

# **DOCUMENTATIE**

## **Tema 1**

NUME STUDENT: Todorean Razvan-Florin  
GRUPA: 30222

# CUPRINS

1.	Obiectivul temei.....	3
2.	Analiza problemei, modelare, scenarii, cazuri de utilizare .....	3
3.	Proiectare.....	4
4.	Implementare.....	4
5.	Rezultate.....	5
6.	Concluzii .....	6
7.	Bibliografie.....	6

## 1. Obiectivul temei

*Implementa unui calculator cu operatii pe polinoame.*

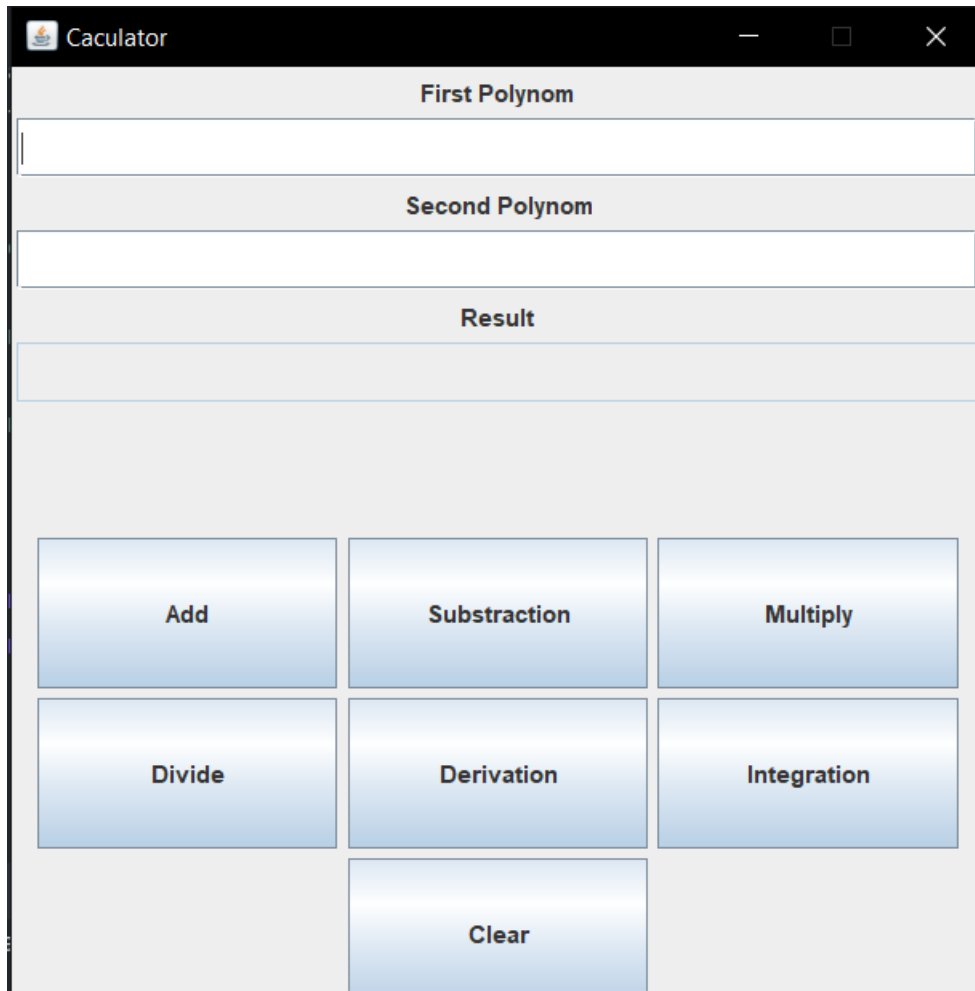
*Obiective secundare:*

*Intelegerea folosirii structurii de date Map.*

*Folosirea expresii regulate ca metoda de transpunere in Map.*

*Familiarizarea cu Junit.*

## 2. Analiza problemei, modelare, scenarii, cazuri de utilizare



In primul si in al doilea TextField se va introduce Polinomul dupa urmatorul exemplu:

