Calculator de polinoame

Documentație

Beltechi Marius Gabriel

Grupa 30227 An 2 semestrul 2

Cuprins

1. Obiectiv

2. Studiul probelemei

3.Implementare

3.1 Diagrame UML

3.2 Clase

3.3 Metode

3.4 GUI

4.Implementare si testare

5.Concluzii

6.Bibliografie

1.Obiectiv

Obiectivul acestui proiect a fost de proiecta si implementa un sistem de procesare a polinoamelor de o singura variabila cu coeficienti intregi. Este un calculator de polinoamne care primeste ca date de intrare doua polinoame si returneaza un singur polinom, realizand operatii de adunare, scadere, inmultire si derivare.

Pe langa partea tehnica putem gasi si o interfata grafica “User Friendly” care poate fi utilizata de catre orice utilizator.

2. Studiul probelemei

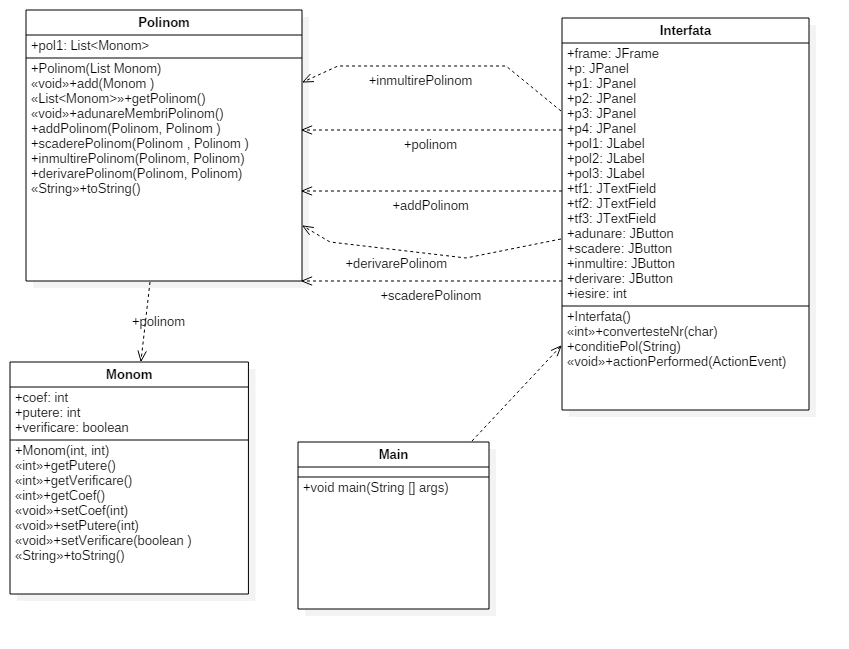
Un polinom este o expresie formata din mai multe monoame. La randul lor si acestea sunt formate din coeficienti sip uteri. Exista mai multe modalitati prin care se poate realiza un polinom. Acestea depinde de modul in care sunt introduse informatiile, modul in care sunt prelucrate si modul in care programatorul a realizat interfata grafica aici existand foarte multe posibilitati.

Exista mai multe posibilitati de a citi un polinom. Una dintre, care presupunea o dificultate mai ridicata, era folosirea unui Regex, care permitea separarea unui string si sa preiau din el doar de ce am nevoie si anume coeficientii si puterile. Eu am ales o cale mai usor de a determina coeficientii si puterile. Am creat doua textField-uri, unul pentru fiecare polinom, si la fiecare am apelat o metoda in care comparam interiorul textField-ului cu anumite caractere si realizam anumite operatii in functie de caracterul pe care il citeam din textField. De exemplu, daca aveam un “x” salvam intr-un vector toate cifrele care erau inainte de x si determinam astfel coeficientul unui monom. Daca era dupa “x” si “^” salvam intr-un alt vector puterea unui monom.

3.Implementare

3.1 Diagrame UML

Unified Modeling Language sau UML pe scurt este un limbaj standard pentru descrierea de modele si specificatii pentru software. UML a fost la bază dezvoltat pentru reprezentarea complexității programelor orientate pe obiect, al căror fundament este structurarea programelor pe clase, și instanțele acestora ( numite și obiecte ). Cu toate acestea, datorită eficienței și clarității în reprezentarea unor elemente abstracte, UML este utilizat dincolo de domeniul IT.



