DOCUMENTATIE TEMA1

CALCULATOR DE POLINOAME

Baleanu Sorina-Diana

Grupa 302210

Profesor laborator: Stoica Anda

# Cuprins:

1.Obiectivul temei

2.Analiza problemei

3.Proiectare

4.Implementare

5.Rezultate

6.Concluzii

7.Bibliografie

# 1.Obiective

## 1.1.Obiectivul principal

Obiectivul principal al acestui proiect este realizarea unui aplicatii cu interfata grafica, in care utilizatorul sa poata efectua divese operatii pe polinoame, cum ar fi adunarea, scaderea si inmultirea a doua polinoame, sau derivarea si integrarea unui polinom. Polinoamele sunt construite din termeni numiti monom care au in componenta lor un coeficient si un grad.

## 1.2.Obiectivele secundare

Obiectivele secundare constau in pasii urmati pentru atingerea obiectvelor principale.

|  |  |  |
| --- | --- | --- |
| Obiectiv secundar | Descriere | Capitol |
| Dezvoltarea de scenarii si cazuri de utilizare | In computere, un scenariu este o naratiune a interactiunilor dintre roluri ( actori in UML ) si sistem. In cazul nostru rolul - un utilizator al aplicatiei.  În inginerie software și sisteme, un caz de utilizare este o listă de acțiuni sau pași de eveniment care definesc de obicei interacțiunile dintre un rol (cunoscut în Unified Modeling Language (UML) ca actor) și un sistem pentru a atinge un obiectiv. Actorul poate fi un sistem uman sau alt sistem extern. | 2 |
| Structuri de date | Alegerea structurilor de date care au fost folosite pentru indeplinirea obiectivului principal | 2 |
| Proiectare clase | Utilizarea unui MVC(Model-View-Controller) si a clasei Monomial si Polynomial | 3 |
| Algoritmi | Algoritimi folositi pentru implementarea operatiilor pe polinoame | 3 |
| Interfata utilizator | Programul detine o interfaţă grafica prin intermediul căreia se realizează comunicarea mult mai simpla cu operatorul uman | 3 |
| Implementarea solutiei | Se va descrie fiecare clasa alaturi de toate metodele implementate in aceasta | 4 |
| Testare | Testarea corectitudinii operatiilor efectuate pe unul sau doua polinoame | 5 |

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

2.1.Analiza problemei

Proiectarea și implementarea unui calculator polinomial cu o interfață grafică prin intermediul căreia utilizatorul poate introduce polinoame, selecteaza operația care trebuie efectuată și afișați rezultatul.

Operatiile care pot fi efectuate de program sunt

* Adunarea a doua polinoame
* Scaderea a doua polinoame
* Inmultirea a doua polinoame
* Derivarea unui polinom
* Integrarea unui polinom

2.2 .Scenarii si cazuri de utilizare

Descrierea operatiei de adunare a doua polinoame

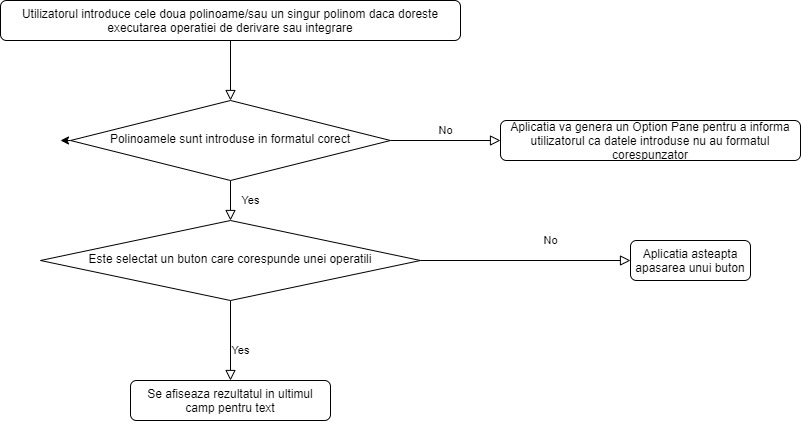
Pasul 1:Utilizatorul va introduce de la tastatura in campul text pentru polinomul 1 un polinom care are termenii de forma “coeficient x^grad”, se va proceda identic si pentru polinomul al doilea care va fi introdus in campul text din dreptul label-ului Polynomial 2.