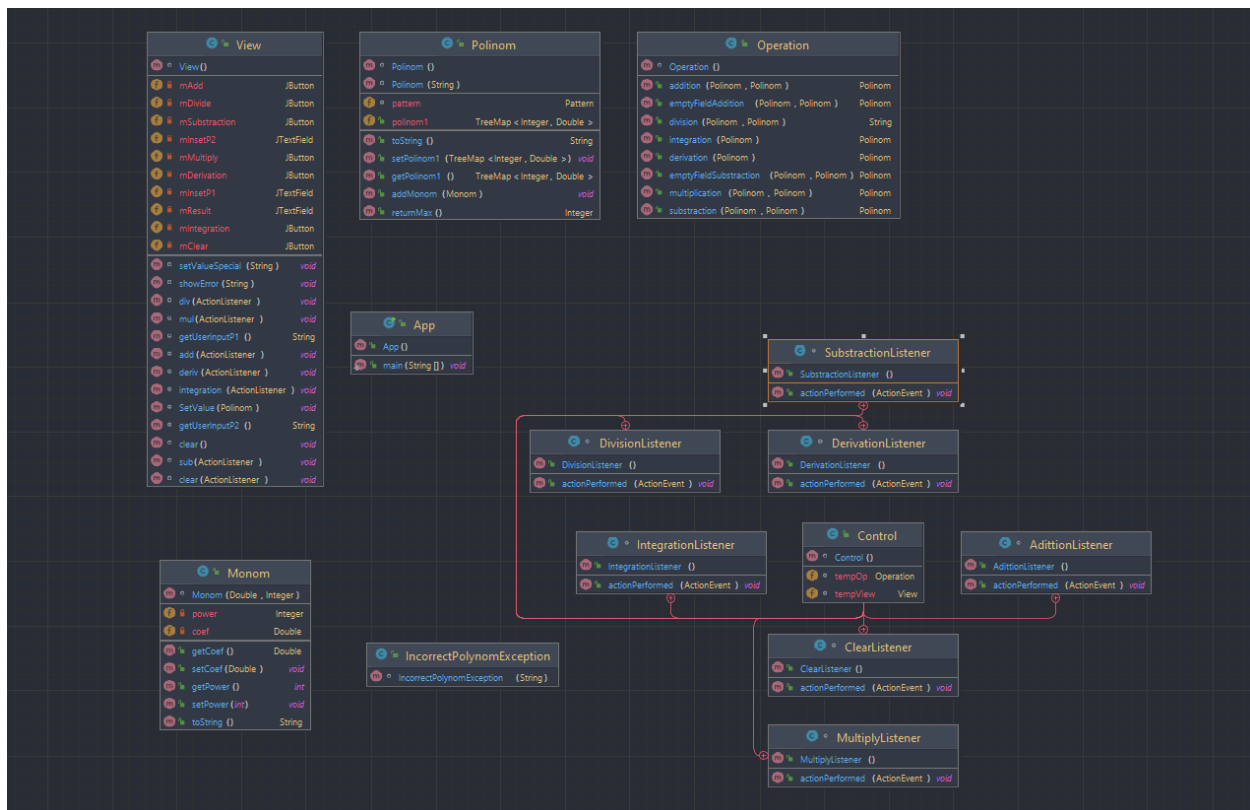
“ $2x^3-4x^4...$ ” fara spatii intre semnele dintre polinoame

Am implementat urmatoarele operatii:

1. Adunare : in care iterez cele doua polinoame si adun coeficientii de la polinoamele cu exponentul egal, iar daca nu se regaeste un monom care sa aiba o putere comuna cu un altul din al doilea polinom il va scrie in rezultat.
2. Scadere: o aplic la fel ca si adunarea.
3. Inmultirea: Am folosit doua iterari in care inmultesc fiecare monom din primul polinom cu fiecare din al doilea polinom.
4. Impartirea:

- i. Am verificat daca unul din cei doi polinomi sunt nuli astfel arunc o exceptie, la fel si daca gradul numitorului este mai mare decat cel al numaratorului.
  - ii. Imi asignez un nou polinom , valorile numaratorului
  - iii. In bucla care merge cat timp acel polinom nu este gol ,iar gradul numaratorului este mai mare decat al numitorului.
  - iv. Imi asignez restul facand o inmultire cu coeficientul resultant din impartirea facuta si coeficientul monomului de la numitor iar cheia va creste cu puterea rezultata de la impartirea facuta anterior.
5. Derivarea si Integrarea : Le fac dupa formula  $x^n = nx^{n-1}$  si  $x^n = x^{n+1}/n+1$ .

### 3. Proiectare



In clasa Polinom am doi constructoare , unul in care doar imi construiesc un nou TreeMap la apelare, iar altul care imi creeaza polinomul dintr-un string pe baza unui pattern matcher cu un regex.

### 4. Implementare

In clasa Monom am un constructor ce continue un coeficient si puterea acestuia si o metoda toString impreuna cu setter si getter.

In clasa Polinom am 2 constructoare din care unul primeste un String pe baza caruia creeaza un nou polinom dupa un anumit pattern. O functie care imi returneaza gradul maxim, si una in care adaug Monom in polinom. O metoda toString.

In clasa Operation am implementat cele 6 operatii pe polinoame.

