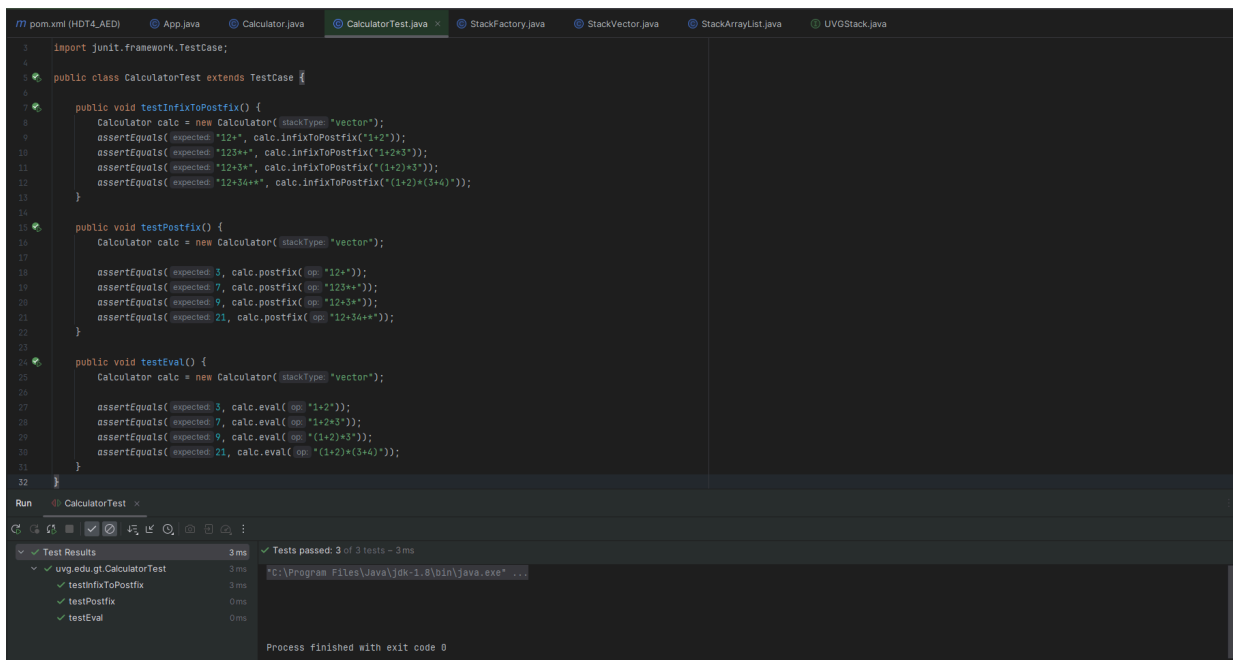


Unit Tests de Calculator



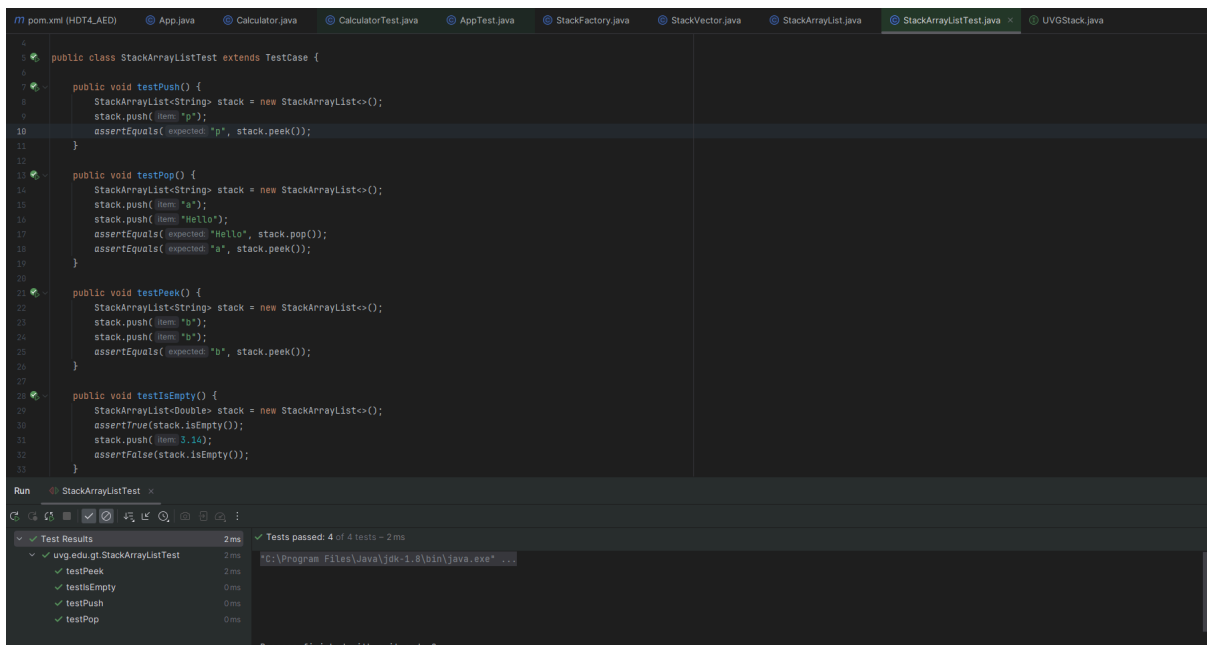
```
1 import junit.framework.TestCase;
2
3 public class CalculatorTest extends TestCase {
4
5     public void testInfixToPostfix() {
6         Calculator calc = new Calculator(stackType: "vector");
7         assertEquals(expected: "12+", calc.infixToPostfix("1+2"));
8         assertEquals(expected: "123+*", calc.infixToPostfix("1+2+3"));
9         assertEquals(expected: "12+3*", calc.infixToPostfix("(1+2)+3"));
10        assertEquals(expected: "12+34+*", calc.infixToPostfix("(1+2)+(3+4)"));
11    }
12
13    public void testPostfix() {
14        Calculator calc = new Calculator(stackType: "vector");
15
16        assertEquals(expected: 3, calc.postfix(op: "12+"));
17        assertEquals(expected: 7, calc.postfix(op: "123+*"));
18        assertEquals(expected: 9, calc.postfix(op: "12+3*"));
19        assertEquals(expected: 21, calc.postfix(op: "12+34+*"));
20    }
21
22    public void testEval() {
23        Calculator calc = new Calculator(stackType: "vector");
24
25        assertEquals(expected: 3, calc.eval(op: "1+2"));
26        assertEquals(expected: 7, calc.eval(op: "1+2+3"));
27        assertEquals(expected: 9, calc.eval(op: "(1+2)+3"));
28        assertEquals(expected: 21, calc.eval(op: "(1+2)+(3+4)"));
