

Documentatie

Calculator de polinoame

Tema numarul 1

Costea Delia

Grupa: 30222

CUPRINS

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

1. Obiectivul temei:

Obiectivul temei a fost implementarea unui calculator de polinoame cu o singura variabila si cu coeficienti intregi. Acest calculator trebuie sa ofere si o interfata grafica, unde utilizatorul poate introduce doua polinoame de acest fel si sa aleaga operatia pe care sa o realizeze calculatorul: adunarea, scaderea, inmultirea, impartirea, derivarea sau integrarea polinoamelor intorduse, si sa observe rezultatul operatiei alese.

Calculatorul trebuie sa fie usor de accesat de catre orice persoana, chiar daca aceasta nu are pregatire tehnica suplimentara.

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

Fiecare polinom este alcatuit din mai multe monoame, acestea contin gradul si coeficientul (de ex: $4x^3 + 2x^2$ contine doua monoame, primul fiind $4x^3$, iar al doilea este $2x^2$), pentru spargerea polinoamelor in monoame am folosit un Regex, care sa imi permita mai usor sa preiau stringul dat de utilizator si sa il impart in monoame, astfel reusind sa fac calculele necesare operatiilor.

Aplicatia interactioneaza cu utilizatorul printr o interfata grafica usor de utilizat care ofera o metoda de introducere a datelor, dar si una de afisare a rezultatului. Unele dintre operatii necesita doua polinoame (ex: adunare, scadere, inmultire si impartire), in timp ce altele necesita un singur polinom (ex: derivare si integrare).

Pentru primul caz de utilizare:

- Operatiile necesita doua polinoame ca si intrare si ofera rezultatul calculului in casuta numita Rezultat, in cazul operatiei de scadere si impartire, din primul polinom se scade/imparte cel de-al doilea polinom.

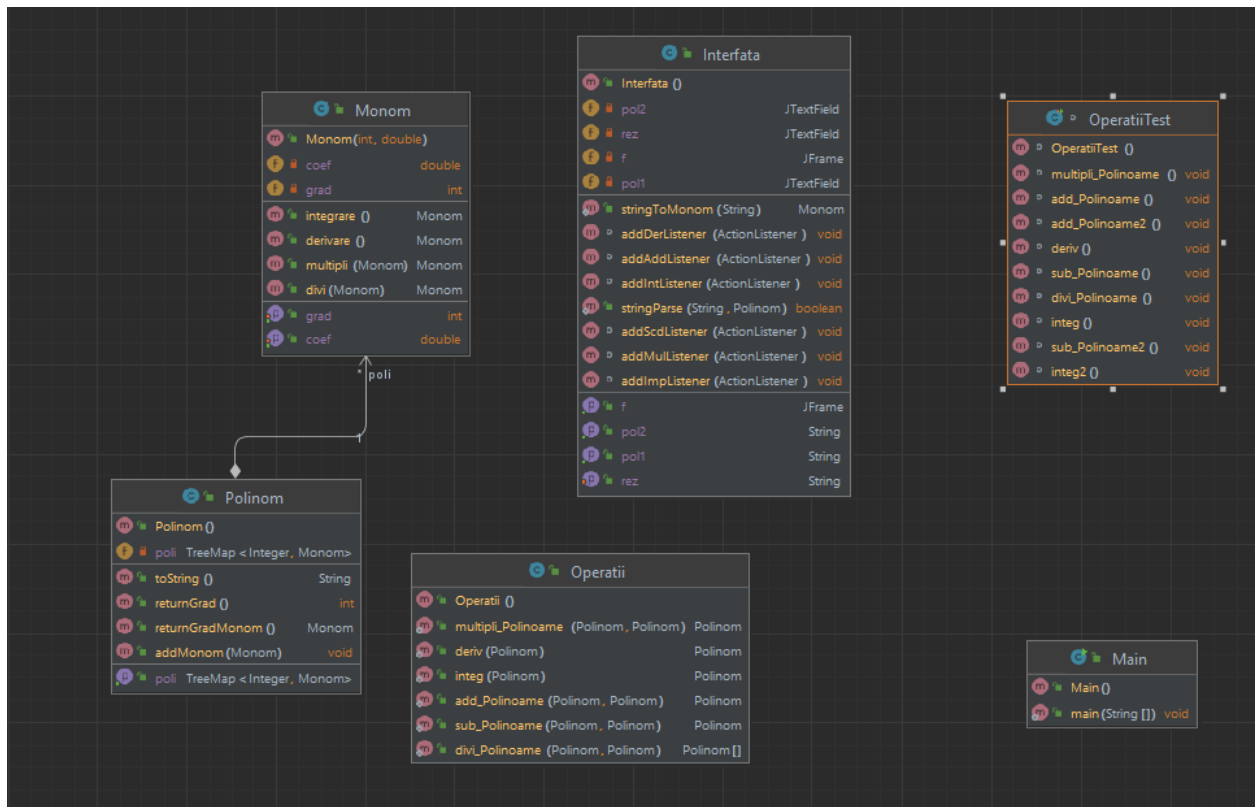
Pentru cel de-al doilea caz de utilizare:

