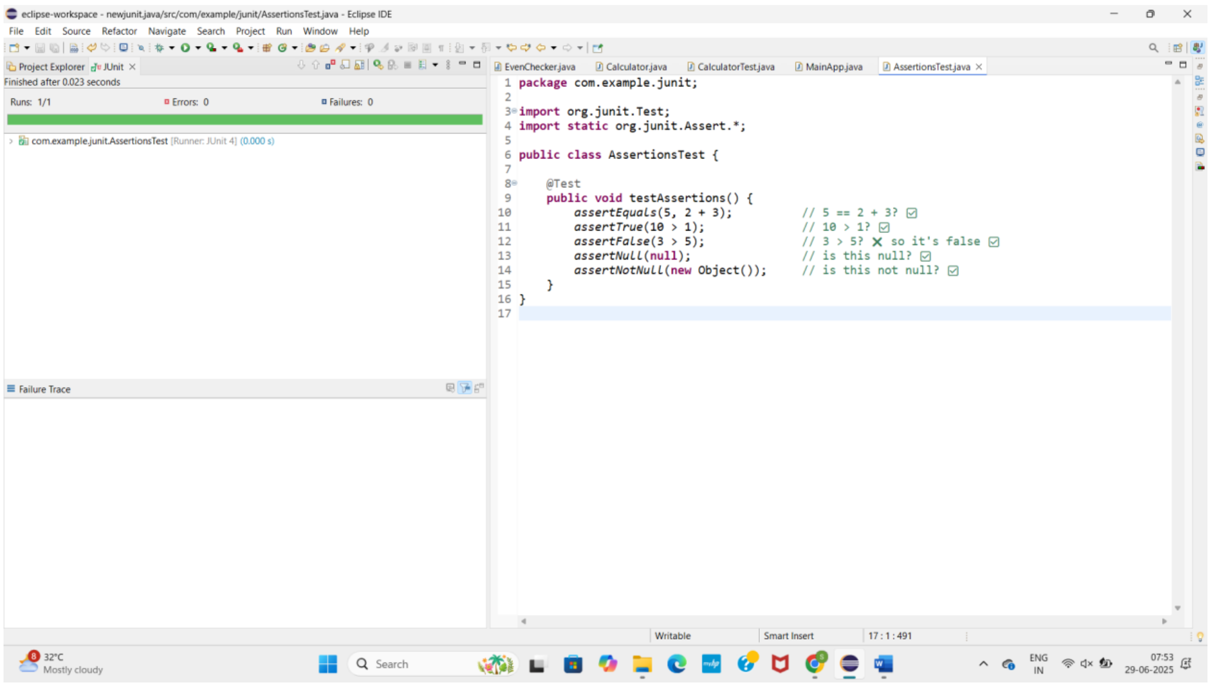
Basic JUnits Exercise

Exercise : Assertions in JUnit

```
package com.example.junit;


import org.junit.Test;

import static org.junit.Assert.*;


public class AssertionsTest {


    @Test

    public void testAssertions() {

        assertEquals(5, 2 + 3);

        assertTrue(10 > 1);

        assertFalse(3 > 5);

        assertNull(null);

        assertNotNull(new Object());

    }

}
```

OUTPUT:



eclipse-workspace - newjunit.java/src/com/example/junit/AssertionsTest.java - Eclipse IDE

