

# DOCUMENTATIE

TEMA 1 – *Calculator de Polinoame*

Mesaros Radu-Andrei  
GRUPA: 30223

## CUPRINS

1. Obiectivul temei.....	<b>Error! Bookmark not defined.</b>
2. Analiza problemei, modelare, scenarii, cazuri de utilizare .....	4
3. Proiectare .....	5
4. Implementare .....	6
5. Rezultate.....	8
6. Concluzii .....	9
7. Bibliografie .....	9

# Obiective

## Obiectiv Principal

Obiectivul principal al proiectului este proiectarea unui calculator de polinoame.

## Obiectiv Secundare

- Analizarea problemei si identificarea cerintelor
- Design-ul calculatorului polinomial
- Implementarea calculatorului
- Testarea calculatorului

# Analiza problemei, modelare, scenarii, cazuri de utilizare

## Cerinte functionale

Se cere proiectarea si implementarea unui calculator care permite utilizatorului sa calculeze anumite operatii intre 2 polinoame introduse de acesta.

Polinoamele introduse trebuie sa aiba coeficientii intregi.

## Operatii

- Adunare
- Scadere
- Inmultire
- Impartire
- Derivare
- Integrare

## Cerinte non-functionale

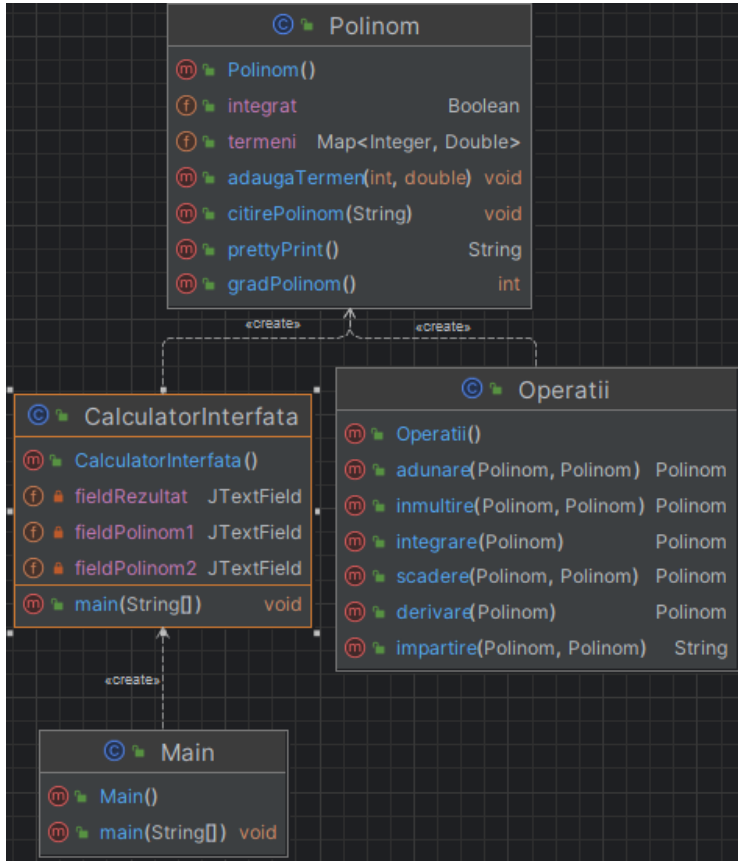
- Interfata responsiva si operatii matematice eficiente
- Teste pentru asigurarea corectitudinii
- Gestionarea erorilor
- Interfata clara, cu butoane si text box-uri sugestive pentru utilizator

## Use-case

Utilizatorul introduce in text box-urile “Polinom 1” si “Polinom 2” un sir de caractere de forma “ $ax^n+bx^m-\dots$ ”. Dupa care va apasa pe unul dintre butoanele cu operatii pentru a vizualiza in campul “Rezultat” rezultatul operatiei. Utilizatorul trebuie sa se asigure ca datele introduse sunt valide. In cazul in care datele sunt introduse gresit, se va afisa pe ecran un mesaj pop-up de eroare, dar cu posibilitatea de a introduce din nou date, fara a reporni calculatorul.

# Proiectare

## Diagrama UML de clase



## Structuri de date

```
public Map<Integer, Double> termeni;
```

Aici se stocheaza termenii fiecarui polinom. Key-ul este gradul termenului, iar valoarea coeficientul

## Algoritmi

Algoritmii folositi sunt cei obisnuiti, fara alte implementari aditionale. Polinoamele sunt parcurse de termen cu termen, se efectueaza operatia in functie de gradul fiecarui termen.