3.2 Clase

***Clasa Interfata:***

Aceasta clasa este granita dintre interfata grafica si programul propriu-zis, unde se petrece interactiunea utilizator-sistem de calcul.

***Clasa Main:***

Aici este clasa cu main-ul in care ii spunem programului ce sa execute. Aici doar am realizat deschiderea interfetei grafice.

***Clasa Polinom:***

In aceasta clasa am declarat polinomul ca o lista de monoame si am realizat operatiile de adunare, scadere, derivare si inmultire. Pe langa acestea am mai facut o functie pentru restrangere a polinomului pentru cazul in care utilizatorul introduce intr-un polinom mai multe monomi care au aceasi putere sa realizeze operatia de adunare intre acesti coeficienti.

3.3 Metode

Metode utilizate in clasa Monom

Aici am declarat structura unui monom. El este format din coeficient si putere. Pe langa acestea am mai declarat si o variabila booleana numita verificare pe care am folosit-o in clasa Polinom pentru a vedea ca am parcurs sau nu un polinom. Pentru fiecare dintre acestea am realizat o metoda de setare a puteri, coeficientului si a verificarii si una de extragere a acesteia.

**public** **int** getCoef()

{

**return** coef;

}

**public** **int** getPutere()

{

**return** putere;

}

**public** **boolean** getVerificare()

{

**return** verificare;

}

**public** **void** setVerificare(**boolean** verificare)

{

**this**.verificare=verificare;

}

**public** **void** setCoef(**int** coef)

{

**this**.coef=coef;

}

**public** **void** setPutere(**int** putere)

{

**this**.putere=putere;

}

Pe langa acestea am mai facut o metoda toString pentru a returna coeficientul si puterea din monom.

Metode utilizate in clasa Polinom

In aceasta clasa am creat o lista de monomi pentru a crea un polinom. Aici am realizat toate operatiile de adunare, scadere, inmultire, derivare si operatia de restrangere a polinomului. Pe langa acestea am mai adaugat o metoda pentru a returna polinomul, pentru a adauga in polinom si una pentru a tipari polinomul.

**public** **void** add(Monom newItem)

{

pol1.add(newItem);

}

**public** List<Monom> getPolinom( )

{

**return** pol1;}

Metoda de **constrangere** a polinomului am realizat-o in metoda: adunareMembriPolinom. Aici am creat un nou polinom pe care l-am parcurs cu doua for-uri si in care am adunat coeficientii la puterile egale. Dupa am parcurs iar polinomul si am sters monomi unde aveam puteri egale, lasand doar un singur monom pentru fiecare putere. De exemplu, daca aveam un polinom care avea trei monomi cu aceeasi putere adunam prima oara primul monom cu al doilea si apoi suma de la primi doi cu al treilea obtinand astfel rezultatul dorit. Apoi am parcurs iar polinomul si am sters monomi cu aceeasi putere lasand-ul doar pe cel cu coeficientul cel mai mare.

**Adunarea** polinomului am realizat-o in metoda: addPolinom. Aceasta metoda primeste doi polinomi ca date de intrare. Aici am initializat o noua lista de mononi si am salvat-o intr-un polinom. Apoi am contrans fiecare dintre acesti doi polinomi pentru a nu avea mai multi coeficienti la aceasi putere. Am creat un nou monom intermediar pe care l-am folosit sa adaug in polinom. Am parcurs in doua for-uri monomii din cele doua polinoame, iar cand am intalnit o putere egala am creat un nou monom la care am trecut la coeficient suma de la cele doua polinoame si am setat verificarea din monom ca fiind “true” deoarece am parcurs acel monom si nu mai dorim sa il mai folosim. Apoi printr-un for am parcurs primul polinom si acolo unde verificarea monomului era “false” am adaugat in polinom. La fel si pentru al doilea polinom.