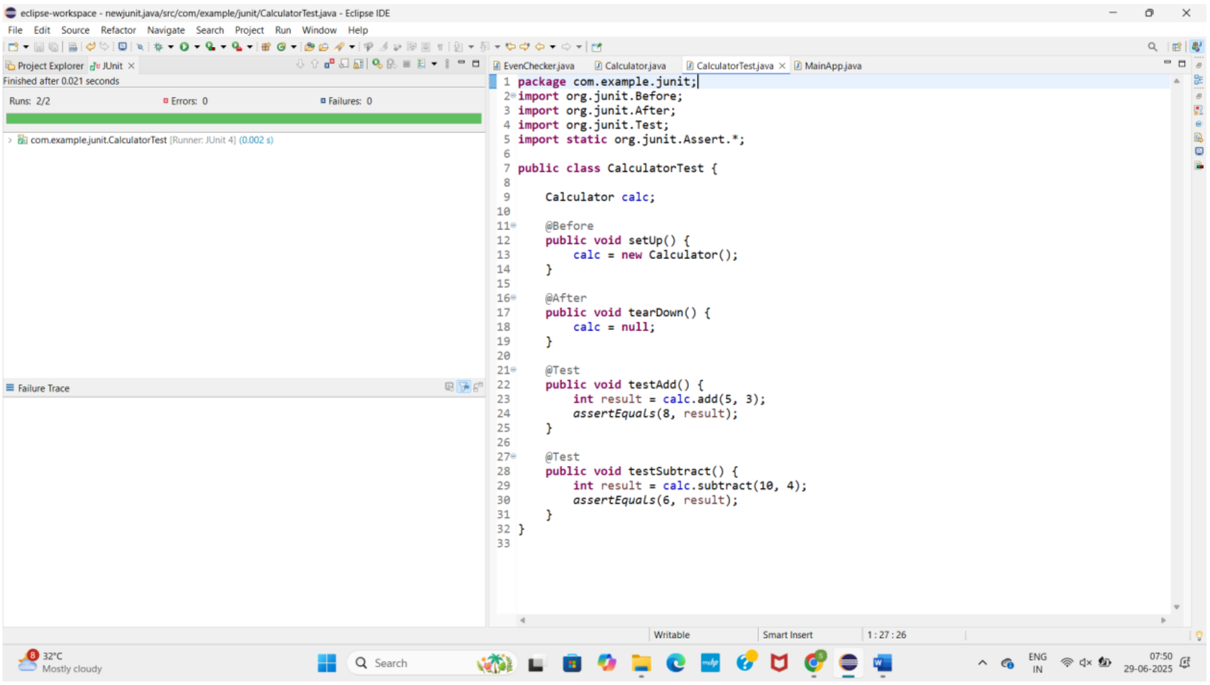
```
Finished after 0.023 seconds
Runs: 1/1        Errors: 0        Failures: 0
> com.example.junit.AssertionsTest [Runner: JUnit 4] (0.000 s)
```

```
package com.example.junit;

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

public class AssertionsTest {

    @Test
    public void testAssertions() {
        assertEquals(5, 2 + 3);           // 5 == 2 + 3? ✓
        assertTrue(10 > 1);               // 10 > 1? ✓
        assertFalse(3 > 5);               // 3 > 5? ✗ so it's false ✓
        assertNull(null);                 // is this null? ✓
        assertNotNull(new Object());      // is this not null? ✓
    }
}
```

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

```
package com.example.junit;

import org.junit.Before;

import org.junit.After;

import org.junit.Test;

import static org.junit.Assert.*;

public class CalculatorTest {

   Calculator calc;

   @Before

   public void setUp() {

      calc = new Calculator();

   }

   public void tearDown() {

      calc = null;

   }

   public void testAdd() {

      int a = 5, b = 3;

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

      assertEquals(8, result);

   }

   public void testSubtract() {

      int a = 10, b = 4;

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

      assertEquals(6, result);

   }

}
```

OUTPUT:



eclipse-workspace - newjunit.java/src/com/example/junit/CalculatorTest.java - Eclipse IDE

```
Finished after 0.021 seconds
Runs: 2/2        Errors: 0        Failures: 0
com.example.junit.CalculatorTest [Runner: JUnit 4] (0.002 s)
```

