WEEK – 2

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

Teardown Methods in Junit

Nipuna.A

Superset id: 6432842

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.

Solution:

Note: to implement exercise 4 , I havents created a separate maven project again. In the existing maven project of setting up Junit and Assertions itself I have added extra codes to implement it.

1)app.java

package com.example;

public class App

{

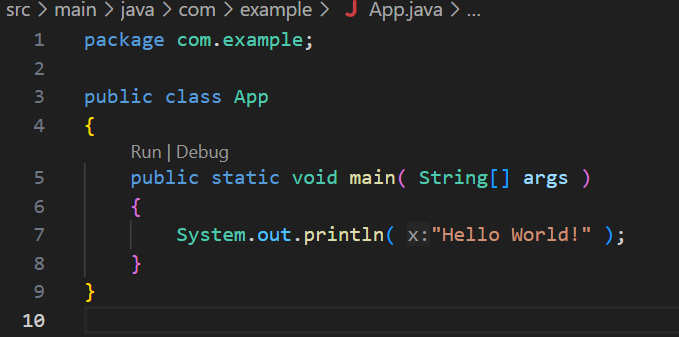
public static void main( String[] args )

{

System.out.println( "Hello World!" );

}

}



2)app.test.java

package com.example;

import org.junit.jupiter.api.Test;

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

public class AppTest {

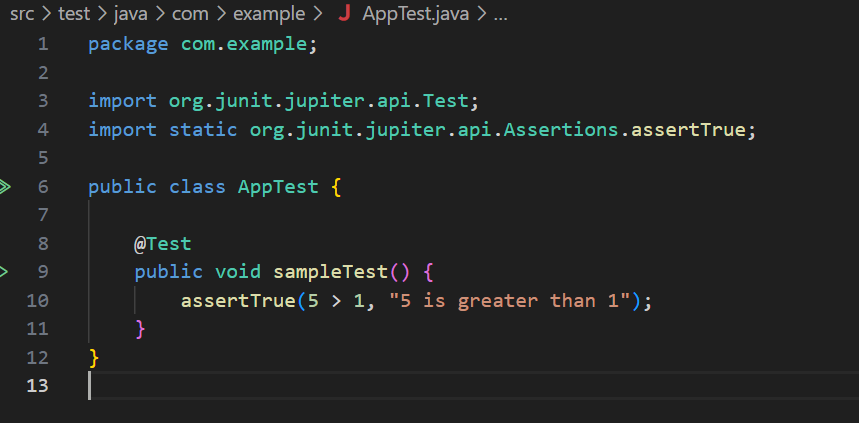
    @Test

    public void sampleTest() {

        assertTrue(5 > 1, "5 is greater than 1");

    }

}



3)pom.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>junit-setup-example</artifactId>