**Scaderea** polinomului am realizat-o in metoda: scaderePolinom. Aceasta metoda primeste doi polinomi ca date de intrare. Aici am initializat o noua lista de mononi si am salvat-o intr-un polinom. Apoi am contrans fiecare dintre acesti doi polinomi pentru a nu avea mai multi coeficienti la aceasi putere. Am creat un nou monom intermediar pe care l-am folosit sa adaug in polinom. Am parcurs in doua for-uri monomii din cele doua polinoame, iar cand am intalnit o putere egala am creat un nou monom la care am trecut la coeficient difereanta de la cele doua polinoame si am setat verificarea din monom ca fiind “true” deoarece am parcurs acel monom si nu mai dorim sa il mai folosim. Apoi printr-un for am parcurs primul polinom si acolo unde verificarea monomului era “false” am adaugat in polinom. Spre deosebire de adunare aici dorim ca semnul coeficientului din al doilea polinom sa aiba semn opus celui pe care il avea initial si din acest motiv salvez intr-o variabila auxiliara coeficientul si il inversez si dup ail adaug in polinom.

**Inmultirea** polinomului am realizat-o in metoda: inmultirePolinom. Aceasta metoda primeste doi polinomi ca date de intrare. Aici am initializat o noua lista de mononi si am salvat-o intr-un polinom. Apoi am contrans fiecare dintre acesti doi polinomi pentru a nu avea mai multi coeficienti la aceasi putere. Am creat un nou monom intermediar pe care l-am folosit sa adaug in polinom. Am parcurs in doua for-uri monomii din cele doua polinoame si am inmultit coeficientul de la monomul din primul polinom cu coeficientul de la monomul din al doilea polinom, la fel si la puteri, si l-am adaugat coeficientul si puterea respectica la monomul intermediar pe care l-am adaugat la polinom.

**Derivarea** polinomului am realizat-o in metoda: derivarePolinom. Aceasta metoda primeste un singur polinom ca date de intrare. Aici am initializat o noua lista de mononi si am salvat-o intr-un polinom. Apoi am contrans polinomul de la intrare pentru a nu avea mai multi coeficienti la aceasi putere.Dupa intr-un for am parcurs polinomul si am inmultit coeficientul cu puterea si l-am adaugat in monomul intermediar pe care l-am adaugat in polinom.

**Integrarea** polinomului am realizat-o in metoda: integrarePolinom. Aceasta metoda primeste un singur polinom ca date de intrare. Aici am initializat o noua lista de mononi si am salvat-o intr-un polinom. Apoi am contrans polinomul de la intrare pentru a nu avea mai multi coeficienti la aceasi putere.Dupa intr-un for am parcurs polinomul si am impartit coeficientul cu puterea si l-am adaugat in monomul intermediar pe care l-am adaugat in polinom.

Metoda toString din clasa polinom am folosit ca sa tiparesc polinomul. Am avut mai multe cazuri. La fiecare monom in functie de coeficientul monomului si cel de dinaintea lui am determinat daca urmeaza sa pun semnul “+” sau “-”.

Metode utilizate in clasa interfata

Metoda conditiePol primeste ca data de intrare un string de la un textField in care se afla polinomul1 sau polinomul 2 si il proceseaza extragand din acesta coeficientul si puterea pentru fiecare monom al polinomului. Am initializat un vector de puteri de coeficienti cu 0 si alte variabile pe care le-am folosit in prelucrarea string-ului. Am salvat fiecare character din sir intr-o variabila pe care am verificat-o daca are anumite valori. De exemplu daca are o cifra am convertit-o si am salvat-o intr-o variabila auxiliara si daca aparea iar cifra variabila auxiliara era inmultita cu zece si i se aduna cifra si tot asa pana apare un alt character diferit de cifra.

Metoda actionPerformed este o interfata care poate sa fie implementata pentru a determina modul in care un eveniment va avea loc. La implementarea unei interfete toate metodele din acea interfata trebuie implementate. Interfata actionListener are o metoda de implementat numita actionPerformed(). Eu am adaugat cate un actionListener pentru fiecare dintre cele butoane pentru operatiile respective.

**3.4 GUI**

Interfarta grafica sau GUI este o interfata cu utilizatorul bazata pe un sistem de afisaj ce utilizeaza elemente grafice. Interfata grafica este numit sistemul de afisaj grafic-vizual pe un ecran. Situate functional intre utilizator si dispositive electronice cum ar fi computere Pentru a prezenta toate informatiile si actiunile disponibile, un GUI ofera pictograme si indicatori vizuali, in contrast cu interfetele bazate pe text, care ofera doar nume de comenzi.

