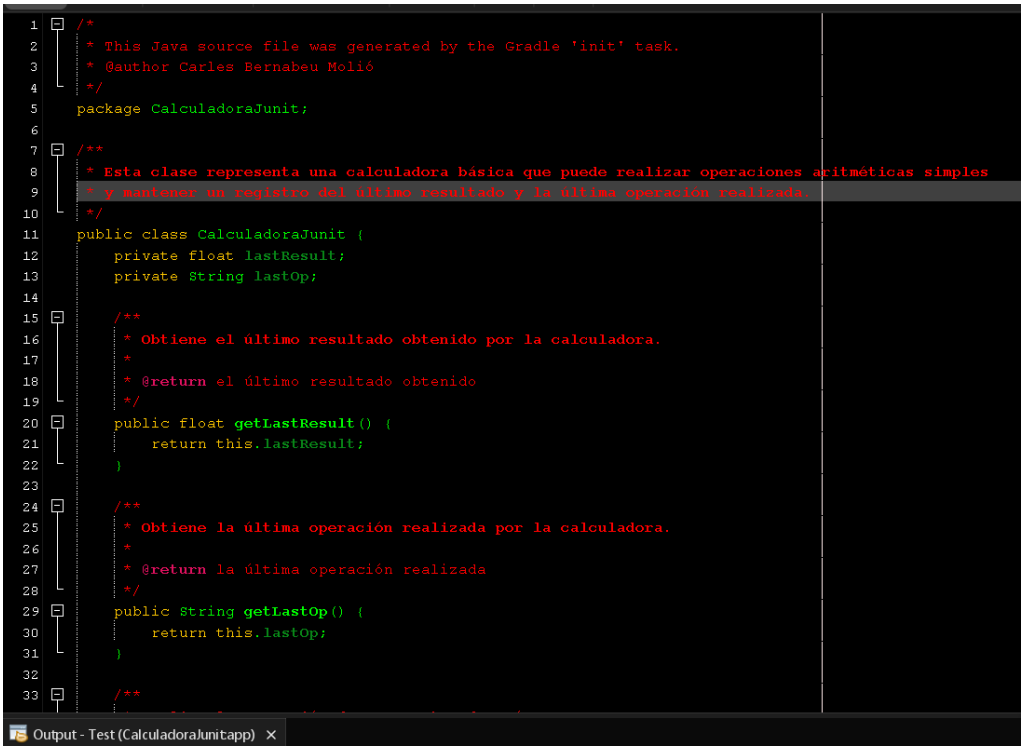
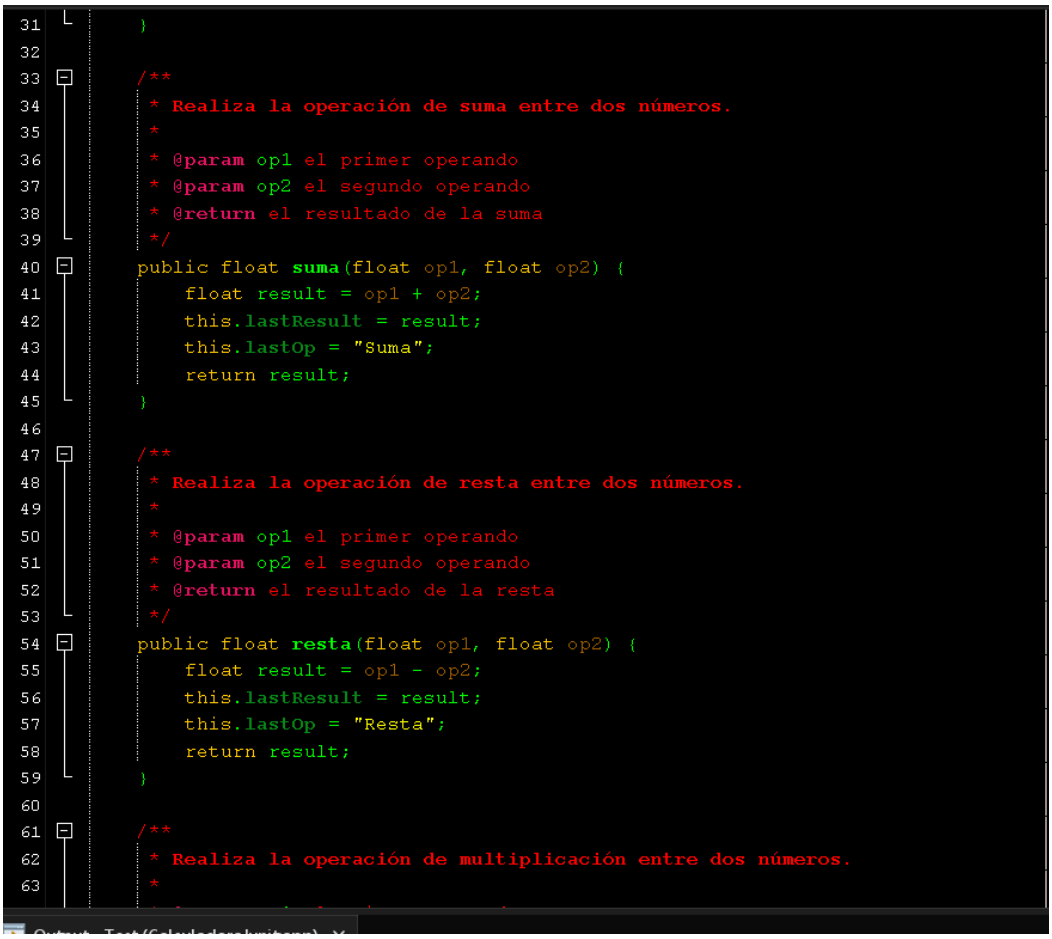
Carles Bernabeu Molió Pràctica: JUnit