  <version>1.0-SNAPSHOT</version>

  <name>junit-setup-example</name>

  <url>http://www.example.com</url>

  <properties>

    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

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

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

  </properties>

  <dependencies>

    <dependency>

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

      <artifactId>junit-jupiter</artifactId>

      <version>5.10.2</version>

      <scope>test</scope>

    </dependency>

  </dependencies>

  <build>

    <plugins>

      <plugin>

        <groupId>org.apache.maven.plugins</groupId>

        <artifactId>maven-surefire-plugin</artifactId>

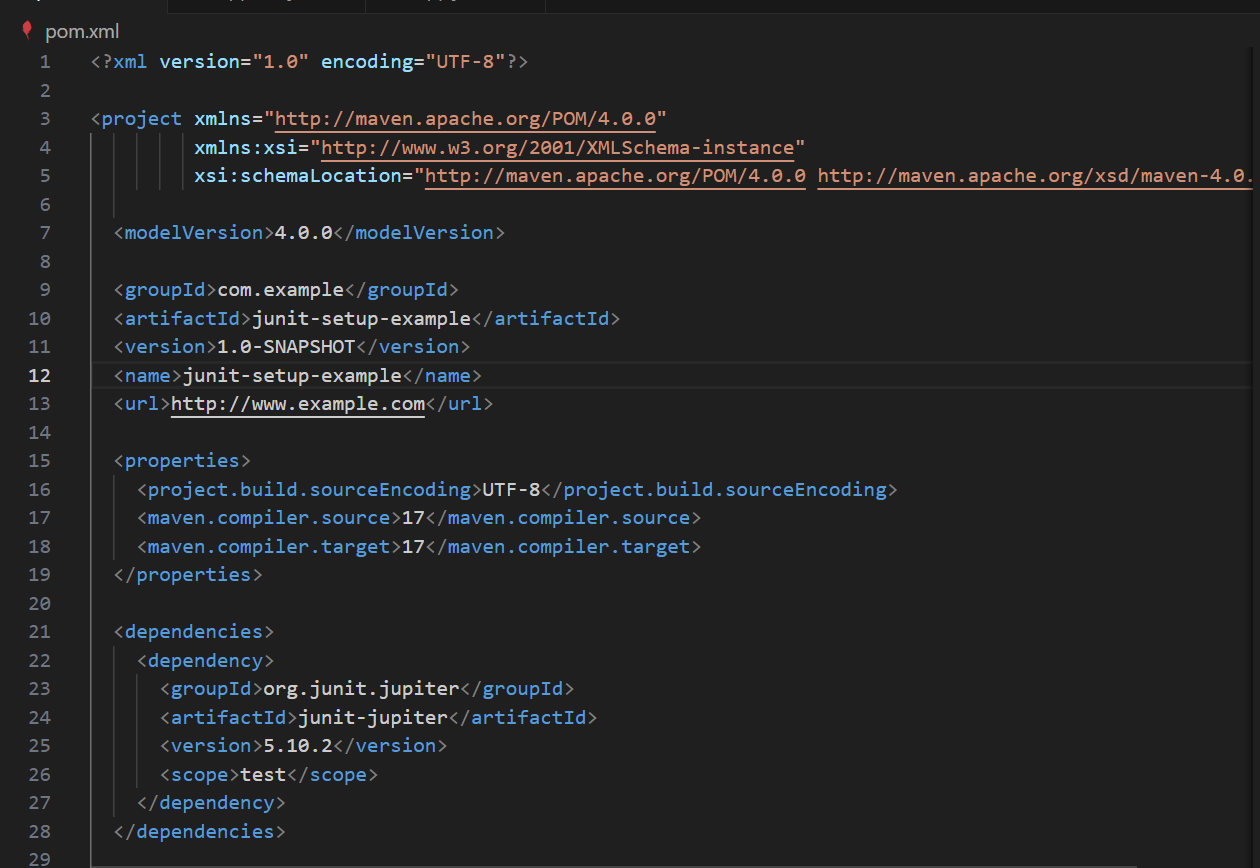
        <version>3.0.0-M7</version>

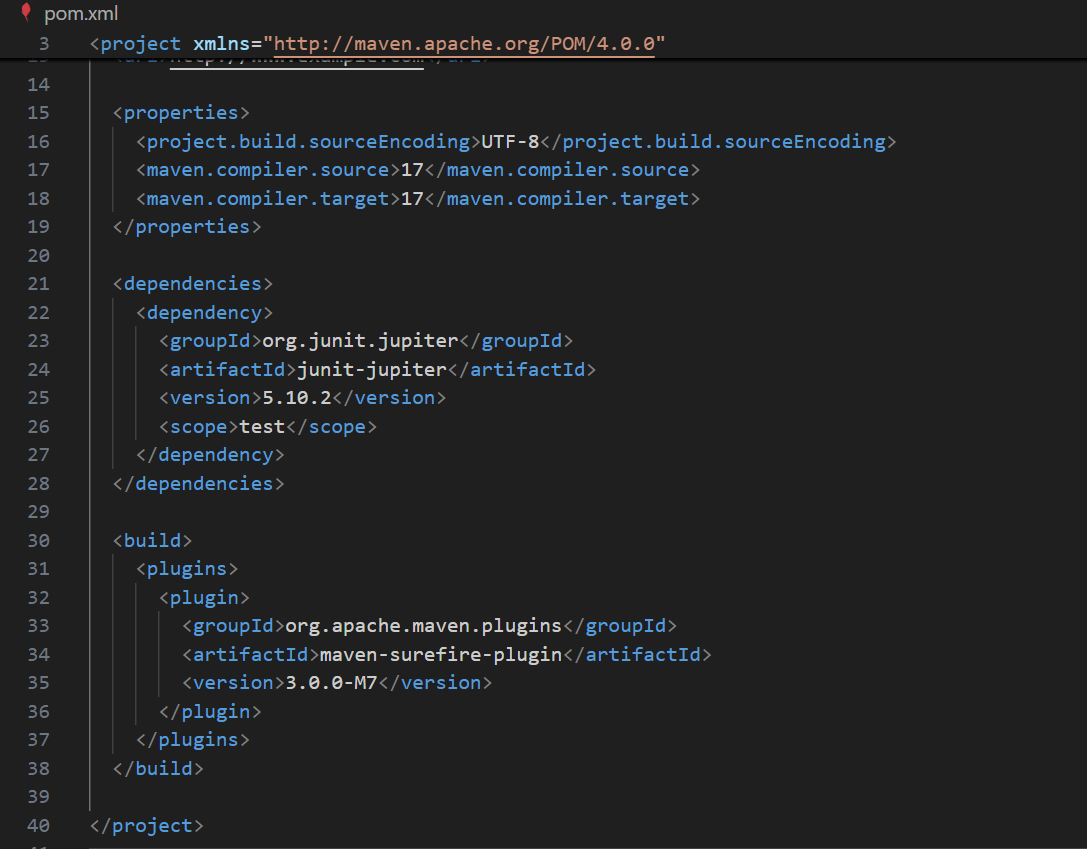