In clasa View mi-am declarat fiecare buton si JTextField , facandu-l pe cel cu rezultatul needitabil, in constructorul acelei clase am atribuit celor doua JPanel-uri butoanele si textfieldurile. Am asezat cele

doua panel-uri intr-un grid cu 2 randuri si o coloana. Si totodata in clasa am metodele pentru fiecare buton cu ActionListener.

In clasa control am create mai multe clase pentru fiecare operatie cu ActionListener.

## 5. Rezultate

```
6. package ro.tuc;

import junit.framework.TestCase;
import org.junit.Test;

public class OperationTest extends TestCase {
    public Polinom P1;
    public Polinom P2;

    public void setUp() {
        P1 = new Polinom("x^2");
        P2 = new Polinom("x^1");
    }

    @Test
    public void testAddition() {
        Operation o = new Operation();
        assertEquals("1.0x^2 + 1.0x^1", o.addition(P1, P2).toString());
    }

    @Test
    public void testSubstraction() {
        Operation o = new Operation();
        assertEquals("1.0x^2 - 1.0x^1", o.substraction(P1,
P2).toString());
    }

    @Test
    public void testMultiply() {
        Operation o = new Operation();
        assertEquals("1.0x^3", o.multiplication(P1, P2).toString());
    }

    @Test
    public void testDivide() throws IncorrectPolynomException {
        Operation o = new Operation();
        P1 = new Polinom("x^3-2x^2+6x^1-5");
        P2 = new Polinom("x^2-1");
        assertEquals("1.0x^1 - 2.0x^0 rest: 7.0x^1 - 7.0x^0",
o.division(P1, P2).toString());
    }

    @Test
    public void testDerivation() {
        Operation o = new Operation();
        assertEquals("2.0x^1", o.derivation(P1).toString());
    }

    @Test
    public void testIntegration() {
```

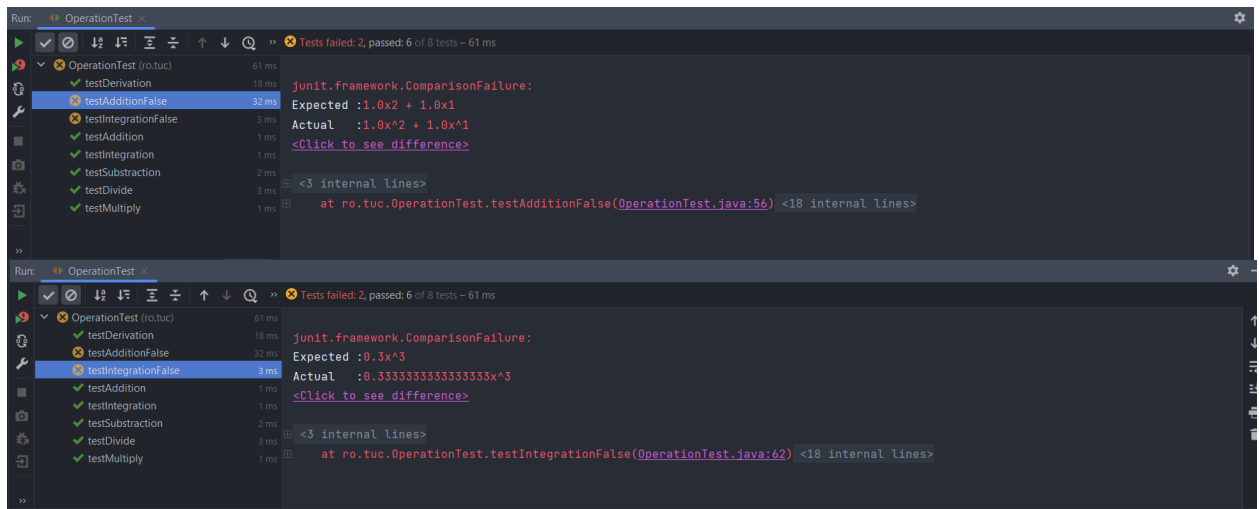
```

        Operation o = new Operation();
        assertEquals("0.5x^2", o.integration(P2).toString());
    }

    @Test
    public void testAdditionFalse() {
        Operation o = new Operation();
        assertEquals("1.0x2 + 1.0x1", o.addition(P1, P2).toString());
    }

    @Test
    public void testIntegrationFalse() {
        Operation o = new Operation();
        assertEquals("0.3x^3", o.integration(P1).toString());
    }
}

```



## 7. Concluzii

Dupa finalizarea acestei teme , am invatat mult mai bine sa folosesc structurile de date de tip Map, totodata sa imi dezvolt abilitatile de debug , deoarece am inatalnit destul de multe erori pe parcursul rezolvarii acestei teme.

## 8. Bibliografie

1. <https://www.baeldung.com/junit-5>
2. <https://regex101.com/>
3. <https://www.geeksforgeeks.org/quantifiers-in-java/>