Interfata grafica are urmatoarele component: frame, panel, label,textfield, buton.

Frame este “rama” in care se adauga toate elementele de care avem nevoie pentru program.

Panel sunt cele patru panouri propriu-zise. Am facut un panou pentru primul polinom, unul pentru al doilea, altul pentru rezultatul polinomului si unul pentru operatii. In fiecare panou am adaugat label, textfield si buton representative acestuie.

Label-urile le-am folosit pentru a arata ce trebuie sa insereze utilizatorul pentru a face o anumita operatie. De exemplu, la polinom1 sa introduca un polinom in textfield-ul respective.

TextField este practice o casuta text in care utilizator introduce un text, iar programul citeste acest text si il modifica conform intructiunilor din spate. Am pus trei textField-uri, doua pentru cele doua polinoame si unul pentru rezultatul lor. De exemplu tf1 si tf2 sunt textField-urile pentru cele doua polinoame introduse de utilizator, iar tf3 este cel pentru afisarea polinomului rezultat.

Butoanele executa o anumita instructiune in momentul in care sunt apasate.

Am pus patru butoane, cate unul pentru fiecare operatie. Pentru fiecare button, s-a făcut o metodă nouă care implică ActionEvent. Astfel, de fiecare dată când are loc o acțiune la un buton, de exemplu a fost apăsat, ActionEvent-ul denumit e, transmite informația la ActionListener care așteaptă astfel de informații. Apoi, au loc evenimentele ce se afla în metodă respective.

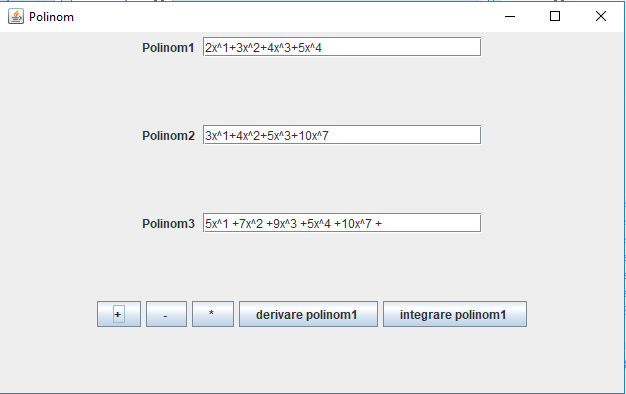
*TextField = spatii dreptunghiulare in care se pot introduce date de la tastatura. Dar pe langa asta pot fi folosite si pentru a afisa rezultatul fara a se putea introduce date de la tastatura.*

*Polinom1= spatiu in care se introduce primul polinom*

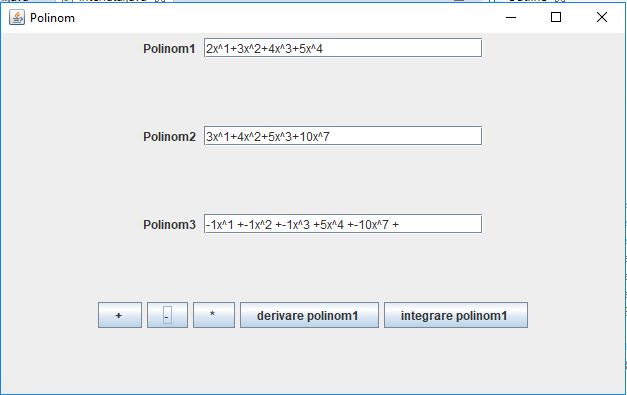
*Polinom2= spatiu in care se introduce al doilea polinom*

