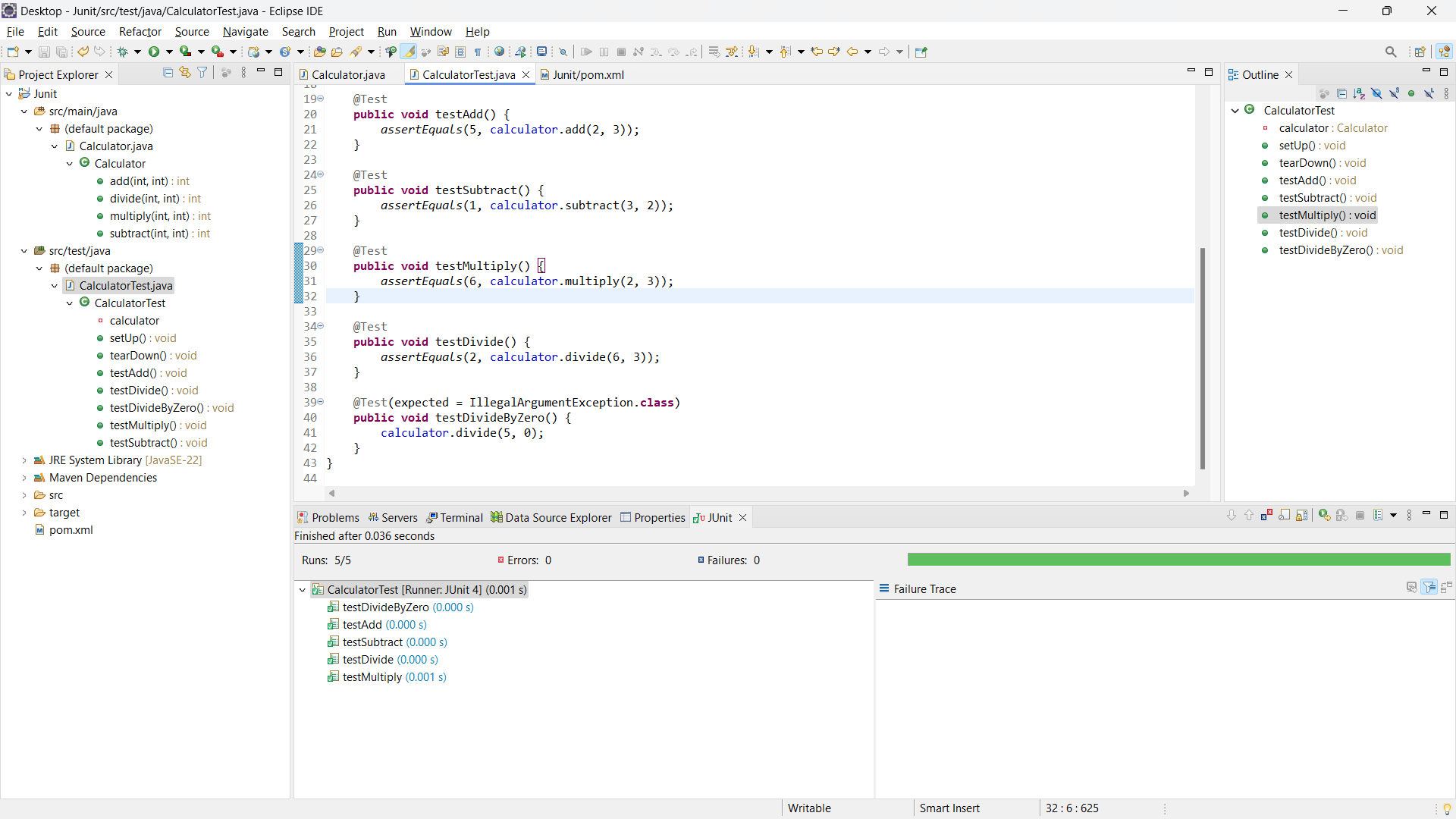
public void testDivideByZero() {

calculator.divide(5, 0);

}

}

**Output :**

****