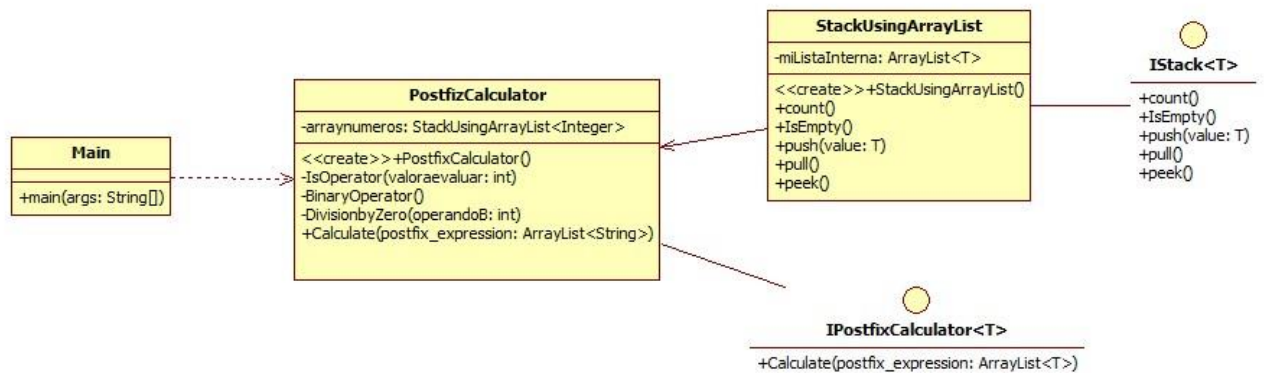


Hoja de Trabajo 2

Link de repositorio de Git: <https://github.com/sharlisbg/HDT2>

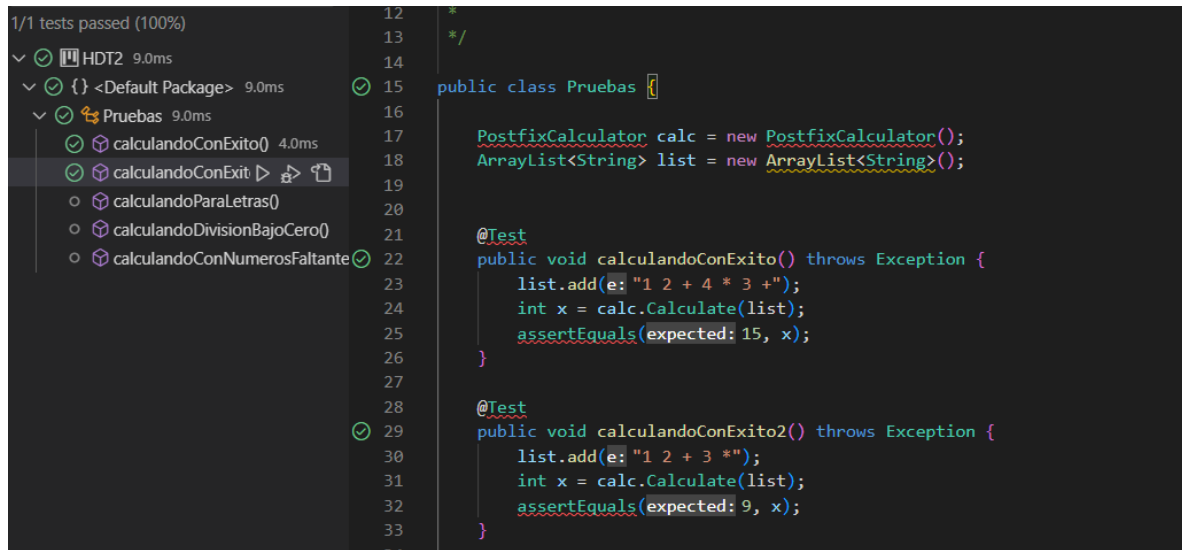
Diagrama UML:



Evidencias del testeo de métodos con la herramienta de VSCode:

Las pruebas que se realizaron son las siguientes y se visualizan de la siguiente forma:

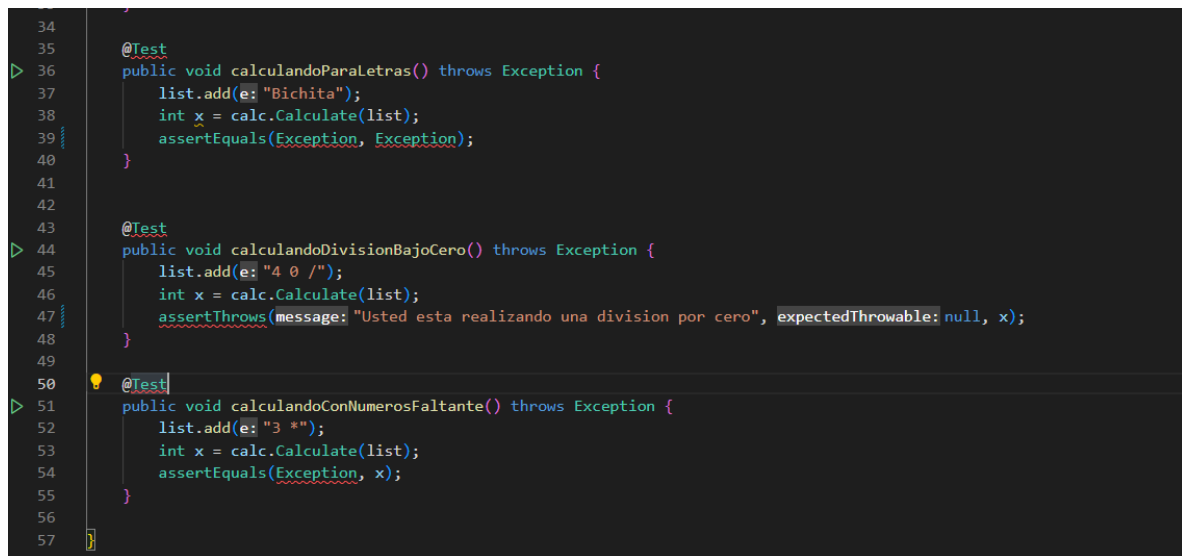
Se prueban que dos operaciones devuelvan los valores esperados



