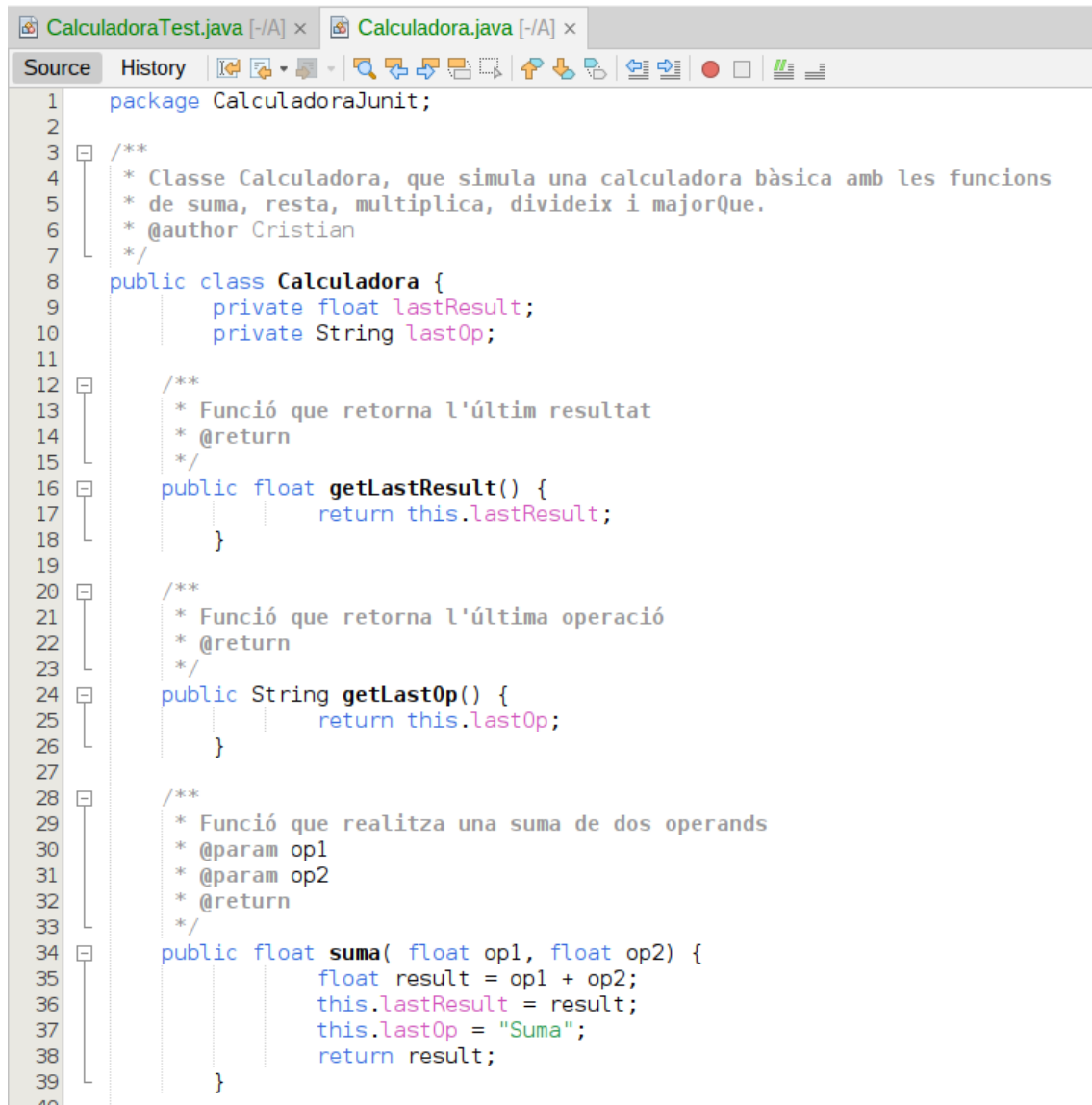


CAPTURES FUNCIONS DE LA CALCULADORA COMENTADES:



```
1 package CalculadoraJunit;
2
3 /**
4  * Classe Calculadora, que simula una calculadora bàsica amb les funcions
5  * de suma, resta, multiplica, divideix i majorQue.
6  * @author Cristian
7  */
8 public class Calculadora {
9     private float lastResult;
10    private String lastOp;
11
12    /**
13     * Funció que retorna l'últim resultat
14     * @return
15     */
16    public float getLastResult() {
17        return this.lastResult;
18    }
19
20    /**
21     * Funció que retorna l'última operació
22     * @return
23     */
24    public String getLastOp() {
25        return this.lastOp;
26    }
27
28    /**
29     * Funció que realitza una suma de dos operands
30     * @param op1
31     * @param op2
32     * @return
33     */
34    public float suma( float op1, float op2) {
35        float result = op1 + op2;
36        this.lastResult = result;
37        this.lastOp = "Suma";
38        return result;
39    }
40}
```

```

40
41
42  /**
43   * Funció que realitza una resta de dos operands
44   * @param op1
45   * @param op2
46   * @return
47   */
48  public float resta(float op1, float op2) {
49      float result = op1 - op2;
50      this.lastResult = result;
51      this.lastOp = "Resta";
52      return result;
53  }
54
55  /**
56   * Funció que realitza una multiplicació de dos operands
57   * @param op1
58   * @param op2
59   * @return
60   */
61  public float multiplica(float op1, float op2) {
62      float result = op1 * op2;
63      this.lastResult = result;
64      this.lastOp = "Multiplica";
65      return result;
66  }
67
68  /**
69   * Funció que realitza una divisió entre dos operands
70   * @param op1
71   * @param op2
72   * @return
73   */
74  public float divideix(float op1, float op2) {
75      float result = op1 / op2;
76      this.lastResult = result;
77      this.lastOp = "Divideix";
78      return result;
79  }

```

```
79
80
81 /**
82  * Funció que retorna si el primer operand és major que el segon
83  * @param op1
84  * @param op2
85  * @return
86  */
87 public boolean majorQue (float op1, float op2) {
88     if (op1 > op2) {
89         return true;
90     }
91     return false;
92 }
93
94 /**
95  * Funció que reestableix els valors inicials del resultat i l'operació
96  */
97 public void restablecer(){
98     this.lastResult = 0;
99     this.lastOp = "Ninguna";
