**Sistem de procesare a polinoamelor**

**Dondas Bianca-Elena**

**Grupa 30228**

**Profesor Dr. Ing. Cristina Pop**

**Cuprins**

# Obiectivul temei

1. Analiza problemei
2. Proiectare
3. Implementare
4. Testare
5. Rezultate
6. Concluzii
7. Bibliografie

# Obiectivul temei

Propuneti, proiectati si implementati un sistem de procesare a polinoamelor de o singura variabila cu coeficienti intregi.

Se cere dezvoltarea un calculator de polinoame care sa perimita: citirea unui polinom de la tastatura (un string de forma: 3x^8 + 12x^3 + 9x) si operatiile matematice de adunare, acadere, inmultire, impartire, derivare, integrare.

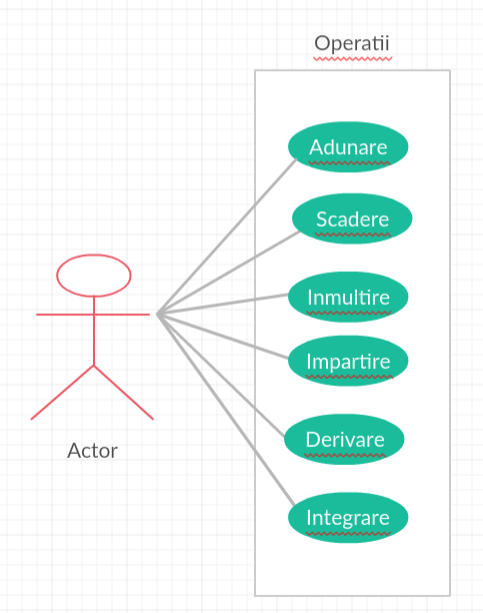
1. **Analiza problemei**

Pentru a rezolva implemantarea unui astfel de sistem avem nevioe sa ne aducem aminte cateva lucruri generale despre polinoame si despre operatiile matematice care se pot face asupra lor.

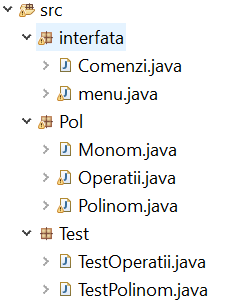
In matematică, un polinom este o expresie construită dintr-una sau mai multe variabile și constante, folosind doar operații de adunare, scădere, înmulțire și ridicare la putere constantă pozitivă întreagă. De exemplu este un polinom

Polinoamele sunt construite din termeni numiți monoame, care sunt alcătuite dintr-o constantă (numită coeficient) înmulțită cu una sau mai multe variabile. Fiecare variabilă poate avea un exponent constant întreg pozitiv.

**Diagrama use case a proiectului:**



1. **Proiectare**

*Pachete*

Pentru o mai buna structura a proiectului am ales sa il impart in pachete: Pol, Test si Interfata. Primul pachet contine clasele: “*Momon”*, *“Polinom”* si *“Operatii”*. Al doilea pachet contine clasele *“TestPolinom”* si *“TestOperatii”* folosite pentru testarea diferitelor functii din clasele din pachetul “Pol”, “*Polinom”* respectiv *“Operatii”*.

Cea de a treia clasa contine clasele *“menu”* si *“Comenzi”* care impreuna formeaza interfata grafica a utilizatorului.

*Pache tul Pol*

*Clasa Monom*

In aceasta clasa se creaza obiectul monom si anumite operatii asupra acestui obiect, cum ar fi: getCoeficient, getPutere, pentru determinarea coeficientului si a puterii monomului, setCoeficient, setPutere pentru schimbarea acestora. O alta functie implementata in aceasta clasa este cea de transformare a unui monom in string.

