

Ws40LibraryUnitTest

jueves, 2 de febrero de 2023 8:45

<https://docs.google.com/document/d/1mtklSDB6cFbZ-VK1RVxagF2U12Pmyha/edit?usp=sharing&oid=111686028395648835270&rtpof=true&sd=true>

```
...va | SimpleCalculator.java | FrmAdd.java | BasicOperationTest.java | TaxTest.java
Source | History | [Icons]

34
35 /**
36  * Test of add method, of class BasicOperation.
37  */
38 @Test
39 public void testAdd() {
40     System.out.println("add");
41     float addend1 = 1.00F;
42     float addent2 = 2.00F;
43     float expResult = 3.00F;
44     float result = BasicOperation.add(addend1, addent2);
45     assertEquals("expected: expResult, actual: result, delta: 0");
46
47     addend1 = 1.2F;
48     addent2 = 2.4F;
49     expResult = 3.6F;
50     result = BasicOperation.add(addend1, addent2);
51     assertEquals("expected: expResult, actual: result, delta: 0");
52 }
53
54
55 /**
56  * Test of add method, of class BasicOperation.
57  */
58 @Test
59 public void testAddNegatives() {
60     System.out.println("add");
61     float addend1 = -56.88F;
62     float addent2 = -235.00F;
63     float expResult = -291.88F;
64     float result = BasicOperation.add(addend1, addent2);
65     assertEquals("expected: expResult, actual: result, delta: 0");
66
67 }
68
69 /**
```

ec.edu.espe.calculator.controller.BasicOperationTest × ec.edu.espe.calculator.controller.TaxT