29    }
30 }
31
32 }
```

Run CalculatorTest

Test Results	3ms	Tests passed: 3 of 3 tests - 3ms
uvv.edu.gt.CalculatorTest	3ms	C:\Program Files\Java\jdk-1.8\bin\java.exe ...
testInfixToPostfix	3ms	
testPostfix	0ms	
testEval	0ms	

Process finished with exit code 0

Unit Tests de StackArrayList



```
1 public class StackArrayListTest extends TestCase {
2
3     public void testPush() {
4         StackArrayList<String> stack = new StackArrayList<>();
5         stack.push(item: "p");
6         assertEquals(expected: "p", stack.peek());
7     }
8
9     public void testPop() {
10        StackArrayList<String> stack = new StackArrayList<>();
11        stack.push(item: "a");
12        stack.push(item: "Hello");
13        assertEquals(expected: "Hello", stack.pop());
14        assertEquals(expected: "a", stack.peek());
15    }
16
17    public void testPeek() {
18        StackArrayList<String> stack = new StackArrayList<>();
19        stack.push(item: "b");
20        stack.push(item: "b");
21        assertEquals(expected: "b", stack.peek());
22    }
23
24    public void testIsEmpty() {
25        StackArrayList<Double> stack = new StackArrayList<>();
26        assertTrue(stack.isEmpty());
27        stack.push(item: 3.14);
28        assertFalse(stack.isEmpty());
29    }
30 }
31
32 }
```