*Clasa Polinom*

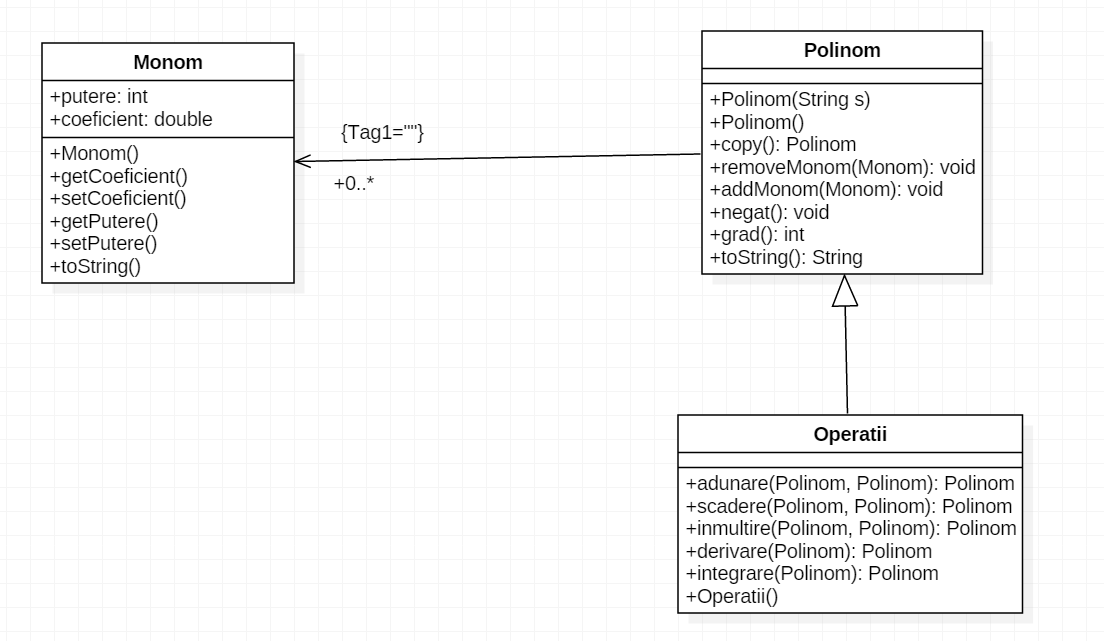
Un polinom este o combinatie de mai multe monoame. Aici un polinom e implementat ca o lista de monoame.

Sunt implememntate metode de adaugare, stergere, negare, determinare a gradului, copiere a unui polinom. Cea mai importanta metoda a acestei clase este cea de transformare a unui string intr-un sir de momoame.

*Clasa Operatii*

Aici sunt implementate operatiile matematice cerute in cerinta temei: adunare, scadere, inmultire, impartire, derivare si integrare a unui polinom.

Mai jos este figura corespunzatoare diagramei UML a pachetului Pol:



Pachetul Interfata

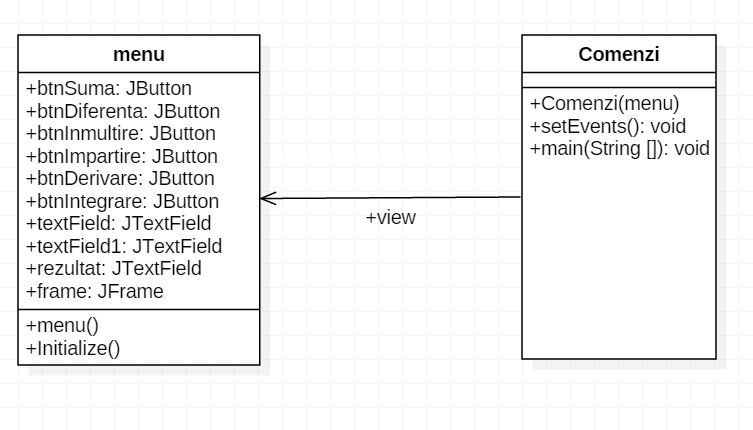
*Clasa menu*

Se creeaza o fereastra (frame), se pun butoanele pentru operatii si textField-urile pentru introducerea polinoamelor si pentru afisarea rezultatului.