Pasul 2 :Utilizatorul va selecta operatia matematica pe care doreste sa o efectueze cu cele doua polinoame, apasand unul din butoanele de sub campurile text pe care sunt scrise explicit numele operatiilor matematice pe care aplicatia le poate executa, adica adunare, scadere sau inmultire.In cazul care este descris acesta ar trebui sa apese butonul pe care este scris “Add”. Exista si doua butoane care corespund unor operatii matematice realizate pe un singur polinom adica derivarea si integrarea, in cazul in care utilizatorul doreste sa efectueze o asemenea operatie nu este necesara introducerea a doua polinoame, deoarece operatia se va executa pe polinomul din primul camp text.

Pasul 3:Dupa apasarea butonului corespunzator operatiei, aplicatia va lua in considerare doar termenii introdusi conform formatului mentionat mai sus, ignorand textul introdus de utilizator ce nu corespunde cu formatul specificat. In cazul in care in unul din campurile text nu este identificat niciun termen introdus corect sau nu exista nimic introdus se va deschide o noua fereastra care va atentiona utilizatorul ca datele introduse nu respecta formatul si operatia selectata nu se poate executa.

Pasul 4:In cazul in care datele sunt introduse corect, programul va genera rezultatul dorit si il va afisa in ultimul chenar de text.

Pasul 5:Polinoamele introduse initial raman in campurile text,astfel fiind posibila efectuarea unei alte operatii pe aceleasi polinoam.