Run StackArrayListTest

Test Results	2ms	Tests passed: 4 of 4 tests - 2ms
uvv.edu.gt.StackArrayListTest	2ms	C:\Program Files\Java\jdk-1.8\bin\java.exe ...
testPeek	2ms	
testIsEmpty	0ms	
testPush	0ms	
testPop	0ms	

Process finished with exit code 0

Unit Tests de StackVector

```
1 public class StackVectorTest extends TestCase {
2
3     public void testPush() {
4         StackVector<String> stack = new StackVector<>();
5         stack.push( item: "p");
6         assertEquals( expected: "p", stack.peek());
7     }
8
9     public void testPop() {
10        StackVector<String> stack = new StackVector<>();
11        stack.push( item: "a");
12        stack.push( item: "Hello");
13        assertEquals( expected: "Hello", stack.pop());
14        assertEquals( expected: "a", stack.peek());
15    }
16
17    public void testPeek() {
18        StackVector<String> stack = new StackVector<>();
19        stack.push( item: "b");
20        stack.push( item: "b");
21        assertEquals( expected: "b", stack.peek());
22    }
23
24    public void testIsEmpty() {
25        StackVector<Double> stack = new StackVector<>();
26        assertTrue( stack.isEmpty());
27        stack.push( item: 3.14);
28        assertFalse( stack.isEmpty());
29    }
30 }
```

Run StackVectorTest

Test Results 3ms Tests passed: 4 of 4 tests - 3ms

- uvg.edu.gt.StackVectorTest 3ms
- testPush 3ms
- testPop 0ms
- testPeek 0ms
- testIsEmpty 0ms
- testPush 0ms

Unit Tests de StackLL

```
4 public class StackLLTest extends TestCase {
5
6     public void testPush() {
7         StackLL<String> stack = new StackLL<>( type: "single");
8         stack.push( item: "p");
9         assertEquals( expected: "p", stack.peek());
10    }
11
12    public void testPop() {
13        StackLL<String> stack = new StackLL<>( type: "double");
14        stack.push( item: "a");
15        stack.push( item: "Hello");
16        assertEquals( expected: "Hello", stack.pop());
17        assertEquals( expected: "a", stack.peek());
18    }
19
20    public void testPeek() {
21        StackLL<String> stack = new StackLL<>( type: "double");
22        stack.push( item: "b");
23        stack.push( item: "b");
24        assertEquals( expected: "b", stack.peek());
25    }
26
27    public void testIsEmpty() {
28        StackLL<Double> stack = new StackLL<>( type: "double");
29        assertTrue( stack.isEmpty());
30        stack.push( item: 3.14);
31        assertFalse( stack.isEmpty());
32    }
33 }
```