*Clasa Comenzi*

In aceasta clase se face conexiunea intre cod si GUI (Graphic User Interface) prezentat ultierior.

Mai jos Mai jos este figura corespunzatoare diagramei UML a pachetului Interfata:



Pachetul Test

*Clasa TestPolinom*

Clasa creata pentru a testa metoda de transformare a unui string in polinom descrisa in clasa Polinom.

*Clasa TestOperatii*

Clasa de test folosita pentru testarea operatiilor de adunare, scadere, inmultire, impartire, derivare si integrare a polinoamelor.

Interfata Graphic User Interface (GUI)

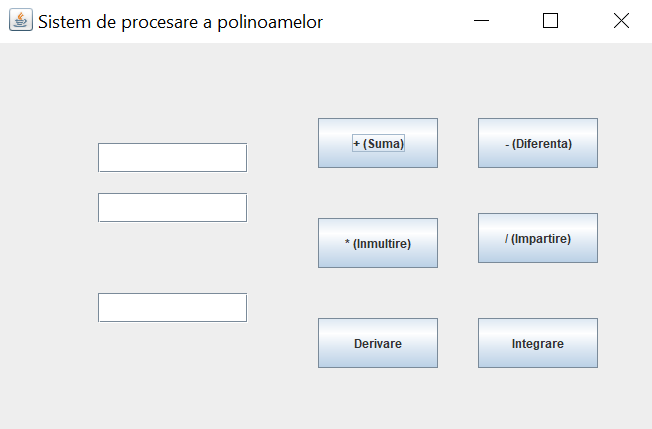
Interfața grafică (în engleză: Graphical User Interface sau GUI este o interfață cu utilizatorul bazată pe un sistem de afișaj ce utilizează elemente grafice. Interfața grafică este numit sistemul de afișaj grafic-vizual pe un ecran, situat funcțional între utilizator și dispozitive electronice cum ar fi computere, dispozitive personale de tip hand-held, aparate electrocasnice și unele echipamente de birou. Pentru a prezenta toate informațile și acțiunile disponibile, un GUI oferă pictograme și indicatori vizuali, în contrast cu interfețele bazate pe text, care oferă doar nume de comenzi (care trebuie tastate) sau navigația text.

Acestea sunt reprezentate de sisteme de programe care, sub o formă sau alta, iniţiază şi întreţin un dialog cu utilizatorul calculatorului, în scopul utilizării şi / sau configurării acestuia.

Avantajele folosirii GUI sunt:

* Interactiune cu computerul mai usoara si mai eficienta pentru utilizator
* Simplificarea instructiunilor complexe, realizata cu ajutorul pictogramelor si a meniurilor
* Initiere intuitiva a comenzilor catre computer
* Programela si fisierele sunt mai usor de manuit si organizat

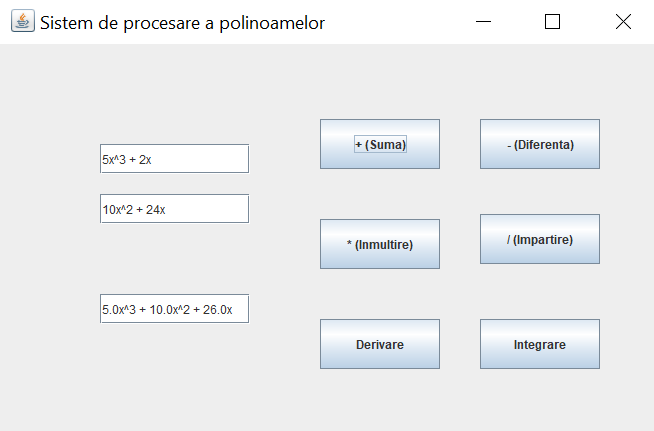
In pictograma de mai jos este ilustrata interfata folosita pentru acest proiect:



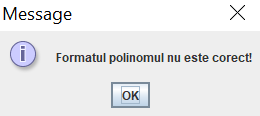
Am folosit trei TextField-uri unul pentru introducerea primului polinom, al doilea pentru introducerea celui de al doilea polinom iar al treilea pentru afisarea rezultatului dupa selectarea operatiei dorite.