PART 1: Captures de les funcions comentades.



```java
/*
 * This Java source file was generated by the Gradle 'init' task.
 * @author Carles Bernabeu Molió
 */
package CalculadoraJunit;

/**
 * Esta clase representa una calculadora básica que puede realizar operaciones aritméticas simples
 * y mantener un registro del último resultado y la última operación realizada.
 */
public class CalculadoraJunit {
    private float lastResult;
    private String lastOp;

    /**
     * Obtiene el último resultado obtenido por la calculadora.
     *
     * @return el último resultado obtenido
     */
    public float getLastResult() {
        return this.lastResult;
    }

    /**
     * Obtiene la última operación realizada por la calculadora.
     *
     * @return la última operación realizada
     */
    public String getLastOp() {
        return this.lastOp;
    }

    /**
```

Output - Test (CalculadoraJunit:app)

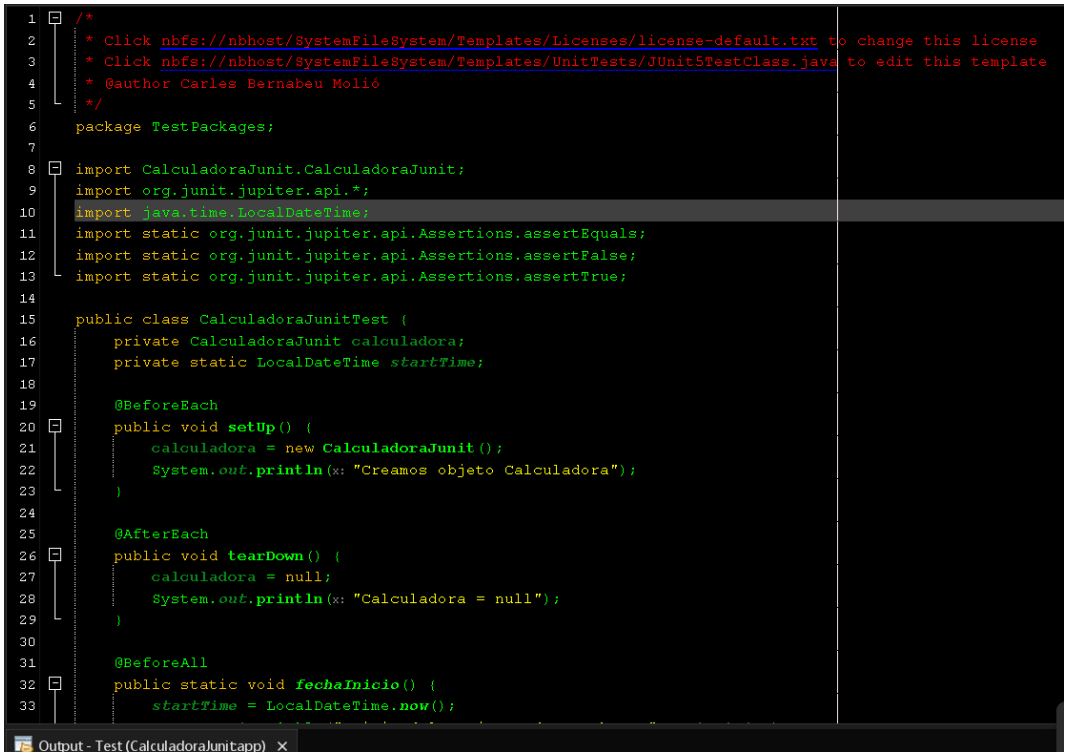


```java
    }

    /**
     * Realiza la operación de suma entre dos números.
     *
     * @param op1 el primer operando
     * @param op2 el segundo operando
     * @return el resultado de la suma
     */
    public float suma(float op1, float op2) {
        float result = op1 + op2;
        this.lastResult = result;
        this.lastOp = "Suma";
        return result;
    }

    /**
     * Realiza la operación de resta entre dos números.
     *
     * @param op1 el primer operando
     * @param op2 el segundo operando
     * @return el resultado de la resta
     */
    public float resta(float op1, float op2) {
        float result = op1 - op2;
        this.lastResult = result;
        this.lastOp = "Resta";
        return result;
    }

    /**
     * Realiza la operación de multiplicación entre dos números.
     *
```

```java
    /**
     * Realiza la operación de multiplicación entre dos números.
     *
     * @param op1 el primer operando
     * @param op2 el segundo operando
     * @return el resultado de la multiplicación
     */
    public float multiplica(float op1, float op2) {
        float result = op1 * op2;
        this.lastResult = result;
        this.lastOp = "Multiplica";
        return result;
    }

    /**
     * Realiza la operación de división entre dos números.
     *
     * @param op1 el dividendo
     * @param op2 el divisor
     * @return el resultado de la división
     */
    public float divideix(float op1, float op2) {
        float result = op1 / op2;
        this.lastResult = result;
        this.lastOp = "Divideix";
        return result;
    }

    /**
     * Comprueba si el primer número es mayor que el segundo.
     *
     * @param op1 el primer número
     * @param op2 el segundo número
```