      </plugin>

    </plugins>

  </build>

</project>





4)AssertionsTest.java

package com.example;

import org.junit.jupiter.api.Test;

import static org.junit.jupiter.api.Assertions.\*;

public class AssertionsTest {

    @Test

    public void testAssertions() {

        assertEquals(5, 2 + 3);

        assertTrue(5 > 3);

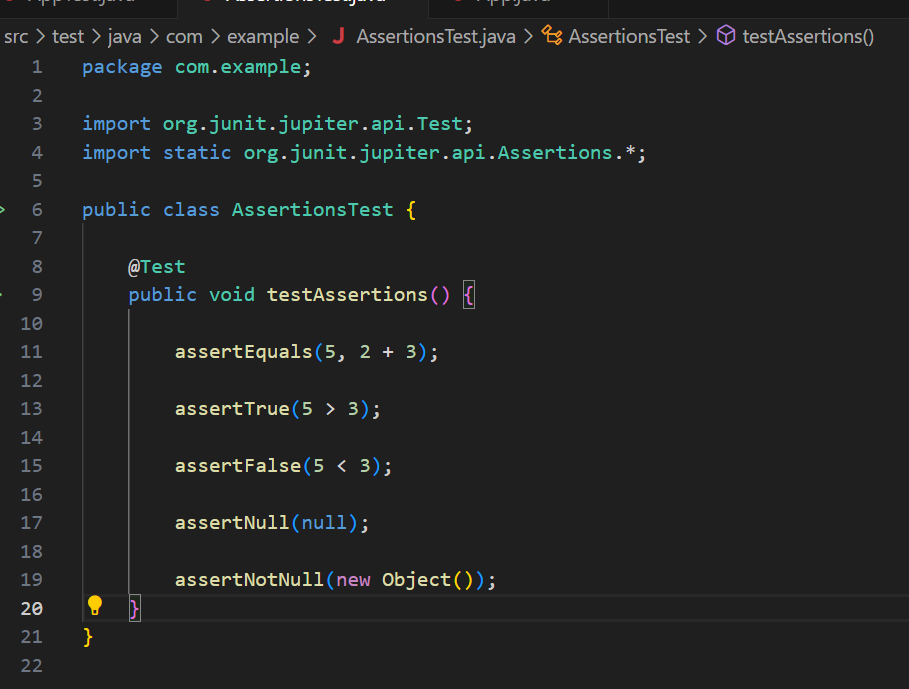
        assertFalse(5 < 3);

        assertNull(null);

        assertNotNull(new Object());

    }

}



THE EXTRA ADDED CODES ARE:

4) in main\java\com]example

(i) calculator.java:

package com.example;

public class Calculator {

    public int add(int a, int b) {

        return a + b;

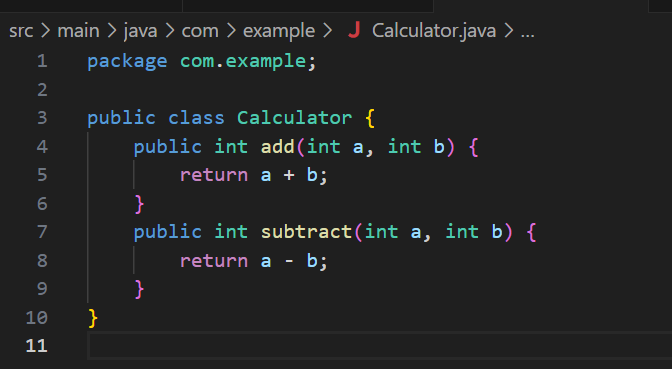
    }

    public int subtract(int a, int b) {

        return a - b;