- Daca operatia aleasa va fi cea de derivare sau integrare, calculul se face asupra primului polinom, rezultatul fiind afisat tot in casuta numita Rezultat.

Pentru ambele cazuri, utilizatorul trebuie sa introduca polinoamele sub forma:

coeficient x^{putere} , chiar daca coeficientul sau puterea este 1, altfel nu va functiona correct.

3. Proiectare



4. Implementare

- **Clasa Main:** Este clasa din care se realizeaza deschiderea interfetei grafice; Metoda `main()` este punctul de start pentru Java Virtual Machine, fara metoda `main` programul nu va fi executat.
- **Clasa Polinom:** Este clasa in care folosesc un `TreeMap` pentru stocarea polinoamelor, care are ca si cheie, gradul, fiind unic, si ca si valoare monomul (gradul si coeficientul).
Cea mai importanta metoda a clasei este cea de adaugare a unui nou monom in polinom, aici am considerat trei cazuri, in cazul in care un polinom contine un monom cu acelasi grad ca si monomul introdus nou, adunam coeficientii, iar daca nu contine adauga un nou monom cu gradul si coeficientul monomului nou. In cazul in care coeficientul devine 0, stergem monomul din polinom.
- **Clasa Monom:** Este clasa in care se afla doua variabile importante: gradul si coeficientul monomului, am folosit incapsularea, ambele fiind declarate private si accesul la ele facandu-se prin settere si gettere.

Metodele din aceasta clasa, reprezinta operatii pe un monoame, fiind mult mai usor de realizat operatiile astfel.

De exemplu operatia de inmultire:

```
public Monom multipli(Monom m1)
{
    Monom m2=new Monom(this.grad+m1.getGrad(), this.coef*m1.getCoef());
    return m2;
}
```

- **Clasa Operatii:** Este clasa in care au loc toate operatiile polinoamelor (adunare, scadere, impartire, inmultire, derivare, si integrare).

Pentru fiecare operatie, am parcurs polinomul cu un monom intr-un for;

Pentru operatiile de adunare si scadere am utilizat metoda addMonom a clasei Polinom, dar pentru operatia de scadere am setat coeficientul ca fiind cu ' - ' in fata.

Pentru celelalte operatii, am folosit metodele din clasa Monom, iar apoi am adaugat intr un nou polimon rezultatele folosind tot metoda addMonom.

De exemplu operatia de adunare si cea de derivare:

```
public static Polinom add_Polinoame(Polinom p1, Polinom p2)
{
    Polinom p3;
    p3=p1;
    for (Monom m1:p2.getPoli().values())
        { p3.addMonom(m1);
        }
    return p3;
}
```

```
public static Polinom deriv(Polinom p1)
{
    Polinom p2=new Polinom();
    for (Monom m1:p1.getPoli().values())
        {
            p2.addMonom(m1.derivare());
        }
    return p2;
}
```

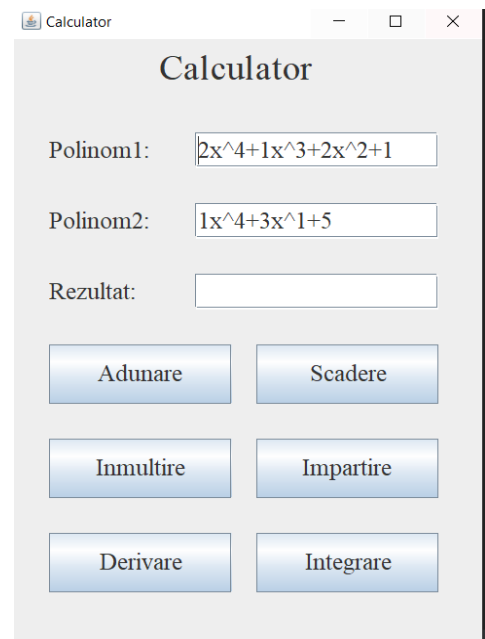
- Clasa Interfata: Aici am implementat interfata grafica, folosind Java Swing,

am declarant si initializat obiectele din interfata: butoanele, frame ul, text field ul, etc, si am adaugat Action Listener butoanelor (care reprezinta operatia aleasa de utilizator), folosind metodele din clasa Operatii. Dupa apasarea oricarui buton, va aparea rezultatul operatiei alese intr-un text field de langa labelul numit “Rezultat”. In cazul in care utilizatorul va introduce date gresite, se va deschide o fereasta noua si va aparea mesajul : “Input incorect”.