# Implementare

```
public class Polinom {
    31 usages
    public Map<Integer, Double> termeni;
    3 usages
    public Boolean integrat;

    35 usages  ⤴ Messa
    public Polinom() {
        this.integrat=false;
        this.termeni = new HashMap<>();
    }

    54 usages  ⤴ Messa
    public void adaugaTermen(int grad, double coeficient) {
        termeni.put(grad, coeficient);
    }

    10 usages  ⤴ Messa
    public void citirePolinom(String polinomStr) {...}

    6 usages  ⤴ Messa
    public int gradPolinom() {...}

    17 usages  ⤴ Messa
    public String prettyPrint() {...}
}
```

Clasa Polinom contine Map-ul termeni, unde sunt stocati termenii fiecarui polinom, key-ul fiind gradul fiecarui termen, iar value este coeficientul termenului. Aici se afla si metoda de adaugare termeni, metoda gradPolinom, care returneaza gradul cel mai mare.

Metoda citirePolinom primeste un string de forma “ $ax^n+bx^m+\dots$ ” si adauga in termeni gradul si coeficientul fiecarui termen din polinomul citit.

Metoda prettyPrint ia din Map-ul termeni gradul si coeficientul fiecarui termen si returneaza polinomul sub forma de string de forma “ $ax^n+bx^m+\dots$ ”.

```
public class Operatii {
    3 usages  ⤴ Messa
    public static Polinom adunare(Polinom p1, Polinom p2) {...}

    3 usages  ⤴ Messa
    public static Polinom scadere(Polinom p1, Polinom p2) {...}
    3 usages  ⤴ Messa
    public static Polinom inmultire(Polinom p1, Polinom p2) {...}

    2 usages  ⤴ Messa
    public static Polinom derivare(Polinom p) {...}

    2 usages  ⤴ Messa
    public static Polinom integrare(Polinom p) {...}

    4 usages  ⤴ Messa
    public static String impartire(Polinom p1, Polinom p2) {...}
}
```

Clasa Operatii contine toate operatiile cerute: adunare, scadere, inmultire, impartire, derivare, integrare. La metodele de adunare, scadere, inmultire avem ca parametrii 2 polinoame, se returneaza un al treilea polinom cu rezultatul dorit. La derivare si integrare avem ca parametru un singur polinom, se returneaza un al doilea polinom cu rezultatul dorit. Metoda de impartire are ca parametri 2 polinoame, dar returneaza un string. String-ul contine mesajul “Cat” si “Rest”, si 2 pretty print-uri de la polinomul cat si rest, pe care le obtinem in metoda. De asemenea, impartirea foloseste operatiile implementate anterior (scaderea, inmultirea etc)

```

public class CalculatorInterfata {
    13 usages
    private final JTextField fieldPolinom1 = new JTextField( columns: 20)
    9 usages
    private final JTextField fieldPolinom2 = new JTextField( columns: 20)
    8 usages
    private final JTextField fieldRezultat = new JTextField( columns: 20)

    3 usages  1 Messa
    public CalculatorInterfata() {
        JPanel c = new JPanel();
        c.setLayout(new GridLayout( rows: 7, cols: 2, hgap: 10, vgap: 10));

        fieldRezultat.setEditable(false);
        JButton adunareButton = new JButton( text: "Adunare (+)");
        JButton scadereButton = new JButton( text: "Scadere (-)");
        JButton inmultireButton = new JButton( text: "Inmultire (*)");
        JButton impartireButton = new JButton( text: "Impartire (/)");
        JButton derivareButton = new JButton( text: "Derivare (dx/dt)");
        JButton integrareButton = new JButton( text: "Integrare (∫)");

        JLabel labelPolinom1 = new JLabel( text: "Primul polinom: ");
        c.add(labelPolinom1);
        c.add(fieldPolinom1);
        JLabel labelPolinom2 = new JLabel( text: "Al doilea polinom: ");
        c.add(labelPolinom2);
        c.add(fieldPolinom2);
        JLabel labelRezultat = new JLabel( text: "Rezultat: ");
        c.add(labelRezultat);
        c.add(fieldRezultat);

        c.add(adunareButton);
        c.add(scadereButton);
        c.add(inmultireButton);
        c.add(impartireButton);
        c.add(derivareButton);
        c.add(integrareButton);

        JFrame frame = new JFrame( title: "Calculator");
        frame.setContentPane(c);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);

        adunareButton.addActionListener(e -> {...});

        scadereButton.addActionListener(e -> {...});

        inmultireButton.addActionListener(e -> {...});

        derivareButton.addActionListener(e -> {...});

        integrareButton.addActionListener(e -> {...});

        impartireButton.addActionListener(e -> {...});
    }
}