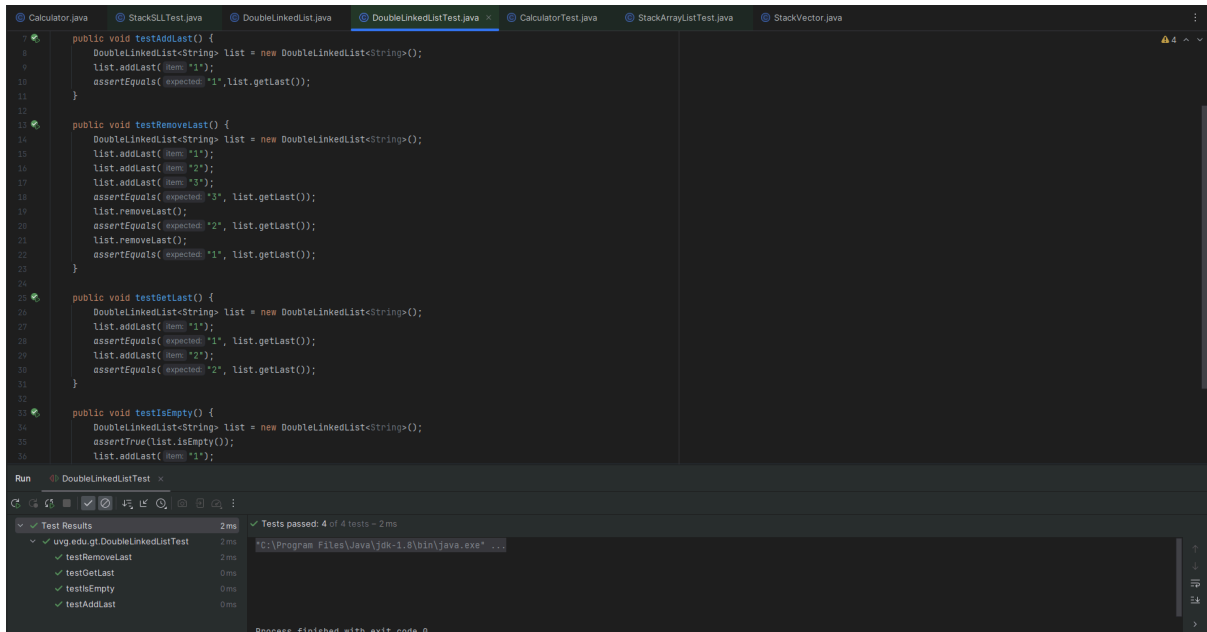
Run StackLLTest

Test Results 4ms Tests passed: 4 of 4 tests - 4ms

- uvg.edu.gt.StackLLTest 4ms
- testPush 3ms
- testPop 1ms
- testPeek 0ms
- testIsEmpty 0ms