The screenshot shows an IDE with a test runner on the left and Java code on the right. The test runner indicates that 1/1 tests passed (100%) and lists several test methods, including 'calculandoConExito()' which is currently selected. The Java code defines a 'Pruebas' class with two test methods: 'calculandoConExito()' and 'calculandoConExito2()'. Both methods use a 'PostfixCalculator' instance to calculate the result of a postfix expression stored in an 'ArrayList' and then use 'assertEquals' to verify the result.

```
12  *
13  */
14
15  public class Pruebas {
16
17      PostfixCalculator calc = new PostfixCalculator();
18      ArrayList<String> list = new ArrayList<String>();
19
20
21
22      @Test
23      public void calculandoConExito() throws Exception {
24          list.add(e: "1 2 + 4 * 3 +");
25          int x = calc.Calculate(list);
26          assertEquals(expected: 15, x);
27      }
28
29      @Test
30      public void calculandoConExito2() throws Exception {
31          list.add(e: "1 2 + 3 *");
32          int x = calc.Calculate(list);
33          assertEquals(expected: 9, x);
34      }
35  }
```

Se hacen operaciones como divisibles en 0, con menos de 2 operadores o con letras, con la finalidad de tener una Exception.



The screenshot shows a portion of the Java code from the previous image, specifically the test methods 'calculandoParaLetras()', 'calculandoDivisionBajoCero()', and 'calculandoConNumerosFaltante()'. These methods test the calculator's behavior with invalid inputs: a string 'Bichita', a division by zero ('4 0 /'), and an expression with fewer than two operators ('3 *'). Each test method uses 'assertEquals' or 'assertThrows' to verify the expected outcome, which is an 'Exception'.

```
34
35
36  @Test
37  public void calculandoParaLetras() throws Exception {
38      list.add(e: "Bichita");
39      int x = calc.Calculate(list);
40      assertEquals(Exception, Exception);
41  }
42
43
44  @Test
45  public void calculandoDivisionBajoCero() throws Exception {
46      list.add(e: "4 0 /");
47      int x = calc.Calculate(list);
48      assertThrows(message: "Usted esta realizando una division por cero", expectedThrowable: null, x);
49  }
50
51  @Test
52  public void calculandoConNumerosFaltante() throws Exception {
53      list.add(e: "3 *");
54      int x = calc.Calculate(list);
55      assertEquals(Exception, x);
56  }
57
58  }
```

Evidencias de instanciar el objeto Calculadora con todas sus funcionalidades para que el usuario interactúe:

```
HojaTrabajoDos - -zsh - 80x24
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos % javac Principal.java
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos % java Principal
Ingrese el numero de la expresion que desea evaluar
1. 1 2 + 3 *
2. 3 4 -
3. 4 0 /
4. 4 2 /
5. 3 4 * 5 +
6. 3 *
6
La operacion es binaria y no hay suficientes operandos
La respuesta a la operacion es 3
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos %
```

```
HojaTrabajoDos - -zsh - 80x24
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos % javac Principal.java
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos % java Principal
Ingrese el numero de la expresion que desea evaluar
1. 1 2 + 3 *
2. 3 4 -
3. 4 0 /
4. 4 2 /
5. 3 4 * 5 +
5
La respuesta a la operacion es 17
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos %
```

```
HojaTrabajoDos - -zsh - 80x24
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos % javac Principal.java
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos % java Principal
Ingrese el numero de la expresion que desea evaluar
1. 1 2 + 3 *
2. 3 4 -
3. 4 0 /
4. 4 2 /
3
Usted esta realizando una división por cero
Index 0 out of bounds for length 0
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos %
```

```
HojaTrabajoDos - -zsh - 80x24
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos % javac Principal.java
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos % java Principal
Ingrese el numero de la expresion que desea evaluar
1. 1 2 + 3 *
2. 3 4 -
3. 4 0 /
4. 4 2 /
2
La respuesta a la operacion es -1
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos %
```

```
HojaTrabajoDos - -zsh - 80x24
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos % javac Principal.java
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos % java Principal
Ingrese el numero de la expresion que desea evaluar
1. 1 2 + 3 *
2. 3 4 -
3. 4 0 /
4. 4 2 /
4
La respuesta a la operacion es 2
leonelcontreras@Leonels-MacBook-Pro HojaTrabajoDos %
```