**Sistem de procesare a polinoamelor**

**Danciu Maria-Alexandra**

**Grupa 30222**

**Cuprins**

# Obiectivul temei

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

# Obiectivul temei

Obiectivul acestei teme este de a implementa un sistem de procesare a polinoamelor de o singură variabilă cu coeficienți întregi .

Scopul acestei aplicații este de a efectua operații de bază cu polinoame : adunare , scădere , înmulțire, derivare și integrare . Utilizatorul poate sa introduca doua polinoame in spatiile afisate, iar apoi poate sa aleaga ce operatii doreste sa efectueze.Ulterior poate observa rezultatul afisat pe ecran.

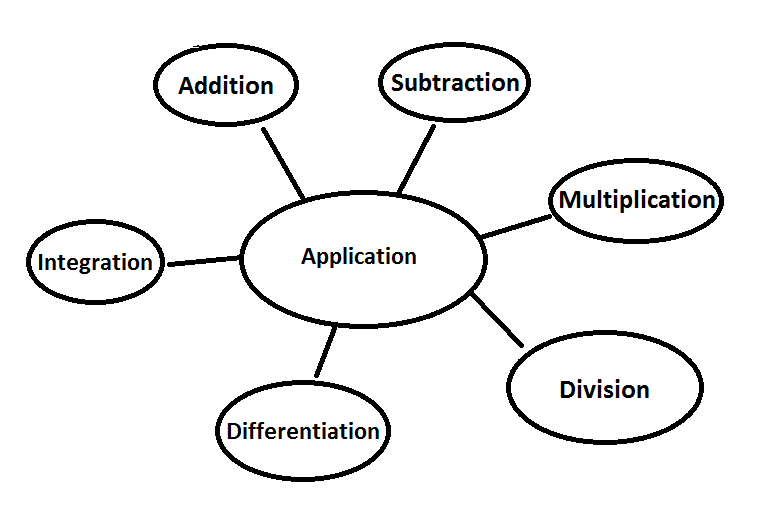
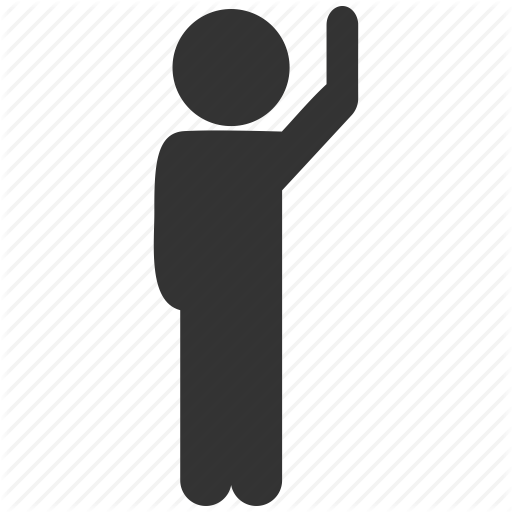
1. **Analiza problemei**

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.

Un polinom de gradul n in nedeterminata X se scrie in forma canonica astfel: P(X) = a0X n + a1X n−1 + ... + an−1X + an, unde a0 6= 0. Numerele a0, a1, ..., an sunt coeficientii polinomului.

Fiecare polinom este alcauit din monoame,care la randul lor sunt alcatuite dintr-o constantă (coeficient) înmulțită cu o variabilă. Fiecare variabilă poate avea un exponent constant întreg pozitiv .Exponentul sugereaza gradul monomului.Gradul polinomului va fi maximul gradelor monoamelor.

**Diagrama use case a proiectului**



Polinoamele se pot introduce intr-un mod cat mai simplu .

Datele de intrare trebuie sa arate de forma ax^m+bx^n+c fara spatiere sau “\*” intre coefficient si variabila.Chiar daca utilizatorul introduce polinomul “dezordonat” la afisare se va scrie acesta in ordinea corespunzatoare gradului.

Interpretorul va analiza X in x.

Exemplu: X+5 va fi interpretat ca : 1x^1+5x^0

2X^4 va fi interpretat ca : 2x^4 (2-coeficient , 4-putere)

**Ghid de utilizare:**

1.Utilizatorul ruleaza aplicatia din clasa Main.

2.Utilizatorul introduce polinomul in spatial afisat pe ecran.(textfield)

3.Apasa pe un puton la alegere in functie de ce operatie vrea sa efectueze asupa lui:

“+” pentru adunare ,”\*” pentru inmultire,”-“pentru scadere.

Polinomul initial va fi afisat aproape de textfield urmand a se elibera spatiul pentru a introduce un nou polinom.