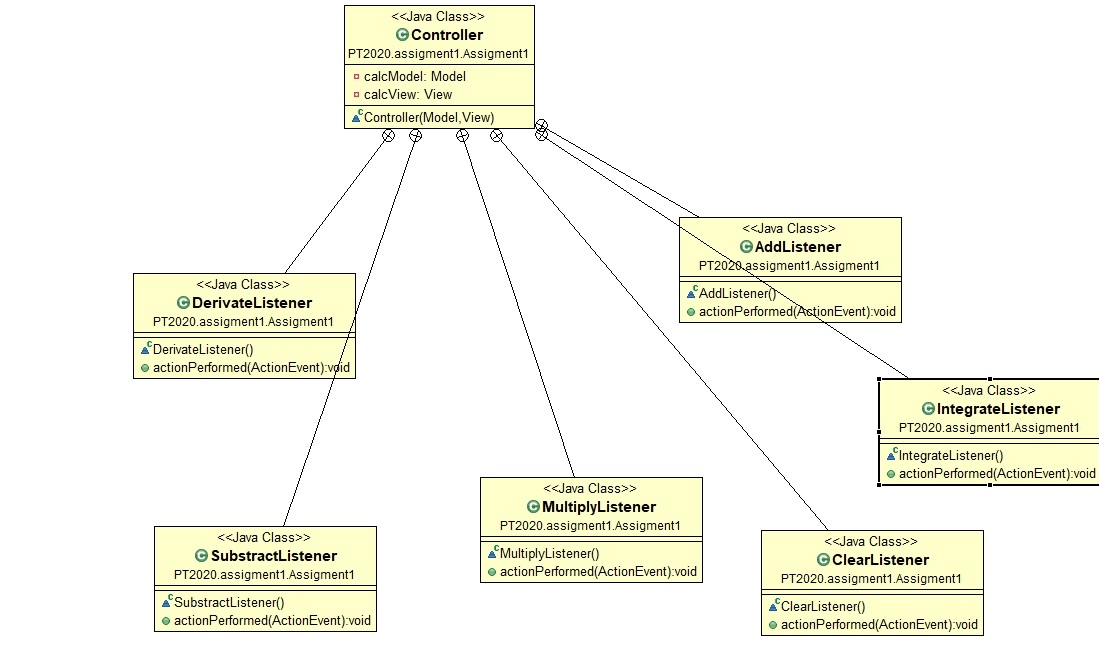
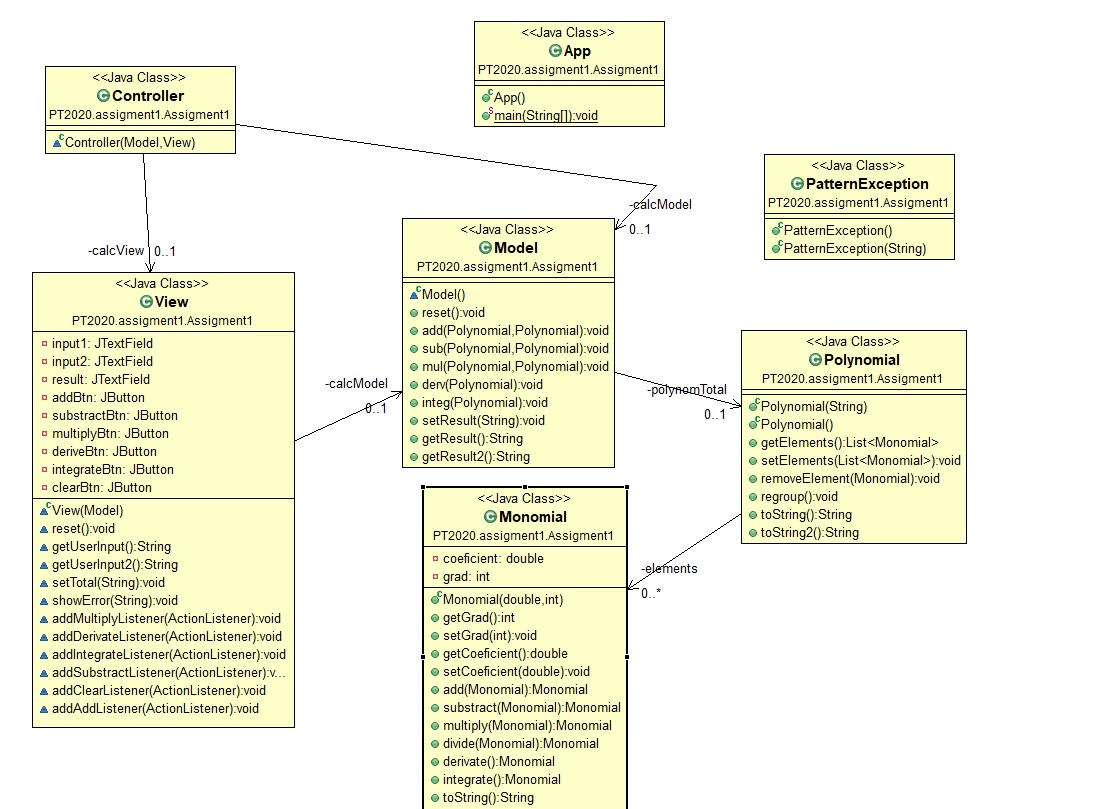
# 3.Proiectare

3.1.Decizii de proiectare

O importanta decizie de proiectare a fost utilizarea Regex-urilor la introducerea polinoamelor.Aceasta decizie permite utilizatorului de aflare a rezultatului dorit chiar daca in campurile text acesta introduce din greseala si altceva in afara de polinomul pe care vrea sa execute o anume operatie. Spre exemplu in cazul in care utilizatorul introduce de la tastatura “6x^3+5x^2pcf” , programul nu va prezenta erori si se va executa operatia pe termenii introdusi corect adica “6x^3+5x^2”

3.2.Diagrame UML

Unified Modeling Language (prescurtat UML) este un limbaj standard pentru descrierea de modele si specificatii software. Diagrama de clase UML Este folosită pentru reprezentarea vizuală a claselor și a interdependențelor, taxionomiei și a relațiilor de multiplicitate dintre ele. Diagramele de clasă sunt folosite și pentru reprezentarea concretă a unor instanțe de clasă, așadar obiecte și a legăturilor concrete dintre acestea.