Pentru text field-urile celor doua polinoame pe care trebuie sa le introduca utilizatorul, am lasat niste date standar la deschiderea interfetei pentru ca utilizatorul sa observe modul in care se introduce datele:

```
p1 = new JLabel( text: "Polinom1: ");
p1.setBounds( x: 30, y: 70, width: 200, height: 50);
p1.setFont(new Font( name: "Times New Roman", Font.PLAIN, size: 20));
pol1 =new JTextField("2x^4+1x^3+2x^2+1");
pol1.setBounds( x: 150, y: 80, width: 200, height: 30);
pol1.setFont(new Font( name: "Times New Roman", Font.PLAIN, size: 20));

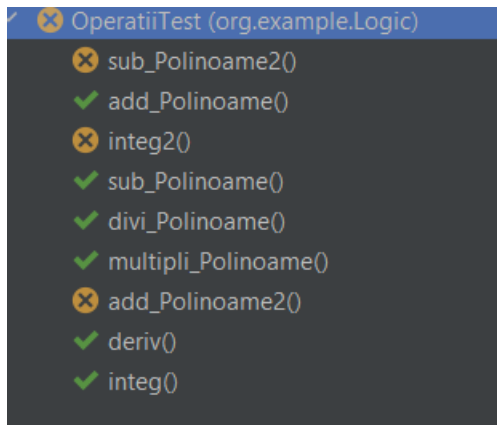
p2 = new JLabel( text: "Polinom2: ");
p2.setBounds( x: 30, y: 130, width: 200, height: 50);
p2.setFont(new Font( name: "Times New Roman", Font.PLAIN, size: 20));
pol2 =new JTextField("1x^4+3x^1+5");
pol2.setBounds( x: 150, y: 140, width: 200, height: 30);
pol2.setFont(new Font( name: "Times New Roman", Font.PLAIN, size: 20));
```



Aspectul interfetei grafice este urmatorul:

5. Rezultate

Pentru testarea metodelor din clasa operatii am utilizat Junit si am verificat daca rezultatele date in urma apelarii metodelor sunt corecte, sau gresite in unele cazuri. Rezultatele testelor:



Exemple de teste corecte si gresite:

@Test

```
void add_Polinoame() {
    Polinom p1=new Polinom();
    Polinom p2=new Polinom();
    Monom m1=new Monom(3,2);
    Monom m2=new Monom(2,3);
    Monom m3=new Monom(1,1);
    Monom m4=new Monom(4,3);
    p1.addMonom(m1);
    p1.addMonom(m2);
    p2.addMonom(m3);
    p2.addMonom(m4);
    Polinom rez=Operatii.add_Polinoame(p1,p2);
    assertEquals(rez.toString(), "+3.0x^4+2.0x^3+3.0x^2+1.0x^1");
}
```

@Test

```
void add_Polinoame2() {
    Polinom p1=new Polinom();
    Polinom p2=new Polinom();
    Monom m1=new Monom(3,2);
    Monom m2=new Monom(2,3);
    Monom m3=new Monom(1,1);
    Monom m4=new Monom(4,3);
    p1.addMonom(m1);
```

```
p1.addMonom(m2);  
p2.addMonom(m3);  
p2.addMonom(m4);  
Polinom rez=Operatii.add_Polinoame(p1,p2);  
assertEquals(rez.toString(), "+3x^4+2x^3+3x^2+1x^1");  
}
```

6. Concluzii

În concluzie, această temă m-a ajutat să îmi îmbunătățesc cunoștințele în limbajul Java și să încep să înțeleg mai bine cum funcționează limbajul. Calculatorul funcționează corect, dar ar putea fi îmbunătățit, de exemplu, funcționează doar dacă scriem coeficientul fiecărui termen, chiar dacă acesta este 1, dacă adun $4x^2$ cu x^2 , nu va funcționa corect, aceeași problemă există și în cazul în care monomul are gradul 1, trebuie scris neapărat x^1 .

7. Bibliografie

1. <https://www.w3schools.com/>
2. <https://dsrl.eu/courses/pt/>
3. <https://www.javatpoint.com/java-regex>