Pentru selectarea operatiilor am folosit sase butoane unul pentru fiecare operatie.

In concluzie tot ce trebuie facut este sa intoducem cele 2 polinoame sa dam click pe butonul respective operatiei dorite, iar rezultatul va aparea pe ecran ca in exemplul de mai jos. Rezultatul este suma celor 2 polinoame.



In cazul in care frmatul celor 2 polinoame nu este correct pe ecran va aparea un mesaj de eroare, apasand ok mesajul va disparea si veti putea corecta polinoamele introduse.



Un exemplu de polinom introdus corect este: 3x + -2x^3 + 25x^5

1. **Implementare**

Mai jos vor fi explicate metodele mai importante din clasele mentionate la punctul 3.

4.1 In clasa Monom au fost create gettere si settere pentru coeficientul si puterea monomuli, dar si suprascrierea clasei toString() care aici returneaza monomul sub forma de string. Se verifica pe rand daca coeficientul, daca puterea e 0, 1sau mai mare ca 1. Daca coeficientul este 0 acesta nu va mai aparea, iar daca puterea este egala cu 1 aceasta nu va mai fi scris fiind returnat doar x.

4.2 Clasa Polinom destinata creerii unui polinom din unui sau mai multe monoame. Este declarata o lista de monoame: *“public ArrayList<Monom> listaMonom=new ArrayList<Monom>();”* Sunt implementate metodele de adaugare, stergere, negare, determinare grad a unui polinom si suprea scrierea metodet toString care va fi folosita la afisare pentru transformarea unui polinom in string.

Tot in aceasta clasa este implementata cea mai importanta metoda a proiectului adica cea de transformare a unui string in polinom. Am luat 5 cazuri posibile:

* Atunci cand exista si coeficient si putere, dar puterea e diferita de 1
  + Exemplu: 10x^5 => se vor extrage puterea si coeficientul care vor fi adaugate intr-un obiect de tip Monom
* Atunci cand exista si coeficient si putere, dar puterea e egala cu 1
  + Exemplu: 5x => in acest caz nu este necesara introducerea sub forma “5x^1” dar aceasta nu este gresita. Daca nu introducem puterea acaeasta va avea implicit valoarea 1
* Atunci cand exista coeficient dar un exista putere (adica pueterea este egala cu 0)
  + Exemplu: 5 => va fi introdus coeficientul 5 si puterea 0
* Atunci cand nu exista coeficient dar exista putere egala cu 1
  + Exemplu: x => implicit coeficientul si puterea sunt egale cu 1
* Atunci cand nu exista coeficienti dar exista putere diferita de 1
  + Exemplu: x^5 => implicit coefficient egal cu 1, iar puterea e 5 fiind extrasa din string

**4.3** Clasa Operatii are implementate ca metode operatiile de adunare, scadere, inmultire, derivare si integrare a polinoamelor.

4.3.1 Operatia de adunare

public Polinom adunare(Polinom p1, Polinom p2)

Se adauga cele doua polinoame in cel de-al trilea polinom folosit petru rezultat, mai apoi se verifica daca in noul polinom exista monoame care au aceeasi putere. In cazul in care exista polinoame cu aceeasi putere, ele vor fi adunate si rezultatul va fi adaugat la noul polinom mai apoi urmand ce monoamele respective sa fie sterse din polinomul rezultat. Daca exista monoame cu coeficientul 0 in noul polinom, acestea vor fi eliminate.

4.3.2 Operatia de scadere

public Polinom scadere(Polinom p1, Polinom p2)

Metoda primeste ca parametrii cele 2 polinoame. Primul polinom este adaugat in polinomul rezultat, cel de-al doilea polinom este negat, urmand apoi sa fie si el adaugat in polinomul rezultat apoi si aplica aceeasi logica ca si in functia de adunare.