3.3.Structuri de date

Structura principala folosita in acest proiect este un CopyOnWriteArrayList de monoame in clasa Polynomial. Pentru aceasta structura de date când utilizăm oricare dintre metodele de modificare - cum ar fi add () sau remove () - întregul conținut al CopyOnWriteArrayList este copiat într-o copie internă.Datorită acestui fapt simplu, se poate itera peste listă într-un mod sigur, chiar și atunci când se întâmplă modificări simultane. Prin alegerea acestei structuri de date, operatiile de adunare si scadere a celor doua polinoame au fost mai usor de implementat.

3.4.Proiectare clase si relatii intre clase

In realizarea cerintelor am folosit urmatoarele clase:

Clasa Monomial:

Aceasta clasa defineste un termen din componenta unui polinom.Monomul este compus dintr-un grad si un coeficient.Coeficientul a fost declarat ca o variabila de tip double pentru ca in urma operatiei de integrare in care coeficientul se imparte la grad poate sa rezulte un numar cu virgula.

Clasa Polynomial:

Aceasta clasa reprezinta o lista de monoame. In aceasta clasa apre si utilizarea regexului care are rolul de a separa termenii introdusi in monoame care sunt adaugate in lista existenta de monoame. Utilizarea Regexurilor permite de asemenea ignorarea termenilor care sunt introdusi prost si genereaza un avertisment in cazul in care in expresia introdusa nu se afla niciun monom.

Clasa Model:

In clasa Model sunt definite operatiile care se pot realiza pe cele doua polinoame cat si operatiile care se realizeaza pe un singur polinom. Aceasta clasa descrie de fapt modul de functionare al aplicatiei. Clasa model nu depinde de clasa View si nici de clasa Controller

Clasa View:

In clasa View este construita interfata grafica care prezinta datele utilizatorului intr-un mod usor de inteles. Aceasta clasa depinde de clasa Model

Clasa Controller:

Aceasta clasa face legatura dintre clasa Model si clasa View. Actioneaza conform actiunilor utilizatorului, are rolul de a trimite comenzi modeluilui care se ocupa cu actualizarea datelor si trimite instructiuni clasei view pentru a actualiza interfata grafica.

Clasa PatternException:

Aceasta clasa a fost creata pentru a creea o exceptie in cazul in care in campurile destinate pentru introducerea polinoamelor nu este identificat niciun monom. Astfel in cazul introducerii unei propozitii spre exemplu “Polinom de gradul 3” in interfata grafica va aparea un avertisment care va informa utilizatorul ca datele introduse nu sunt corecte.

3.5.Algoritmi

Adunare monoame

Acest algoritm presupune generarea unui nou monom care are ca coeficient suma coeficientilor rezultati prin adunarea celor 2 monoame si gradul acelasi ca si gradul celor doua monoame.

Scadere monoame

Acest algoritm presupune generarea unui nou monom care are ca coeficient diferenta coeficientiolor rezultati prin scaderea celor 2 monoame si gradul acelasi ca si gradul celor doua monoame.

Inmultire monoame

Acest algoritm presupune generarea unui nou monom care are ca coeficient produsul coeficientiolor rezultati prin inmultirea celor 2 monoame si gradul reprezinta suma gradelor celor 2 monoame.