Tests passed: 80,00 %

- testAdd Failed: expected:<3.6> but was:<3.6000001>
expected:<3.6> but was:<3.6000001>
junit.framework.AssertionFailedError
at ec.edu.espe.calculator.controller.BasicOperationTest.testAdd(BasicOperationTe
at java.base/jdk.internal.reflect.DirectMethodHandleAccessor.invoke(DirectMetho
- testAddNegatives passed (0,001 s)
- testSubstract passed (0,0 s)
- testDivide passed (0,0 s)
- testMultiply passed (0,0 s)

```

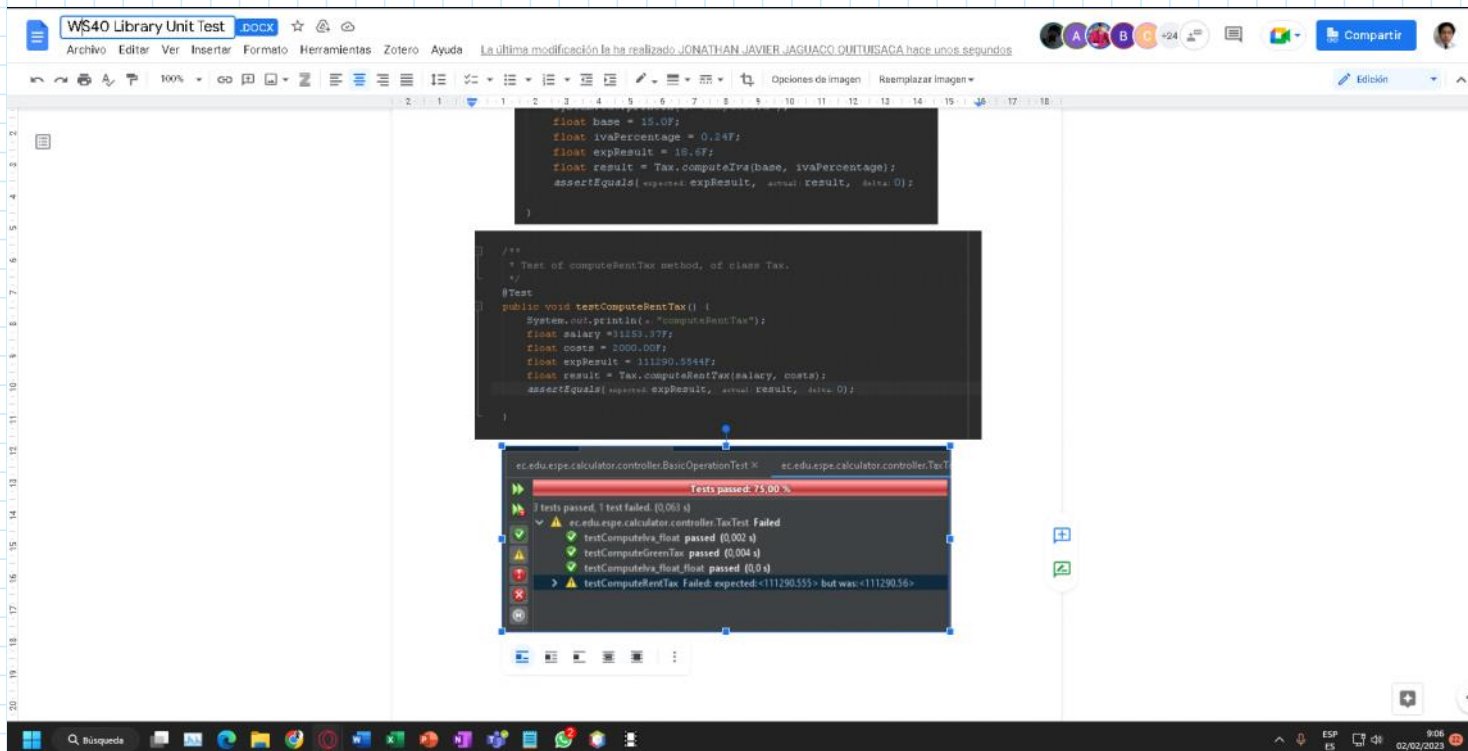
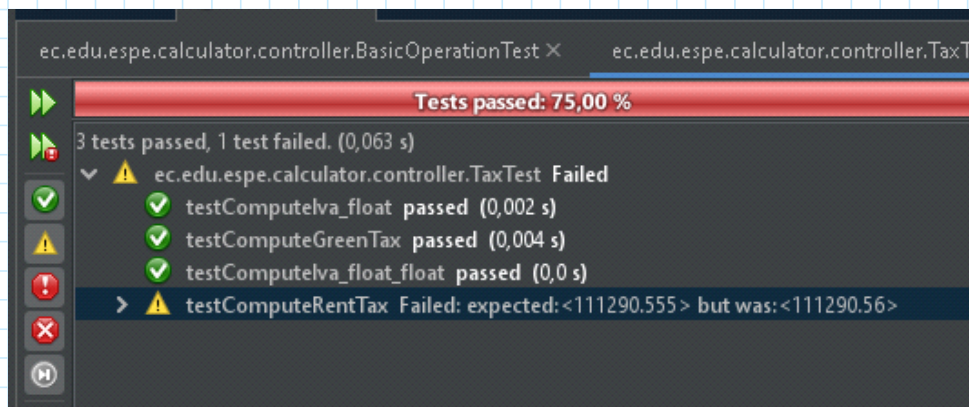
@Test
public void testComputeIva_float_float() {
    System.out.println("computeIva");
    float base = 15.0F;
    float ivaPercentage = 0.24F;
    float expectedResult = 18.6F;
    float result = Tax.computeIva(base, ivaPercentage);
    assertEquals(expected: expectedResult, actual: result, delta: 0);
}

```

```

/**
 * Test of computeRentTax method, of class Tax.
 */
@Test
public void testComputeRentTax() {
    System.out.println("computeRentTax");
    float salary = 31253.37F;
    float costs = 2000.00F;
    float expectedResult = 111290.5544F;
    float result = Tax.computeRentTax(salary, costs);
    assertEquals(expected: expectedResult, actual: result, delta: 0);
}

```



```

public class Tax {

    static final DecimalFormat df = new DecimalFormat("0.00");

    /*public static float computeIva(float base, float ivaPercentage) {
        float iva;
        iva = base * ivaPercentage;
        return iva;
    }*/

    public static double computeIva(double base) {
        double iva = 0;
        iva = base * 0.12F;
        df.format(iva);
        return iva;
    }
}