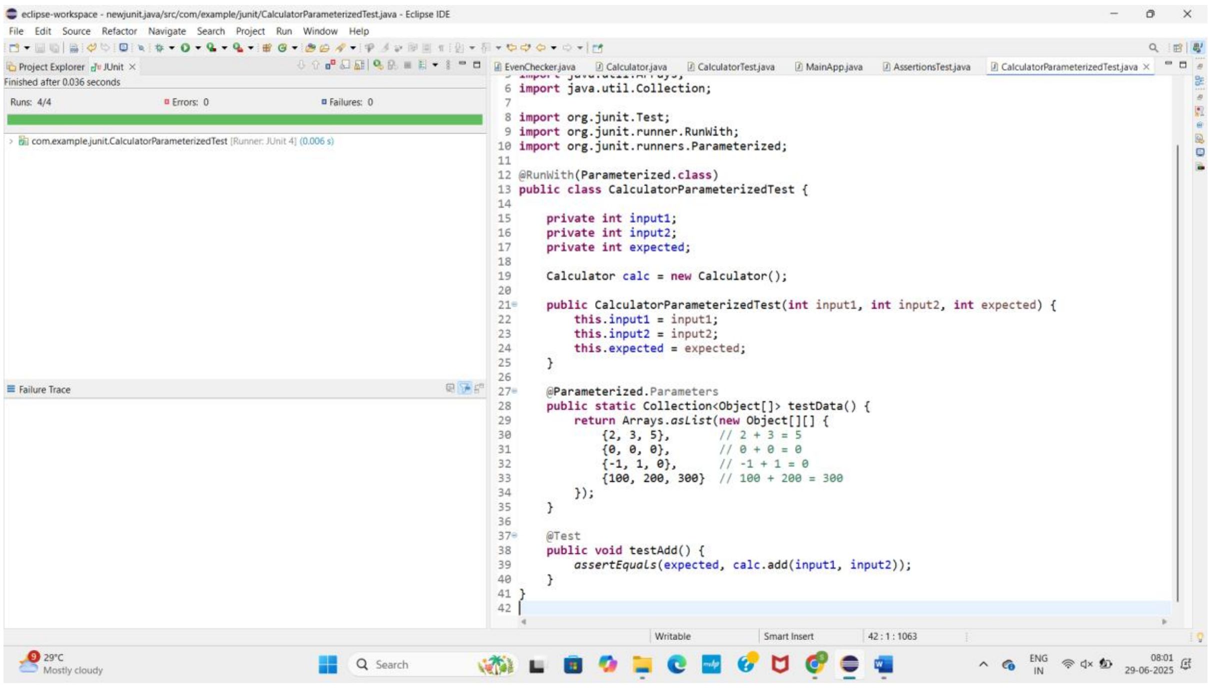
```java
package com.example.junit;
import org.junit.Before;
import org.junit.After;
import org.junit.Test;
import static org.junit.Assert.*;

public class CalculatorTest {

    Calculator calc;

    @Before
    public void setUp() {
        calc = new Calculator();
    }

    @After
    public void tearDown() {
        calc = null;
    }

    @Test
    public void testAdd() {
        int result = calc.add(5, 3);
        assertEquals(8, result);
    }

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

Exercise : Parameterized Tests

```java
package com.example.junit;

import static org.junit.Assert.assertEquals;

import java.util.Arrays;

import java.util.Collection;

import org.junit.Test;

import org.junit.runner.RunWith;

import org.junit.runners.Parameterized;

@RunWith(Parameterized.class)

public class CalculatorParameterizedTest {

   private int input1;

   private int input2;

   private int expected;

   Calculator calc = new Calculator();

   public CalculatorParameterizedTest(int input1, int input2, int expected) {

      this.input1 = input1;

      this.input2 = input2;

      this.expected = expected;

   }
```

```
    @Parameterized.Parameters
    public static Collection<Object[]> testData() {
        return Arrays.asList(new Object[][] {
            {2, 3, 5},       // 2 + 3 = 5
            {0, 0, 0},       // 0 + 0 = 0
            {-1, 1, 0},      // -1 + 1 = 0
            {100, 200, 300}  // 100 + 200 = 300
        });
    }
    @Test
    public void testAdd() {
        assertEquals(expected, calc.add(input1, input2));
    }
}
```

Output:



eclipse-workspace - newjunit.java/src/com/example/junit/CalculatorParameterizedTest.java - Eclipse IDE

```
Finished after 0.036 seconds
Runs: 4/4        Errors: 0        Failures: 0
com.example.junit.CalculatorParameterizedTest [Runner: JUnit 4] (0.006 s)
```

```
import java.util.Collection;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.junit.runners.Parameterized;

@RunWith(Parameterized.class)
public class CalculatorParameterizedTest {

    private int input1;
    private int input2;
    private int expected;

    Calculator calc = new Calculator();

    public CalculatorParameterizedTest(int input1, int input2, int expected) {
        this.input1 = input1;
        this.input2 = input2;
        this.expected = expected;
    }

    @Parameterized.Parameters
    public static Collection<Object[]> testData() {
        return Arrays.asList(new Object[][] {
            {2, 3, 5},       // 2 + 3 = 5
            {0, 0, 0},       // 0 + 0 = 0
            {-1, 1, 0},      // -1 + 1 = 0
            {100, 200, 300}  // 100 + 200 = 300
        });
    }

    @Test
    public void testAdd() {
        assertEquals(expected, calc.add(input1, input2));
    }
}
```