Derivare monom

Derivarea monomului presupune ca primul coeficient sa fie produsul dintre coeficientul monomului ce trebuie a fi derivat si gradul acestuia, noul monom va avea gradul egal cu gradul curent minus 1

Integrare monom

La integrarea monomului coeficientul nou va fi coeficentul vechi impartit la gradul actual minus unu, iar coeficientul nou este cel actual incrementat cu valoarea 1

Adunare polinoame

Adunarea a doua polinoame este implementata prin parcurgerea in paralele a celor doua polinoame cu doua foreach-uri, la fiecare iterare se verifica daca gradul monoamelor curente este egal.In cazul in care monoamele au acelasi grad se efectueaza pe aceste doua monoame operatia de adunare a monoamelor si se sterg din lista de elemente a polinoamelor parcurse si rezultatul adunarii se adauga in lista elements declarata ca variabila locala a metodei .Dupa ce s-a terminat parcurgerea celor 2 polinoame si adunarea termenilor care au acelasi grad se vor mai face cate o parcurgere separata pentru fiecare polinom in lista de elemente pentru a adauga la polinomul rezultat si termenii care nu au suferit in urma operatiei de adunare pentru fiecare polinom. La finalul metodei se va seta lista de elemenete a polinomului total

Scadere Polinoame

Algoritmul de scadere este implementat similar cu cel de adunare singurele diferente fiind ca in parcurgerea simultana se foloseste operatia de scadere implementata in clasa monom, iar la parcurgeera de la final a celui de-al doilea polinom termenii care nu au suferit modificari in urma scaderii sunt adaugati cu coeficentul lor inmultit cu –1.

Inmultire Polinoame

Operatia de inmultire a celor doua polinoame presupune parcurgerea simultana a celor doua polinoame, in cazul inmultirii nu mai este necesar sa se verifice gradul monoamelor deoarece operatia de inmultire se va face intre monoame indiferent de gradul lor.Fiecare monom din primul polinom trebuie inmultit cu fiecare monom din polinomul al doilea.

Dupa efectuarea acestei inmultiri se va utiliza o functie de regrupare a termenilor cu acelasi grad care e posibil,sa rezulte in urma inmultirii a doua polinoame.

Derivare Polinom

Algoritmul de derivare presupune o simpla parcurgere a polinomului ce trebuie derivat in care fiecarui termen i se aplica metoda de derivare conceputa in clasa monom.

Integrare Polinom

Identic cu algoritmul de derivare,doar ca se aplica operatia de integrare a monomului.

3.6.Interfata utilizator

Interfața grafică sau Graphical User Interface este o [interfață cu utilizatorul](https://ro.wikipedia.org/wiki/Interfa%C8%9B%C4%83_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. Folosim o interfata grafica User-Friendly cu scopul de a putea fi folosit acest calculator de polinoame si de persoane non-specializate.

Elementele care alcatuiesc interfata grafica construita in clasa View sunt:

Frame-ul

Frame-ul este o fereastră de nivel superior, cu un titlu și un chenar.Un frame, implementat ca o instanță a clasei JFrame, este o fereastră care are decorațiuni, cum ar fi un chenar, un titlu și acceptă componente de buton care închid fereastra. Aplicațiile cu GUI includ de obicei cel puțin un cadru. Uneori, aplicațiile folosesc și cadre.

Panel-ul

Clasa JPanel oferă containere de uz general pentru componente ușoare. Am folosit un Panel in care am adaugat componentele ce sunt prezentate mai jo, anume campurile text si butoanele.

Text Field-urile

Text Field-urile sunt spatii dreptunghiulare in care se pot introduce date de la tastatura.Cu ajutorul lor se poate afisa rezultatul fara a se introduce date de la tastatura. Ele sunt implementate ca instante ale clasei JTextField.

Sunt utilizate 3 chenare text, 2 sunt folosite pentru introducerea polinoamelor de la tastatura si unul este folosit pentru afisarea rezultatului obtinut in urma efectuarii operatiei dorite

* + userInputTf - campul de text pentru introducerea primului polinom
  + userInpuTf2 –campul de text folosit pentru introducerea polinomului 2
  + TotalTf-campul de text in care se afiseaza rezultatul final

Butoanele

Butoanele sunt instante ale clasei JButton, rolul lor este de a efectua diverse comenzi atunci cand sunt apasate de catre utilizator in interfata grafica. Fiecare buton are inscriptionat pe el un text care indica utilizatorului actiunea ce urmeaza a fi efectuata de catre program in momentul apasarii acestuia.