```

Clasa CalculatorInterfata este o interfata grafica a unui calculator polinomial implementata folosind JavaSwing. Interfata are un design simplu si intuitiv, usor de utilizat si inteles. In cadrul acestei clase sunt definite componente precum butoane, campuri de introducere a datelor (TextField-uri), etichete (Label-uri) si actiuni (ActionListener-uri) asociate fiecarui buton.

Atunci cand utilizatorul interactioneaza cu un buton, datele introduse sunt preluate si procesate corespunzator. De exemplu, pentru operatii precum adunare, scadere, inmultire, impartire, derivare sau integrare a polinoamelor, se folosesc metodele corespunzatoare din clasele Polinom si Operatii.

In cazul unei introduceri incorecte a datelor, utilizatorul este informat printr-un mesaj de eroare afisat intr-o fereastra de dialog.

Aceasta interfata grafica ofera o modalitate comoda si eficienta de utilizare a operatiilor polinomiale, permitand utilizatorului sa efectueze diverse calcule fara a fi nevoie sa cunoasca detaliile operatiilor matematice subiacente.

# Rezultate si Testare

## Adunare

$$P1=2x^2+3x+4, P2= 3x^2+2x+1$$

$$\text{Rezultat: } 5x^2+5x+5$$

## Scadere

$$P1=2x^2+3x+4, P2= 3x^2+2x+1$$

$$\text{Rezultat: } -x^2+x+3$$

## Inmultire

$$P1=2x^2+3x+4, P2= 3x^2+2x+1$$

$$\text{Rezultat: } 6x^4+13x^3+6x^2$$

## Impartire

$$P1= x^3-2x^2+6x-5, P2= x^2-1$$

$$\text{Rezultat1: Cat: } x - 2.0, \text{ Rest: } 7.0x - 7.0$$

$$P1= x^3-2x^2+6x-5, P2= x^4-1$$

$$\text{Rezultat2: Gradul primului polinom este mai mic}$$

$$P1= x^3-2x^2+6x-5, P3= 0$$

$$\text{Rezultat3: Nu se poate face impartirea cu } 0$$

## Derivare

$$P1= 3x^2+2x+1$$

$$\text{Rezultat: } 6x+2$$

## Integrare

$$P1= 3x^2+2x+1$$

$$\text{Rezultat: } x^3+x^2+x$$

```
✓ Tests passed: 6 of 6 tests - 9 ms
C:\Users\Utilizator\.jdk\openjdk-21.0.2\bin\java.exe ...
Initializare calculator
Testul de adunare s-a incheiat cu succes
Testul de inmultire s-a incheiat cu succes
Testul de impartire s-a incheiat cu succes
Testul de impartire cu polinom de grad mai mare s-a incheiat cu succes
Testul de impartire cu 0 s-a incheiat cu succes
Testul de integrare s-a incheiat cu succes
Testul de derivare s-a incheiat cu succes
Testul de scadere s-a incheiat cu succes
Din 6 teste, au trecut 6 teste
```



# Concluzii

În opinia mea, cel mai greu a fost începutul proiectului. Nu știam exact de unde să îl apuc, cum să încep, ce plan să îmi fac. Dar după ce am reușit să fac o ‘bază’ a proiectului, operațiile au fost destul de ușor de implementat. Anumite cazuri ‘speciale’ au durat mai mult de implementat, dar am reușit până la urmă.

Pe viitor, s-ar putea implementa operații pe mai multe polinoame, nu doar pe 2. O altă implementare ar putea fi folosirea mai multor operații în același timp, unde ne am putea folosi de paranteze.

# Bibliografie

[https://gitlab.com/mariustotoian225/POO\\_Lab\\_Materials\\_30223\\_2023/-/blob/main/lab\\_12/src/test/java/CalculatorTest.java?ref\\_type=heads](https://gitlab.com/mariustotoian225/POO_Lab_Materials_30223_2023/-/blob/main/lab_12/src/test/java/CalculatorTest.java?ref_type=heads)

<https://stackoverflow.com/>

<https://dsrl.eu/courses/pt/>

<https://docs.oracle.com/javase/tutorial/uiswing/index.html>

<https://www.baeldung.com/junit-5>