100 }
101
102
103
104
```

CAPTURES DE LA CLASSE DE PROVES (CALCULADORATEST)

```
CalculadoraTest.java [-/A] x  Calculadora.java [-/A] x
Source History
1 package CalculadoraJUnit;
2
3 import java.time.LocalDateTime;
4 import java.time.format.DateTimeFormatter;
5 import org.junit.jupiter.api.AfterEach;
6 import org.junit.jupiter.api.AfterAll;
7 import static org.junit.jupiter.api.Assertions.*;
8 import org.junit.jupiter.api.BeforeEach;
9 import org.junit.jupiter.api.BeforeAll;
10 import org.junit.jupiter.api.Test;
11
12
13 /**
14  * Classe per testar el correcte funcionament de les funcions de la calculadora
15  * @author Cristian
16  */
17 public class CalculadoraTest {
18
19     Calculadora calculadora;
20
21
22     public CalculadoraTest() {
23     }
24
25     @BeforeAll
26     public static void setUpClass() {
27         LocalDateTime hora = LocalDateTime.now();
28         DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern: "HH:mm:ss.SSS");
29         String horaFormatejada = hora.format(formatter);
30         System.out.println(horaFormatejada+"---->Inici del test");
31     }
32
33     @AfterAll
34     public static void tearDownClass() {
35         LocalDateTime hora = LocalDateTime.now();
36         DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern: "HH:mm:ss.SSS");
37         String horaFormatejada = hora.format(formatter);
38         System.out.println(horaFormatejada+"---->Fi del test");
39     }
40
```

```

40
41
42 @BeforeEach
43 public void setUp() {
44     LocalDateTime hora = LocalDateTime.now();
45     DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern: "HH:mm:ss.SSS");
46     String horaFormatejada = hora.format(formatter);
47     System.out.println(horaFormatejada+"---->Inici de la prova");
48     calculadora=new Calculadora();
49 }
50
51 @AfterEach
52 public void tearDown() {
53     LocalDateTime hora = LocalDateTime.now();
54     DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern: "HH:mm:ss.SSS");
55     String horaFormatejada = hora.format(formatter);
56     System.out.println(horaFormatejada+"---->Fi de la prova");
57     calculadora.restablecer();
58 }
59
60 @Test
61 public void testSuma(){
62     System.out.println(x: "\tPROVA SUMA");
63     float op1=2.0f;
64     float op2=3.0f;
65     float resultat;
66     resultat = calculadora.suma(op1,op2);
67     float esperat= 5.0f;
68     assertEquals(expected:esperat, actual: resultat);
69 }