In aceasta aplicatie sunt folosite 6 butoane care corespund cu numarul operatiilor implementate dar si un buton folosit pentru resetarea rezultatului la 0.

# 5.Implementare

Clasa Monomial:

In aceasta clasa au fost implementate urmatoarele metode:

* Metodele de get si set ale coeficientului si gradului unui monom
* Metoda add- metoda in care este definita adunarea dintre doua monoame, aceasta metoda returneaza un nou monom , rezultatul adunarii monomului curent cu monomul primit ca parametru al metodei
* Metoda substract – construita precum metoda add, doar ca pentru scaderea a doua monoame
* Metoda multiply– asemenea metodelor de mai sus, insa pentru inmultire
* Metoda derivate- in aceasta metoda este returnat un nou monom care este rezultat al derivarii monomului care apeleaza metoda
* Metoda integrate- construita ca metoda derivate, doar ca pentru operatia de integrare a unui monom
* Moetoda toString-este definit modul de afisare al monoamelor, in cazul in care coeficientul este zero monomul respectiv nu mai este afisat, in cazul in care monomul este de gradul 0, monomului respectiv ii este afisat doar coeficientul,iar pentru gradul 1 se va afisa doar coeficientul si x.Restul monoamelor vor fi afisate conform modelului “coeficient x^grad”

Clasa Polynomial:

In clasa Polynomial au fost implementate urmapoarele metode:

* Metoda de get si set pentru lista de monoame
* Metoda toString-folosita pentru concatenarea monoamelor la afisare

Clasa Model:

In clasa Model sunt definite operatiile prin urmatoarele metode:

* Metoda reset-reseteaza valoarea polinomului rezultat la 0
* Metoda add-se realizeaza adunarea a doua polinoame ,folosindu-se operatia de adunare implementata in clasa de monom si creeaza un nou polinom rezultat
* Metoda sub-la fel ca metoda add doar ca pentru scadere
* Metoda mul-asemenea metodei add doar ca pentru inmultire
* Metoda derv-aceasta metoda calculeaza derivata unui polinom ,utilizand metoda de derivare implementata in clasa monom si creeaza unu nou polinom care are ca rezultat polinomul rezultat in urma derivarii
* Metoda integ-ca si in metoda derv doar ca pentru operatia de integrare a unui polinom
* Metoda setValue- setaza valoarea polinomului total pornind de la un String dat
* Metoda getValue – convereste polinomul intr-un sting apeland metoda toString din clasa Polynomial

Clasa View:

In clasa View avem implementarea urmatoarelor metode

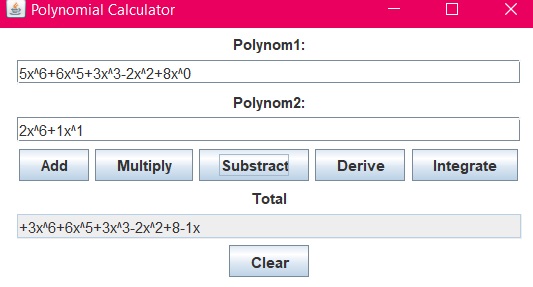
* + Metoda reset-resteaza valoarea care va fi afisata in campul text pentru total la valoarea 0
  + Metoda getUserInput-metoda care preia textul introdus in primul camp text
  + Metoda getUserInput2- metoda care preia textul introdus in al doilea camp text
  + Metoda setTotal-seteaza totalul in campul de text in care aceste trebuie sa fie afisat
  + Metoda ShowError- deschide un Option Pane in cazul uneie evantuale erori
  + Metoda addAddListener-adauga action listenerul pentru butonul de adunare
  + Metoda addSubstractListener- adauga action listenerul pentru butonul de adunare
  + Metoda addMultiplyListener adauga action listenerul pentru butonul de adunare
  + Metoda addDerivateListener adauga action listenerul pentru butonul de adunare
  + Metoda addIntegrateListener adauga action listenerul pentru butonul de adunare
  + Metoda addClearListener adauga action listenerul pentru butonul de adunare