4.Se apasa butonul “=” pentru a se genera rezultatul ce va aparea pe ecran.

5.Pentru a efectua o alta operatie este necesara apasarea butonului de stergere pentru a nu intampina posibile erori la urmatoarele calcule.

1. **Proiectare**

*Clasa Monom*

In aceasta clasa se creaza obiectul monom si operatiile asupra acestui obiect, cum ar fi: getCoeficient, getGrad, pentru determinarea coeficientului si a puterii monomului, setCoeficient, setGrad pentru modificarea acestora. Alte functii implementate in aceasta clasa este cea de transformare a unui monom in string si operatiile caracteristice pe monom:adunare,inmultire,comparare,

*Clasa Polinom*

Un polinom este implementat ca o lista de monoame.Sunt implememntate metode de adunare,inmultire,copiere,stergere si afisare.

*Clasa Operatii*

Aici sunt implementate operatiile matematice cerute in cerinta temei: adunare, scadere, inmultire .Totodata se realizeaza un algoritm efficient de citire a polinoamelor cu ajutorul unui matcher find.

Mai jos este figura corespunzatoare diagramei UML a pachetului Pol:

Clasa Interfata

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

Se utilizeaza un indicator de tip intreg,pentru detectarea tipului de operatie realizat:1-adunare,2-scadere,3-inmultire.

Butoane:butonAdunare,butonScadere,butonInmultire,butonStergere,butonEgal

*Clasa Main*

In aceasta clasa se face conexiunea intre cod si GUI (Graphic User Interface) prezentat ultierior.De aici se va lansa aplicatia pentru utilizare.

Interfata Graphic User Interface (GUI)

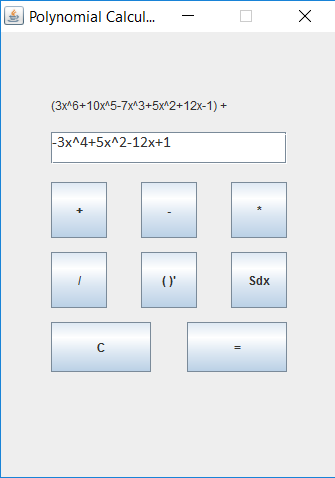
Interfața grafică (în engleză: Graphical User Interface sau GUI este o interfață cu utilizatorul bazată pe un sistem de afișaj ce utilizează elemente grafice. Interfața grafică este numit sistemul de afișaj grafic-vizual pe un ecran, situat funcțional între utilizator și dispozitive electronice cum ar 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:

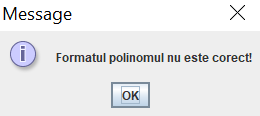


Ulterior am eliminat butoanele de impartire derivare si integrare deoarece nu am reusit sa implementez metodele .

Am folosit 2 TextField-uri unul pentru introducerea primului polinom, al doilea pentru introducerea celui de al doilea polinom .Dupa introducerea primului polinom aceste se muta din textfield-ul principal,se elibereaza spatial pentru noul polinom introdus .Rezultatul se va afisa tot in textfield-ul principal.

Pentru selectarea operatiilor am folosit butoanele specific.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.

Daca formatul primului polinom nu respecta cerintele atunci aplicatia afiseaza un mesaj corespunzator fara a putea realiza operatia .Se asteapta introducerea intr-un format correct.



Un exemplu de polinom introdus corect este: 6x + 2x^3 + 25x^2

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.

Operatia de adunare pe monoame este prezentata mai jos:

Public void add(Monom m)

{

if (this.grad==m.getGrad())

this.coeficient=this.coeficient+m.getCoeficient();

}

Daca monomul dat la parametru are acelasi grad cu monomul pe care se apeleaza operatia atunci ,pe obiectul apelat ,coeficientul acestuia va fi suma dintre coeficientul lui si coeficientul monomului dat ca parametru.

Operatie de inmultire pe monoame este prezentata mai jos:

public void inmultire(Monom m)

{ this.coeficient=this.coeficient\*m.coeficient;

this.grad+=m.grad;

}

La operatia de inmultire se inmultesc coeficientii monoamelor si se aduna gradele

4.2 Clasa Polinom destinata creerii unui polinom din unui sau mai multe monoame. Este declarata o lista de monoame in constructor *:private list<Monom>listaMonoame. lisaMonoame=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. Cazuri posibile:

* + Se verifica daca exista macar un element in lista
  + Daca este doar constanta se afiseaza cu un string format rexzultatul .In acest caz se verifica daca gradul este 0 Exemplu: String s=” “
  + s=String.format(“%s%.2f”,s,(float)list.get(list.size()-1).getCoeficient());
  + Exemplu: 5 => va fi introdus coeficientul 5 si puterea 0
* Daca exista foar o variabila se afisa coeficientul si gradul variabilei x.
* 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.

