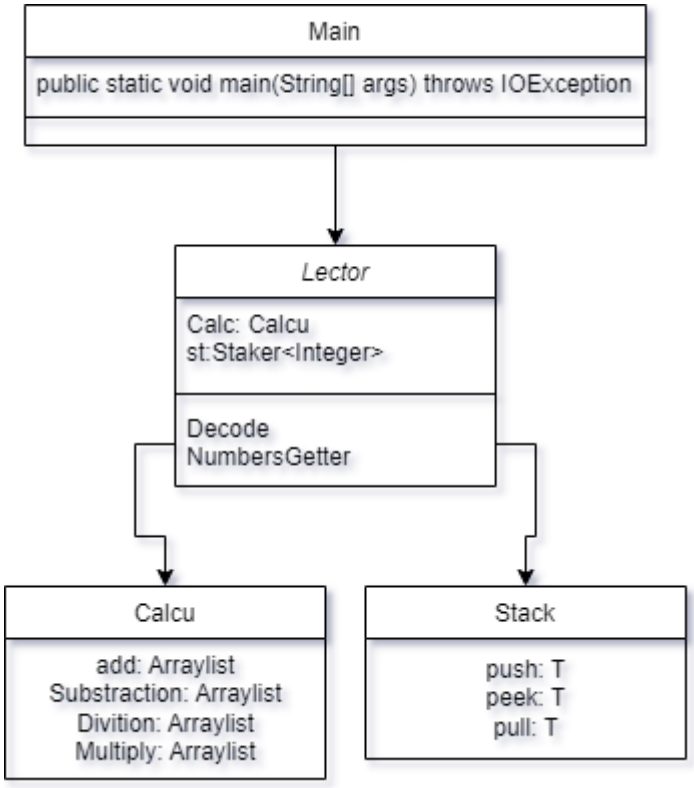
UML



Junit Test

```java
@Test
public void testAdd() {
    System.out.println(x: "add");
    ArrayList<Integer> num = new ArrayList<Integer>();
    num.add(e: 3);
    num.add(e: 5);
    Operator instance = new Operator();
    String expResult = "8";
    String result = instance.add(num);
    assertEquals(expResult, result);
}

@Test
public void testQuit() {
    System.out.println(x: "quit");
    ArrayList<Integer> num = new ArrayList<Integer>();
    num.add(e: 3);
    num.add(e: 5);
    Operator instance = new Operator();
    String expResult = "2";
    String result = instance.quit(num);
    assertEquals(expResult, result);
}

@Test
public void testDiv() {
    System.out.println(x: "div");
    ArrayList<Integer> num = new ArrayList<Integer>();
    num.add(e: 2);
    num.add(e: 6);
    Operator instance = new Operator();
    String expResult = "3";
    String result = instance.div(num);
    assertEquals(expResult, result);
}
```

```java
@Test
public void testMulti() {
    System.out.println(x: "multi");
    ArrayList<Integer> num = new ArrayList<Integer>();
    num.add(e: 3);
    num.add(e: 5);
    Operator instance = new Operator();
    String expResult = "15";
    String result = instance.multi(num);
    assertEquals(expResult, result);
}
```

```java
public void testDecodeExpresion() {
    System.out.println(x: "DecodeExpresion");
    String line = "3 3 3 3 3 *";
    Decoder instance = new Decoder();
    String expResult = "243";
    String result = instance.DecodeExpresion(line);
    assertEquals(expResult, result);
}
```