Clasa Controller:

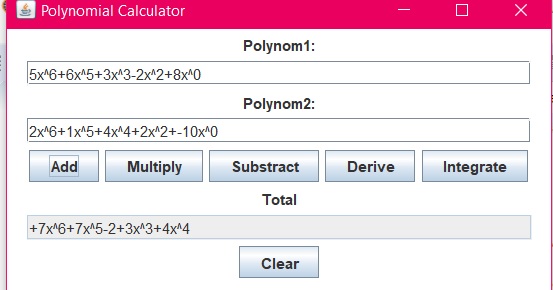
In aceasta clasa nu este implementata nicio metoda, aici gasindu-se doar clasele interne care implementeaza Action Listenerele pentru butoanele care reprezinta operatiile.

# 6.Rezultate

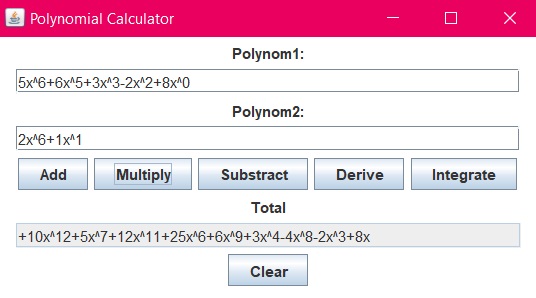
Adunare



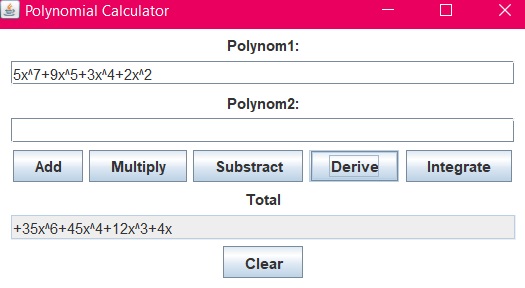
Scadere



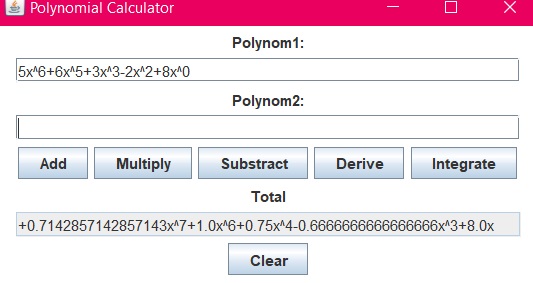
Inmultire



Derivare



Integrare



# 7.Concluzii

In concluzie realizarea acestui proiect a fost foarte utila , amintind de o mare parte din principalele concepte ale porgramarii orientate pe obiecte invatate semestrul trecut.

O dezvoltare ulterioara care ar putea fi facuta pentru ca aplicatia sa fie mai usor de utilizat ar putea fi spre exemeplu atentionarea mult mai clara a utilizatorului in cazul in care datele sunt introduse gresit, adica programul sa ii spuna utilizatorului exact care termen a fost introdus prost si sa vina cu o sugestie de corectare.

Alte dezvoltari ale aplicatiei ar putea fi realizarea de operatii sucesive sau posibilitatea de a se efectua operatii pe mai mult de doua polionoame deodata.

# 8.Bibliografie

* <http://users.utcluj.ro/~igiosan/Resources/POO/Lab/12-Testarea_Unitara.pdf>
* <https://www.baeldung.com/java-copy-on-write-arraylist>
* <https://www.vogella.com/tutorials/JavaRegularExpressions/article.html>
* <https://stackoverflow.com/questions/36490757/regex-for-polynomial-expression>
* <https://stackoverflow.com/questions/28859919/java-regex-separate-degree-coeff-of-polynomial>