```

```

69
70 @Test
71 public void testResta(){
72     System.out.println(x: "\tPROVA RESTA");
73     float op1=2.0f;
74     float op2=3.0f;
75     float resultat;
76     resultat = calculadora.resta(op1,op2);
77     float esperat= -1.0f;
78     assertEquals(expected:esperat, actual: resultat);
79 }
80
81 @Test
82 public void testMultiplica(){
83     System.out.println(x: "\tPROVA MULTIPLICA");
84     float op1=2.0f;
85     float op2=3.0f;
86     float resultat;
87     resultat = calculadora.multiplica(op1,op2);
88     float esperat= 6.0f;
89     assertEquals(expected:esperat, actual: resultat);
90 }
91
92 @Test
93 public void testDivideix(){
94     System.out.println(x: "\tPROVA DIVIDEIX");
95     float op1=3.0f;
96     float op2=2.0f;
97     float resultat;
98     resultat = calculadora.divideix(op1,op2);
99     float esperat= 1.5f;
100     assertEquals(expected:esperat, actual: resultat);
101 }

```

```

102
103     @Test
104     public void testMajorQue(){
105         System.out.println(x: "\tPROVA MAJOR QUE");
106         float op1=2.0f;
107         float op2=3.0f;
108         boolean resultat;
109         resultat = calculadora.majorQue(op1,op2);
110         boolean esperat= false;
111         assertEquals(expected:esperat, actual: resultat);
112     }
113 }
114

```

CAPTURA DE LA EXECUCIÓ DEL TEST JUNIT

The screenshot shows an IDE with two tabs: `CalculadoraTest.java` and `Calculadora.java`. The `CalculadoraTest.java` tab is active, showing the following code:

```

16  */
17  public class CalculadoraTest {
18
19      Calculadora calculadora;
20
21
22      public CalculadoraTest() {
23      }
24
25      @BeforeAll
26      public static void setUpClass() {
27          LocalTime hora = LocalTime.now();
28          DateTimeFormatter formatter = DateTimeFormatter.ofPattern(pattern: "HH:mm:ss.SSS");
29          String horaFormatejada = hora.format(formatter);
30          System.out.println(horaFormatejada+"---->Inici del test");
31      }
32
33      @AfterAll

```

Below the code editor, the `Test Results` tab is active, showing the output of the test execution. The output is as follows:

```

Gradle Test Run :app:test started x
Tests passed: 100,00 %
All 5 tests passed. (0,183 s)
✓ CalculadoraJunit.CalculadoraTest passed
  ✓ testResta() passed (0,042 s)
  ✓ testSuma() passed (0,016 s)
  ✓ testMultiplica() passed (0,022 s)
  ✓ testMajorQue() passed (0,004 s)
  ✓ testDivideix() passed (0,004 s)
23:39:17.324---->Inici del test
23:39:17.364---->Inici de la prova
PROVA RESTA
23:39:17.389---->Fi de la prova
23:39:17.412---->Inici de la prova
PROVA SUMA
23:39:17.420---->Fi de la prova
23:39:17.440---->Inici de la prova
PROVA MULTIPLICA
23:39:17.445---->Fi de la prova
23:39:17.450---->Inici de la prova
PROVA MAJOR QUE
23:39:17.452---->Fi de la prova
23:39:17.458---->Inici de la prova
PROVA DIVIDEIX
23:39:17.459---->Fi de la prova
23:39:17.469---->Fi del test

```