4.3.1 Operatia de adunare

public Polinom adunare(Polinom polinom1, Polinom polinom2)

Metoda primeste ca doi parametrii ,2 obiecte de tip Polinom.

Initial se copiaza primul polinom cu o metoda corespunzatoare intr-un obiect Polinom copierePolinom1.

Se parcurge monom cu monom Polinomul 2,iar in copie de adauga treptat continutul fiecarui monom din Polinom 2.

Se returneaza copia cu rezultatul final.

4.3.2 Operatia de scadere

public Polinom scadere(Polinom p1, Polinom p2)

Metoda primeste ca parametru doua obiecte de tip Polinom.

In copierePolinom1 se realizeaza copierea primului polinom.In copierePolinom2 se realizeaza copierea celui de-al doilea polinom.Cu un for se parcurge copia2 iar fiecare monom este inmultit cu (-1) pentru a putea realiza de fapt operatia de adunare:

P1-P2=P1+(-P2)

Metoda

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 se vor tine cont de urmatoarele reguli:

* 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 Interfata

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

Clasa Interfata extinde JFrame.

Se vor importa pachete corespunzatoare din java swing.

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

In constructorul clasei Interfata am declarant:

**this.setTitle(“Calculator de polinoame”);**

Pentru setarea titlului unei ferestre,in acest acest caz a aplicatiei.

Am mai utilizat:

this.setLayout(null);

Pentru afisare marginilor.

this.setSize(500,500); dimensiunea ferestrei

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

Pentru afisarea butoanelor pe fereastra principala am utlizat urmatoarele functii:

Private JButton butonAdunare- pentru declarare

butonAdunare.setBounds(50,200,60,60);

butonAdunare=new JButton(“+”);

-pentru setarea locatiei si a dimensiunilor butonului

-x ,y,width,height

butonAdunare.setFocusable(false);

butonAdunare.addActionListener(new doAdunare());

Se adauga proprietati butonului pentru a-I da utilitate si a genera rezultatul.

Pentru fiecare buton s-au realizat o serie de instructiuni specific.

In final se adauga butonul pe APLICATIE.

this.add(butonAdunare);

**4.5** Comenziile butoanelor

In aceasta parte se face conexiunea intre cod si interfata.

Se verifica ce indicator al operatiei este,se inlocuieste X cu x .Se afiseaza un mesaj daca in urma citirii formatul nu este corespunzator,altfel se realizeaza operatia dorita.

Pentru fiecare buton creat , am apelat metoda de addActionListener(new ActionListener()) , iar in momentul acesta se creeaza o instanta de clasa ActionListener si suprascriu metoda actionPerformed in care scriu ce doresc sa faca butonul respectiv . In acest mod , toate metodele sunt scrise in clasa Gui.

1. **Testare**

Initial am verificat in clasa Main daca am realizat correct metodele.Am testat pe obiecte cateva exemple relative simple.Am extras din lista,am adaugat in lista,am efectuat operatii pe polinoame dar si exclusive pe monoame.

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.

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.

1. **Rezultate**

Dupa toti pasii necesari am reusit sa obtinem un calculator ce realizeaza multiple operatii.

1. **Concluzii**

Aceasta aplicatie de calculator de polinoame realizeaza corect operatiile de baza:adunare,inmultire,scadere.Ca o dezvoltare ulterioara as putea sa extend aceste operatii adaugand impartirea,integrarea,derivarea.Totodata, o posibila implementare poate consta in modificarea algoritmului in asa fel incat coeficientii monoamelor sa fie de tip real si astfel rezultatul integrarii sa fie un rezultat cat mai correct (real).

In urma realizarii acestui proiect mi-am imbogatit cunostiinte de programare orientate pe obiect in urma utlizarii Interfetelor din java swing.

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>
* [**http://users.utcluj.ro/~igiosan/teaching\_poo.html**](http://users.utcluj.ro/~igiosan/teaching_poo.html)
* [**https://www.tutorialspoint.com/java/java\_regular\_expressions.htm**](https://www.tutorialspoint.com/java/java_regular_expressions.htm)
* [**https://www.tutorialspoint.com/javaexamples/java\_simple\_gui.htm**](https://www.tutorialspoint.com/javaexamples/java_simple_gui.htm)