```

```

public static double computeRentTax(float monthlySalary, float costs) {

    double value;
    double iessContribution = (monthlySalary * 0.0945F) * 12;
    value = (monthlySalary * 12) - iessContribution - costs;

    if (value <= 11310.00F) {
        value = value * 0;
    }
    if (value > 11310.00F && value <= 14410.00F) {
        value = (value - 11310.00F) * 0.05F;
    }
    if (value > 14410.00F && value <= 18010.00F) {
        value = ((value - 14410.00F) * 0.10F) + 155.00F;
    }
    if (value > 18010.00F && value <= 21630.00F) {
        value = ((value - 18010.00F) * 0.12F) + 515.00F;
    }
    if (value > 21630.00F && value <= 31630.00F) {
        value = ((value - 21630.00F) * 0.15F) + 949.40F;
    }
    if (value > 31630.00F && value <= 41630.00F) {
        value = ((value - 31630.00F) * 0.20F) + 2449.40F;
    }
    if (value > 41630.00F && value <= 51630.00F) {
        value = ((value - 41630.00F) * 0.25F) + 4449.40F;
    }
    if (value > 51630.00F && value <= 61630.00F) {
        value = ((value - 51630.00F) * 0.30F) + 6949.40F;
    }
    if (value > 61630.00F && value <= 100000.00F) {
        value = ((value - 61630.00F) * 0.38F) + 9949.40F;
    }
    if (value > 100000.00F) {
        value = ((value - 100000.00F) * 0.37F) + 23376.88F;
    }

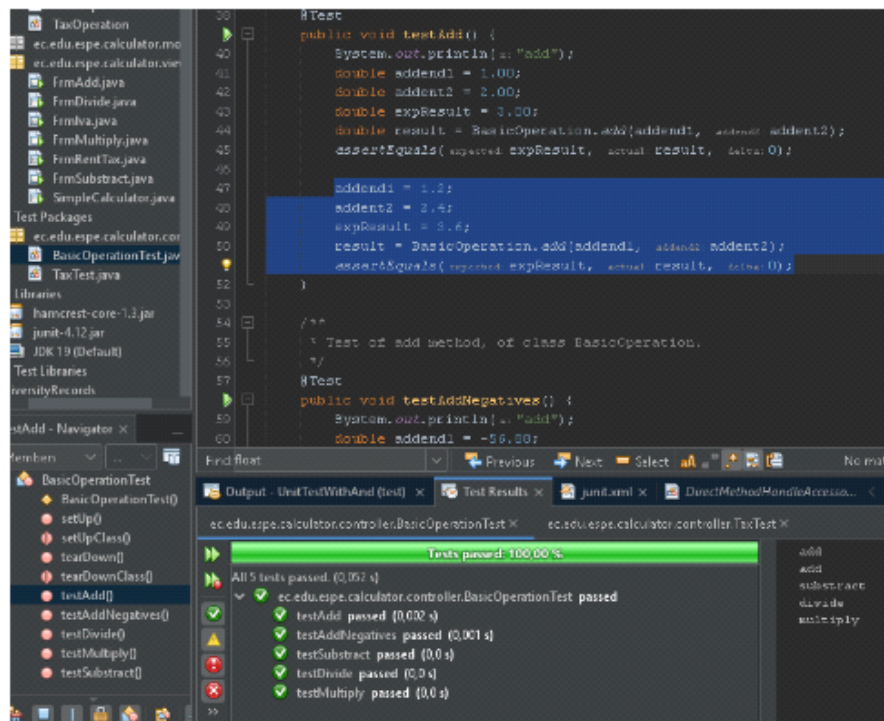
    df.format(value);

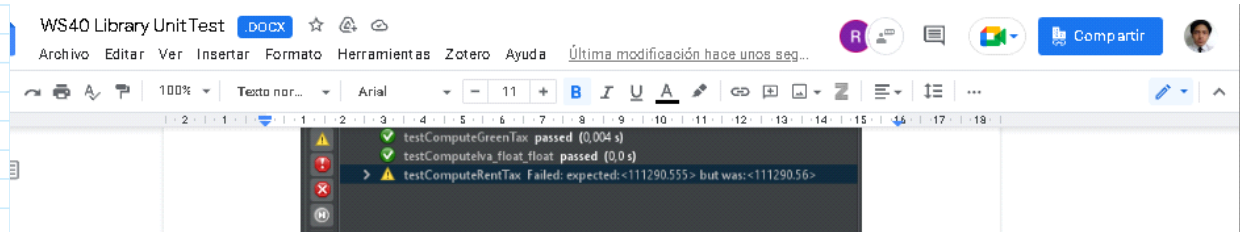
    return value;
}

```

Calculator|Tests Passed (Limiting and rounding to 4 decimal places with Math.round(). and using double)

```
public class BasicOperation {  
  
    public static double add(double addend1, double addend2) {  
        double sum;  
        sum = addend1 + addend2;  
        sum = (double) Math.round(sum * 1000d) / 1000d;  
        return sum;  
    }  
}
```





Taxes Tests Passed (Limiting and rounding to 4 decimal places with `Math.round()`, and using double)