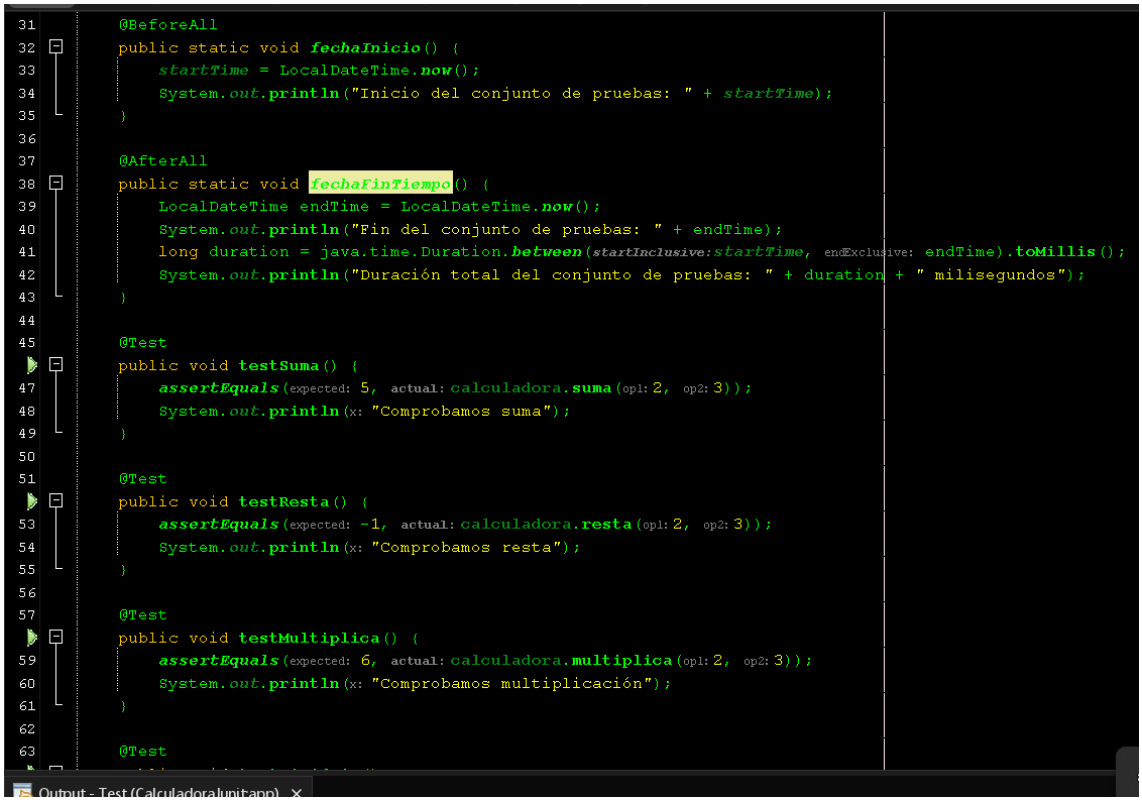
```java
        return result;
    }

    /**
     * Comprueba si el primer número es mayor que el segundo.
     *
     * @param op1 el primer número
     * @param op2 el segundo número
     * @return true si el primer número es mayor que el segundo, false en caso contrario
     */
    public boolean majorQue(float op1, float op2) {
        return op1 > op2;
    }

    /**
     * Restablece el último resultado y la última operación realizada a sus valores predeterminados.
     */
    public void restablecer() {
        this.lastResult = 0;
        this.lastOp = "Ninguna";
    }
}
```