4.3.3 Operatia de inmultire

public Polinom inmultire(Polinom p1, Polinom p2)

In aceasta metoda se inmulteste fiecare monom din primul polinom cu fiecare monom din cel de al doilea polinom. Daca exista doua monoame in rezultat care au aceeasi putere si se aduna in caz afirmativ.

4.3.4 Operatia de inmultire

Aceasta operatie nu este implememtata in acest proiect, urmand sa fie implementata intr-o dezvoltare ulterioara a proiectului

4.3.5 Operatia de derivare

public Polinom derivare(Polinom p1)

Aceasta metoda implementeaza pentru fiecare monom din polinomul transmis ca parametru formula: ()’ =

4.3.6 Operatia de integrare

public Polinom integrare(Polinom p1)

In aceasta metoda am utilizat dua cazuri:

* Atunci cand gradul polinomlui este egal cu 0 se aplica formula: = const x
* Atunci cand gradul polinomuli este diferit de 0 se aplica formula: =

**4.4** Clasa menu

In clasa menu am declarat si instantiat elementele grafice: frame, textFiedu-ri si butoane.

Pentru initializarea, afisarea, dimmensiunea ferestrei am folosit urmatorul cod:

private JFrame frame; -pentru declarare

frame = new JFrame(); - pentru initializarea ferestrei

frame.setBounds(700, 300, 680, 450); - pentru setarea locatiei ferestrei e ecran

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE); - pentru iesirea din fereastra apasand butonul X din coltul dreapta sus al ferestrei

frame.getContentPane().setLayout(null); - pentru afisarea marginilor

frame.setTitle("Sistem de procesare a polinoamelor"); - pentru setarea unui titlu pentru fereastra

Pentru afisarea butoanelor pe fereastra principala am utlizat urmatoarele functii:

Private JButton btnSuma; - pentru declarare

btnSuma = new JButton("+ (Suma)"); - pentru scrierea in interiorul butonului

btnSuma.setBounds(320, 75, 120, 50); - pentru setarea locatiei si a dimensiunii butonului

frame.getContentPane().add(btnSuma); - pentru afisarea butonului pe fereastra

**4.5** Clasa Comenzi

In aceasta clasa se face conexiunea intre cod si interfata.

Un exemplu de cod “din spatele” unui buton:

view.btnSuma.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

Operatii op = **new** Operatii(); // se declara functiile din clasa Operatii

Polinom polinom1 = **new** Polinom(view.textField.getText()); // se ia primul polinom din primul textField

Polinom polinom2 = **new** Polinom(view.textField1.getText()); // se ia a doilea polinom din al doilea textField

Polinom rezultat = **new** Polinom(); // se declara polinomul in care se va retine rezultatul

rezultat = op.adunare(polinom1, polinom2); // se adauga in rezultat polinoamele dupa alegerea si efectuarea operatiei dorite

view.rezultat.setText(rezultat.toString()); // se afiseaza rezultatul pe ecran

}

});

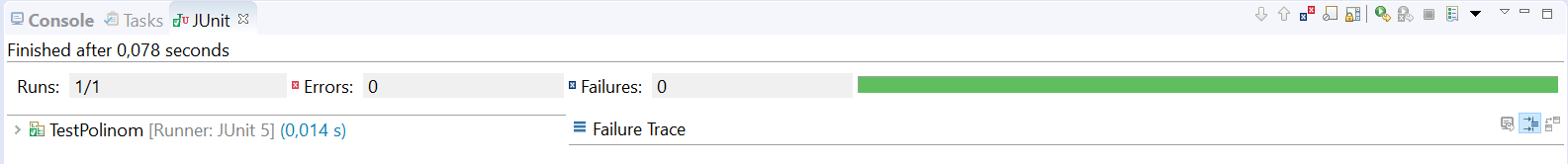
1. **Testare**

**5.1** Clasa TestPolinom

In clasa PolinomTest am ales sa testez doua metode implementate in clasa Polinom, toString().