*Polinom3= spatiu in care se afiseaza rezultatul*

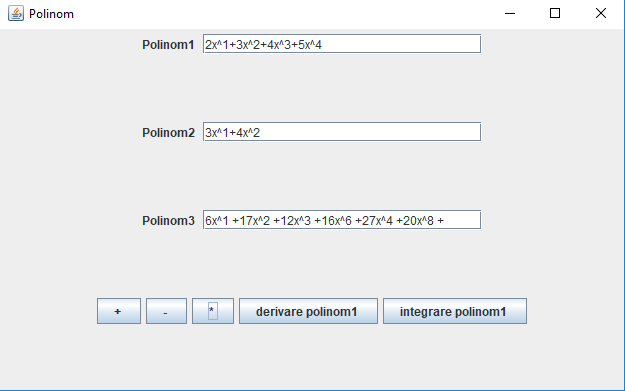
4.Implementare si testare



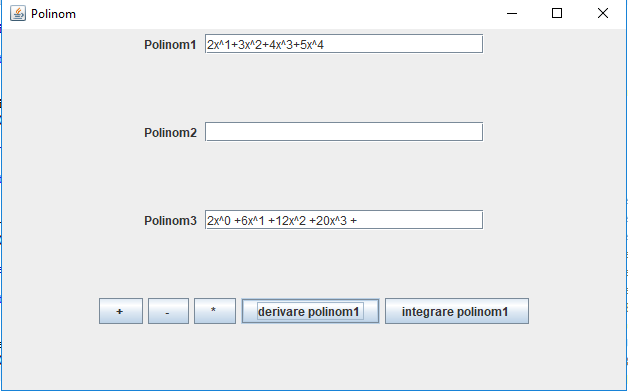
**Adunarea** se realizeaza dupa urmatoarele operatii: se introduce coeficientii si puterile pentru polinomul unu. Dupa se introduce coenficientii si puterile pentru polinomul doi si se apasa butonul “+” care efectueaza adunarea propriu-zisa in polinomul trei. Rezultatul va fi afisat in textField-ul de la polinomul trei.



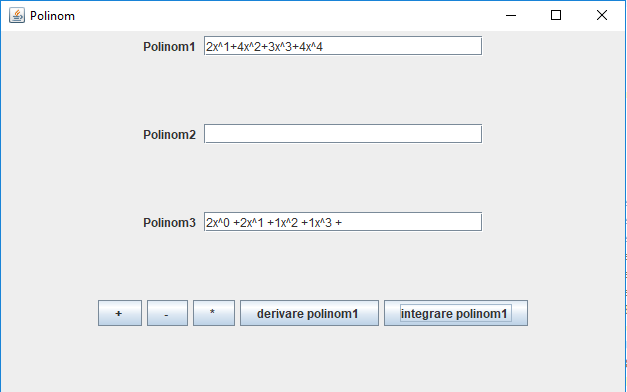
**Scaderea** se realizeaza dupa urmatoarele operatii: se introduce coeficientii si puterile pentru polinomul unu. Dupa se introduce coenficientii si puterile pentru polinomul doi si se apasa butonul “-” care efectueaza scaderea propriu-zisa in polinomul trei. Rezultatul va fi afisat in textField-ul de la polinomul trei.



**Inmultirea** se realizeaza dupa urmatoarele operatii: se introduce coeficientii si puterile pentru polinomul unu. Dupa se introduce coenficientii si puterile pentru polinomul doi si se apasa butonul “\*” care efectueaza inmultirea propriu-zisa in polinomul trei. Rezultatul va fi afisat in textField-ul de la polinomul trei. Daca nu introducem valori la primul sau la al doilea polinom nu se va putea realiza operatia de inmultire.



**Derivarea** se realizeaza dupa urmatoarele operatii: se introduce coeficientii si puterile pentru polinomul unu. Dupa se apasa butonul “derivare polinom1” care efectueaza derivarea propriu-zisa in polinomul trei. Rezultatul va fi afisat in textField-ul de la polinomul trei. Acesta realizeaza derivarea primului polinom, in cel de al doilea nefiind necesar introducerea unor valori.



**Integrarea** se realizeaza dupa urmatoarele operatii: se introduce coeficientii si puterile pentru polinomul unu. Dupa se apasa butonul “integrare polinom1” care efectueaza integrarea propriu-zisa in polinomul trei. Rezultatul va fi afisat in textField-ul de la polinomul trei. Acesta realizeaza integrarea primului polinom, in cel de al doilea nefiind necesar introducerea unor valori.

**5. Concluzii**

In concluzie sunt de parere ca in acest proiect, am invatat sa proiectez si sa utilizez o interfata grafica, am aprofundat mai bine limbajul JAVA, implementarea paradigmelor OOP.

6.Bibliografie

1. Youtube
2. Wikipedia
3. <https://beginnersbook.com/>
4. <https://examples.javacodegeeks.com>