PARTE 2: Captures de la classe de proves.



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/UnitTests/JUnit5TestClass.java to edit this template
 * @author Carles Bernabeu Molió
 */
package TestPackages;

import CalculadoraJunit.CalculadoraJunit;
import org.junit.jupiter.api.*;
import java.time.LocalDateTime;
import static org.junit.jupiter.api.Assertions.assertEquals;
import static org.junit.jupiter.api.Assertions.assertFalse;
import static org.junit.jupiter.api.Assertions.assertTrue;

public class CalculadoraJunitTest {
    private CalculadoraJunit calculadora;
    private static LocalDateTime startTime;

    @BeforeEach
    public void setUp() {
        calculadora = new CalculadoraJunit();
        System.out.println(x: "Creamos objeto Calculadora");
    }

    @AfterEach
    public void tearDown() {
        calculadora = null;
        System.out.println(x: "Calculadora = null");
    }

    @BeforeAll
    public static void fechaInicio() {
        startTime = LocalDateTime.now();
```

Output - Test (CalculadoraJunitapp)



```
    @BeforeAll
    public static void fechaInicio() {
        startTime = LocalDateTime.now();
        System.out.println("Inicio del conjunto de pruebas: " + startTime);
    }

    @AfterAll
    public static void fechaFinTiempo() {
        LocalDateTime endTime = LocalDateTime.now();
        System.out.println("Fin del conjunto de pruebas: " + endTime);
        long duration = java.time.Duration.between(startInclusive:startTime, endExclusive: endTime).toMillis();
        System.out.println("Duración total del conjunto de pruebas: " + duration + " milisegundos");
    }

    @Test
    public void testSuma() {
        assertEquals(expected: 5, actual: calculadora.suma(op1: 2, op2: 3));
        System.out.println(x: "Comprobamos suma");
    }

    @Test
    public void testResta() {
        assertEquals(expected: -1, actual: calculadora.resta(op1: 2, op2: 3));
        System.out.println(x: "Comprobamos resta");
    }

    @Test
    public void testMultiplica() {
        assertEquals(expected: 6, actual: calculadora.multiplica(op1: 2, op2: 3));
        System.out.println(x: "Comprobamos multiplicación");
    }

    @Test
```

Output - Test (CalculadoraJunitapp)

```java
    }

    @Test
    public void testDivideix() {
        assertEquals(expected: 2, actual: calculadora.divideix(op1: 6, op2: 3));
        System.out.println(x: "Comprobamos división");
    }

    @Test
    public void testMajorQue() {
        assertTrue(condition:calculadora.majorQue(op1: 5, op2: 3));
        assertFalse(condition:calculadora.majorQue(op1: 3, op2: 5));
        System.out.println(x: "Comprobamos mayor");
    }

    @Test
    public void testRestablir() {
        calculadora.resta(op1: 2, op2: 3);
        calculadora.restablecer();
        assertEquals(expected: 0, actual: calculadora.getLastResult());
        assertEquals(expected: "Ninguna", actual: calculadora.getLastOp());
        System.out.println(x: "Restablecemos calculadora");
    }
}


    // TODO add test methods here.
    // The methods must be annotated with annotation @Test. For example:
    //
    // @Test
    // public void hello() {}
```

PART 3: Captura de l'execució del test de JUnit.