Testul pentru metoda public String toString() presupune crearea unui nou polinom, dupa care se verifica daca este egal cu un polinom sub forma de String. Se creeaza monoamele polinomului, dupa care se adauga la el. In final se apeleaza metoda toString() pentru noul polinom creat si se verifica daca rezultatul returnat de metoda este egal cu un String.

In imaginea de mai jos se poate observa ca testul a rulat cu succes, fara failures.



**5.2** Clasa TestOperatii

In clasa TestOperatii am implementat cinci teste, unul pentru operatia de adunare, unul pentru operatia de scadere, unul pentru operatia de inmultire, unul pentru operatia de derivare si unul pentru operstia de integrare.

Operatia de adunare presupune crearea a doua polinoame, iar mai apoi apelul functiei adunare (Polinom p1, Polinom p2) din clasa Operatii. Rezultatul returnat se verifica cu un String ales.

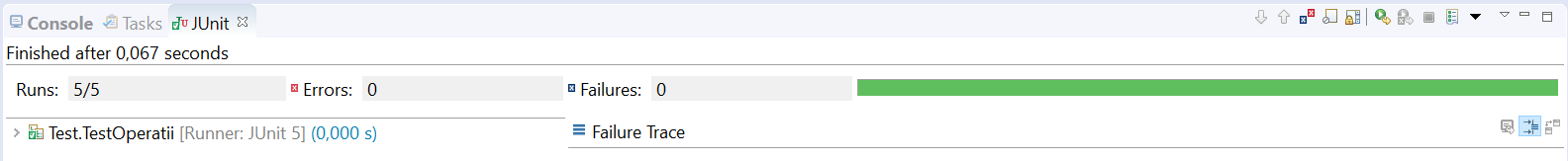
Operatia de scadere presupune crearea a doua polinoame, iar mai apoi apelul functiei scadere(Polinom p1, Polinom p2) din clasa Operatii. Rezultatul returnat se verifica cu un String ales.

Operatia de inmultire presupune crearea a doua polinoame, iar mai apoi apelul functiei inmultire (Polinom p1, Polinom p2) din clasa Operatii. Rezultatul returnat se verifica cu un String ales.

Operatia de derivare presupune crearea unui singur polinoam, iar mai apoi apelul functiei adunare (Polinom p1) din clasa Operatii. Rezultatul returnat se verifica cu un String ales.

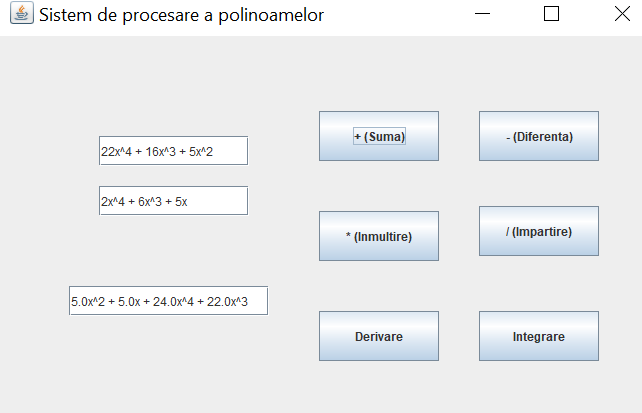
Operatia de integrare presupune crearea unui singur polinoam, iar mai apoi apelul functiei adunare (Polinom p1) din clasa Operatii. Rezultatul returnat se verifica cu un String ales.

In figura urmatoare se poate observa ca toate cele cinci teste au rulat fara erori:

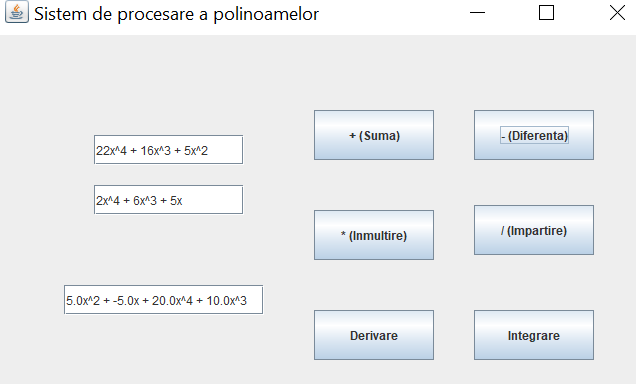


1. **Rezultate**

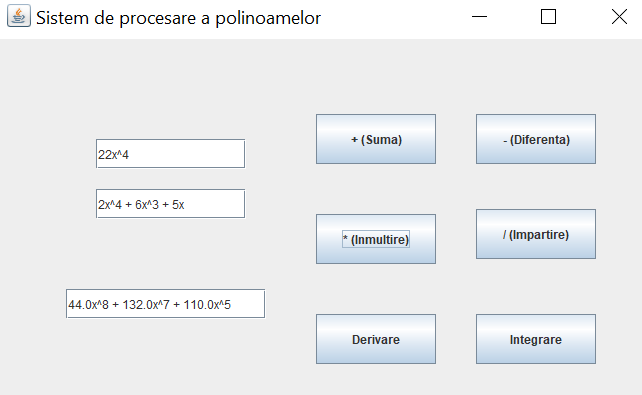
In figura de mai jos este rezultatul operatiei de adunare a celor doua polinoame:



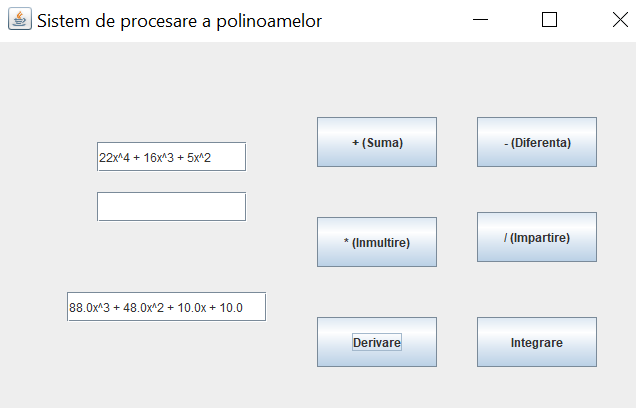
In figura de mai jos este rezultatul operatiei de scadere a celor doua polinoame:



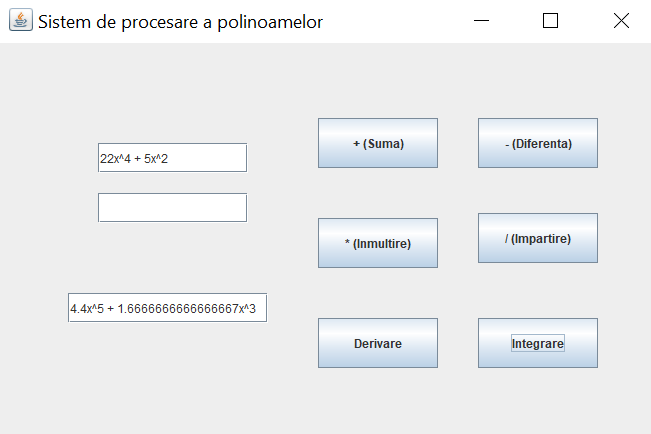
In figura de mai jos este rezultatul operatiei de inmultire a celor doua polinoame:



In figura de mai jos este rezultatul operatiei de derivare a polinoamului:



In figura de mai jos este rezultatul operatiei de integrare a polinoamului:



1. **Concluzii**

In concluzie acesta a fost un proiect amplu, care necestia cunostine de programare orientata pe obiecte, putine cunostinte matematice. Din acest proiect am invatat sa lucram cu interfata, sa transform un string in polinom si invers.

Pentru o dezvoltare ulterioara a proiectului algoritmii ar putea fi imbunatiti, iar functia de impartire a polinomului ar putea fi implementata.

1. **Bibliografie**

* <https://ro.wikipedia.org/wiki/Polinom>
* <http://www.audio.ro/stiatica/ce-este-gui-avantajele-folosirii-gui-si-componentele-de-baza/?lang=ro>