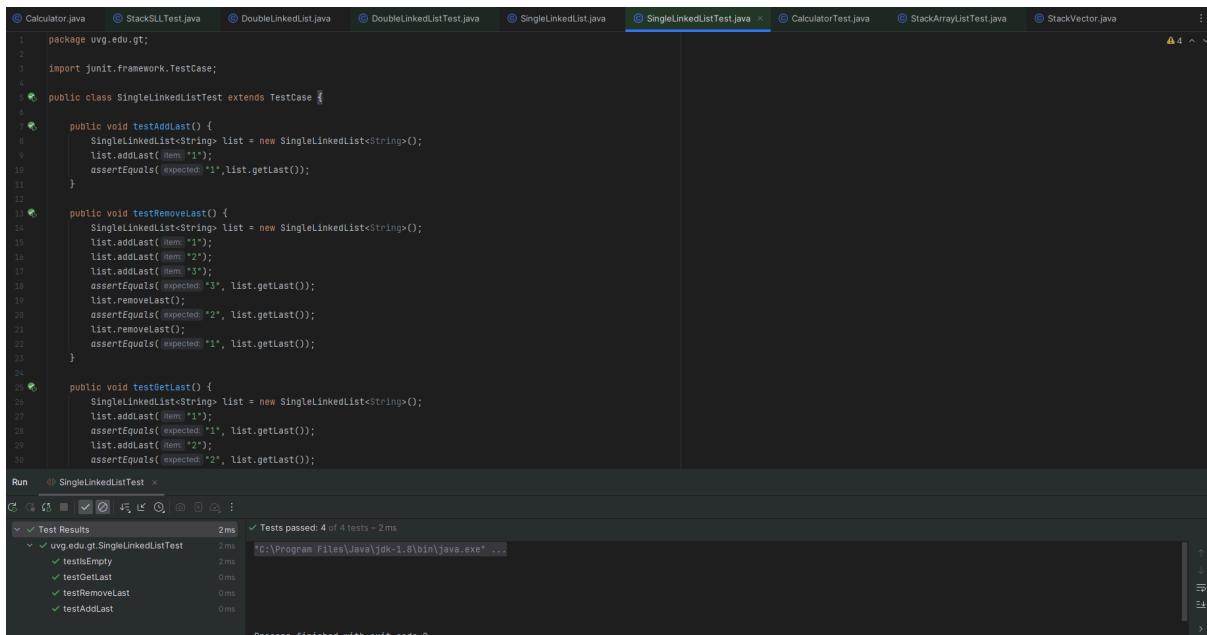
Process finished with exit code 0

Unit Test de Double Linked List



```
7 public void testAddLast() {
8     DoubleLinkedList<String> list = new DoubleLinkedList<String>();
9     list.addLast("1");
10    assertEquals("1", list.getLast());
11 }
12
13 public void testRemoveLast() {
14     DoubleLinkedList<String> list = new DoubleLinkedList<String>();
15     list.addLast("1");
16     list.addLast("2");
17     list.addLast("3");
18     assertEquals("3", list.getLast());
19     list.removeLast();
20     assertEquals("2", list.getLast());
21     list.removeLast();
22     assertEquals("1", list.getLast());
23 }
24
25 public void testGetLast() {
26     DoubleLinkedList<String> list = new DoubleLinkedList<String>();
27     list.addLast("1");
28     assertEquals("1", list.getLast());
29     list.addLast("2");
30     assertEquals("2", list.getLast());
31 }
32
33 public void testIsEmpty() {
34     DoubleLinkedList<String> list = new DoubleLinkedList<String>();
35     assertTrue(list.isEmpty());
36     list.addLast("1");
37 }
38
39 Run DoubleLinkedListTest
40
41 Test Results 2ms Tests passed: 4 of 4 tests - 2ms
42
43   ✓ uvw.edu.gt.DoubleLinkedListTest 2ms
44     ✓ testRemoveLast 2ms
45     ✓ testGetLast 0ms
46     ✓ testIsEmpty 0ms
47     ✓ testAddLast 0ms
48
49 Process finished with exit code 0
```

Unit Test de Single Linked List



```
1 package uvw.edu.gt;
2
3 import junit.framework.TestCase;
4
5 public class SingleLinkedListTest extends TestCase {
6
7     public void testAddLast() {
8         SingleLinkedList<String> list = new SingleLinkedList<String>();
9         list.addLast("1");
10        assertEquals("1", list.getLast());
11    }
12
13    public void testRemoveLast() {
14        SingleLinkedList<String> list = new SingleLinkedList<String>();
15        list.addLast("1");
16        list.addLast("2");
17        list.addLast("3");
18        assertEquals("3", list.getLast());
19        list.removeLast();
20        assertEquals("2", list.getLast());
21        list.removeLast();
22        assertEquals("1", list.getLast());
23    }
24
25    public void testGetLast() {
26        SingleLinkedList<String> list = new SingleLinkedList<String>();
27        list.addLast("1");
28        assertEquals("1", list.getLast());
29        list.addLast("2");
30        assertEquals("2", list.getLast());
31    }
32
33    public void testIsEmpty() {
34        SingleLinkedList<String> list = new SingleLinkedList<String>();
35        assertTrue(list.isEmpty());
36        list.addLast("1");
37    }
38
39 }
40
41 Run SingleLinkedListTest
42
43 Test Results 2ms Tests passed: 4 of 4 tests - 2ms
44
45   ✓ uvw.edu.gt.SingleLinkedListTest 2ms
46     ✓ testIsEmpty 2ms
47     ✓ testGetLast 0ms
48     ✓ testRemoveLast 0ms
49     ✓ testAddLast 0ms
50
51 Process finished with exit code 0
```