    }

}



5) in test\java\com\example

(i)CalculatorTest.java:

package com.example;

import org.junit.jupiter.api.AfterEach;

import org.junit.jupiter.api.BeforeEach;

import org.junit.jupiter.api.Test;

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

public class CalculatorTest {

    private Calculator calculator;

    @BeforeEach

    public void setUp() {

        calculator = new Calculator();

        System.out.println("Setup: Calculator instance created.");

    }

    @AfterEach

    public void tearDown() {

        calculator = null;

        System.out.println("Teardown: Calculator instance cleared.");

    }

    @Test

    public void testAddition() {

        int a = 5;

        int b = 3;

        int result = calculator.add(a, b);

        assertEquals(8, result);

    }

    @Test

    public void testSubtraction() {

        int a = 10;

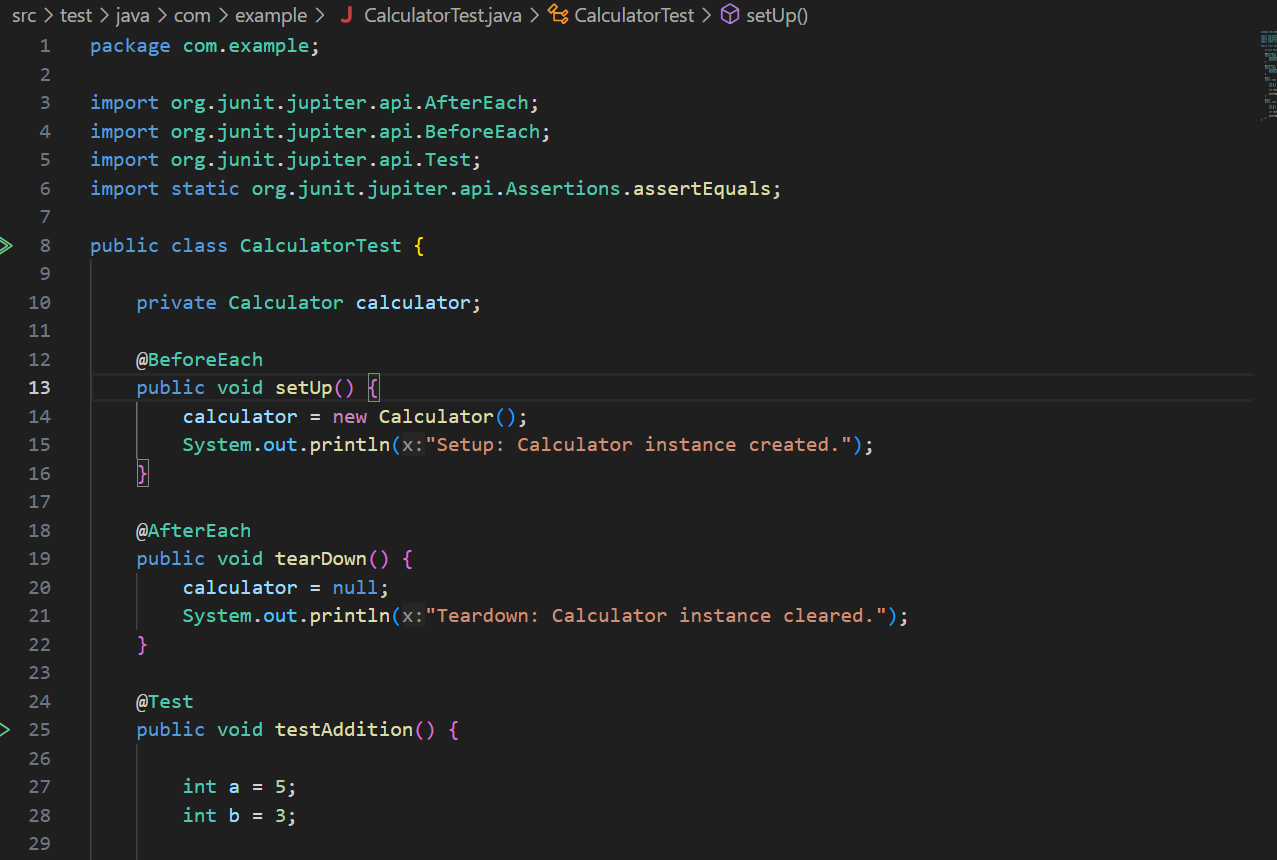
        int b = 4;

        int result = calculator.subtract(a, b);

        assertEquals(6, result);

    }

}



Output:

