Sistem de procesare a polinoamelor

Nume: Adam Cristina-Ioana

Grupa:30224

Facultatea de Automatica si Calculatoare

Specializarea: Calculatoare

Cuprins:

[1.Obiectivul temei 3](#_Toc509572750)

[2. Analiza problemei, asumptitii, scenarii, cazuri de utilizare, erori 3](#_Toc509572751)

[2.1 Analiza problemei 3](#_Toc509572752)

[2.2 Asumptitii 4](#_Toc509572753)

[2.3. Scenarii 5](#_Toc509572754)

[3. Proiectare(decizii de proiectare, diagram UML, structure de date, proiectare clase, interfete, relatii, packages, algoritmi, interfata utilizator, modul de tratare a erorilor) 9](#_Toc509572755)

[4. Implementare 12](#_Toc509572756)

[5. Testare 13](#_Toc509572757)

[6. Rezultate 13](#_Toc509572758)

[7. Cocluzii 13](#_Toc509572759)

[8.Bibliografie 13](#_Toc509572760)

# 1.Obiectivul temei

Obiectivul principal al temei este de a implementa o aplicatie care efectueaza operatii pe polinoame, adica un modul in care acestea pot fi create si manipulate prin intermediul paradigmei orientate pe obiecte. Operatiile efectuate sunt: adunare, scadere, inmultire, impartire, derivare si integrare.

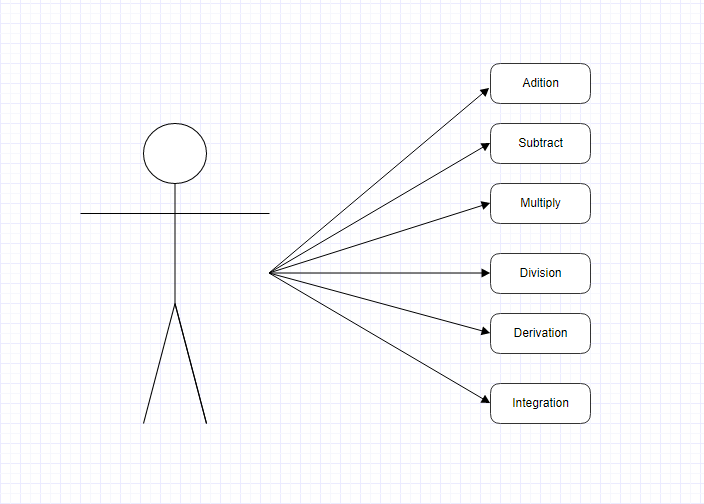
Obiectivele secundare ale proiectului sunt urmatoarele: datorita interfatei grafice a acestui proiect se pot testa intr un mod foarte simplu toate operatiile. Aplicatia pemite efectuarea tuturor operatiilor intr-un mod cat mai simplu. Prin apasarea butonului de “Adding” ,noua ni se preia din JTextField-ul nostru primul polinom ,asupra caruia se va face o parsare ,adica o impartire in monoame.Un polinom este de forma:2x^2+1x^1+4x^0 , monoamele acestui polinom sunt :2x^2, 1x^1 si 4x^0. Pentru a folosi coeficientul si puterea acestor polinoame o sa ne luam un float pentru coeficient si un int pentru putere ,pe care o sa le adaugam la lista noastra de monoame. Adunarea a doua polinoame se face: se cauta polinoamele care au aceeasi puetere si li se aduna coeficientii , dupa care se adauga intr-o noua lista numita rezultat , dupa care se iau termenii ramasi din prima lista si se adauga in lista rezultat, iar la fel se procedeaza si pentru a doua lista.La operatia de scadere se aplica algoritmul de adunare singura diferenta fiind ca prima data se va face 0 minus al doilea polinom. Inmultirea presupune parcurgerea ambelor liste si inmultirea coeficientiilor care se vor adauga la lista rezultat, iar puteriile vor fi adunate si de asemenea si ele vor fi adaugate la lista rezultat.Operatia de derivare se face prin inmultirea coeficientului cu puterea, dupa care se va face putere-1, iar integrarea se va face prin pastrarea coeficientului si la putere se va adauga 1 iar tot monomul va fi supra putere+1.

Toate aceste operatii vor fi dicutate mai pe larg in subpunctul 4 din cuprins.

# 2. Analiza problemei, asumptitii, scenarii, cazuri de utilizare, erori

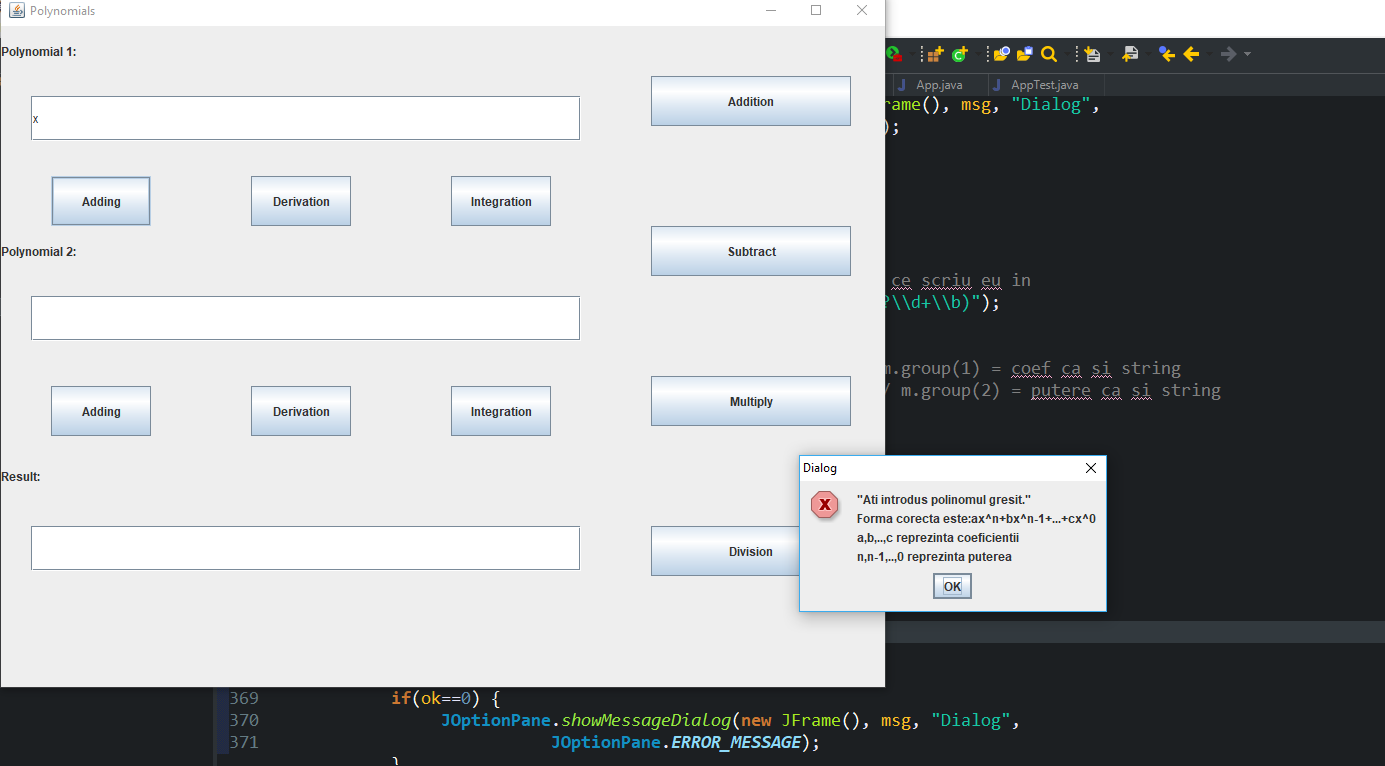
## 2.1 Analiza problemei

Diagrama use case ne prezinta toate operatiile pe care le poate efectua un utilizator. Utilizarea diagramelor use case nu este absolute necesara pentru a scrie o specificatie cu use case-uri, aceasta este utila pentru a crea o imagine generala asupra sistemului.



## 2.2 Asumptitii

In momentul in care utilizatorul nu introduce polinomul in forma correct ii va aparea un mesaj, care il atentioneaza ca nu a facut acest lucru cum trebuie. De exemplu introducem: un caracter sau nu avem forma 1x^0 ,si scriem doar x ,atunci va aparea acel mesaj. Mesajul este urmatorul:

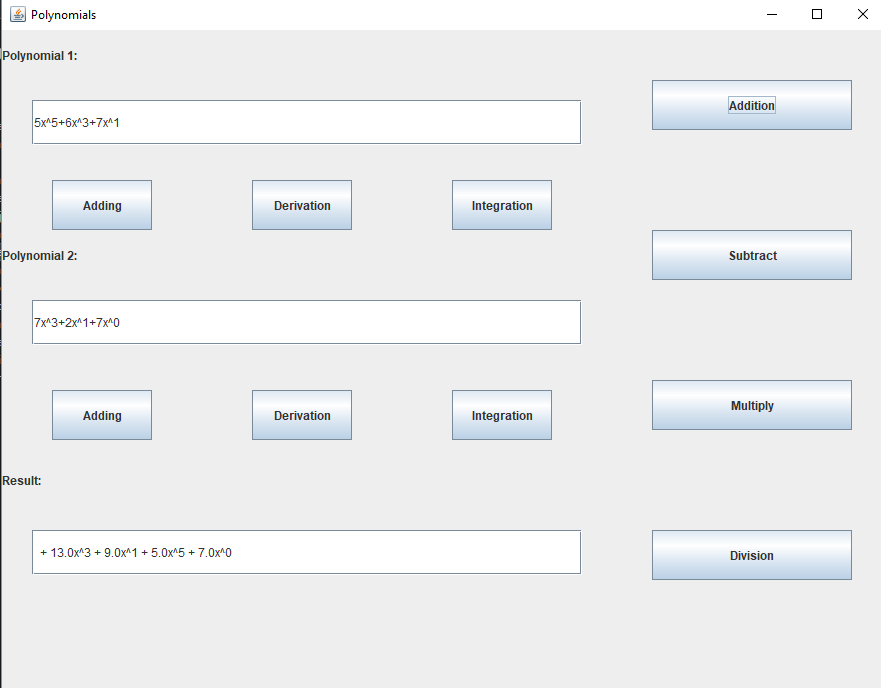


In momentul in care utilizatorul introduce polinomul in JTextField, acesta trebuie sa apese butonul de ”Adding” deoarece parsarea polinomului se face doar in momentul in care noi efectuam o actiune asupra acelui buton.

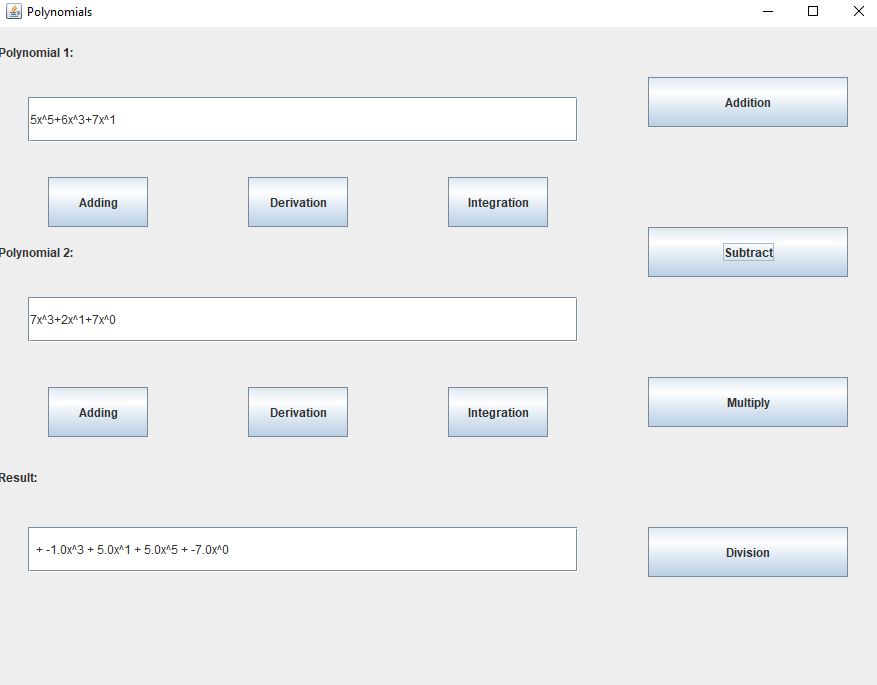
## 2.3. Scenarii

In continuare vom lua doua polinoame si vom efectua operatii asupra lor

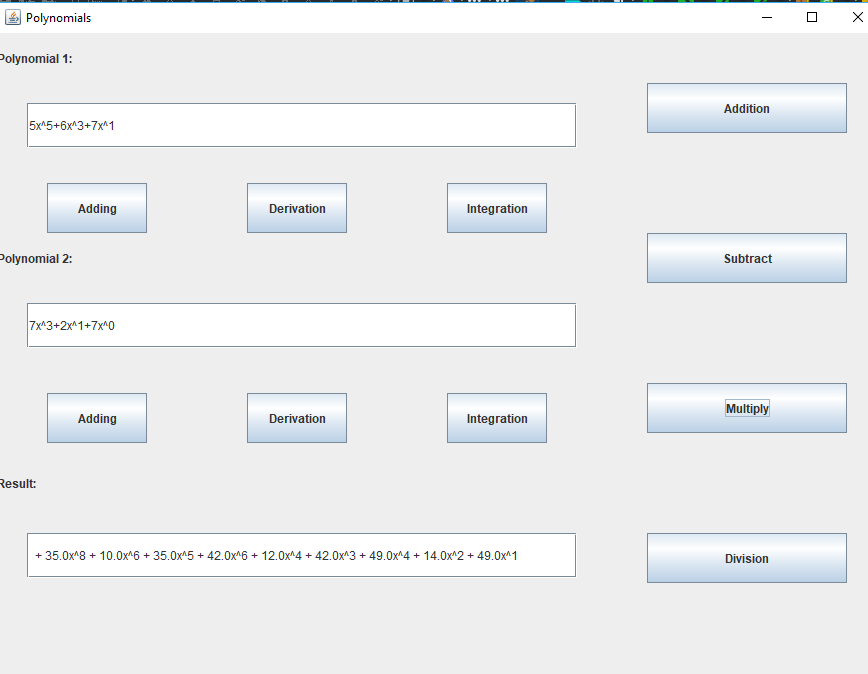
1. Adunare



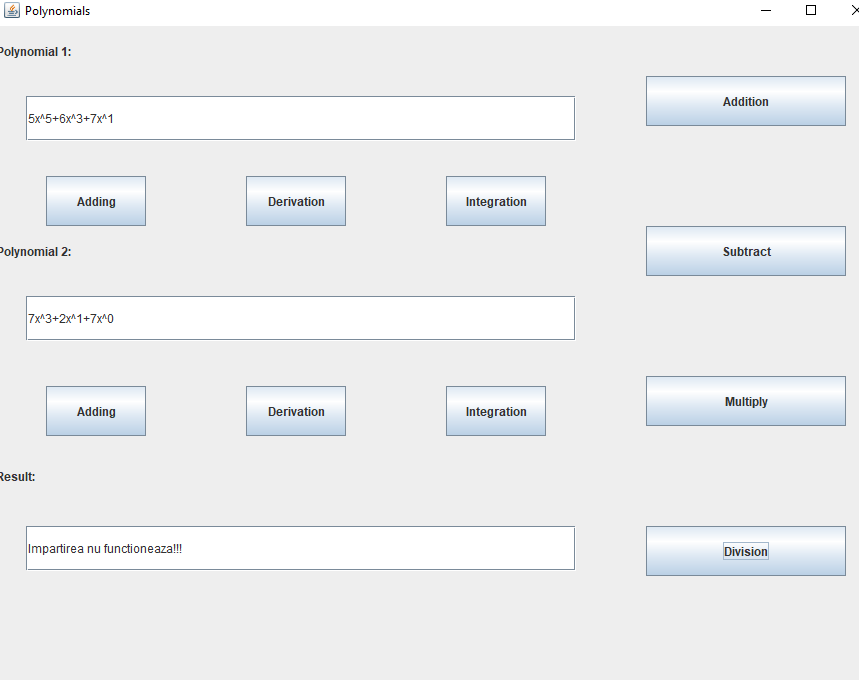
1. Scadere



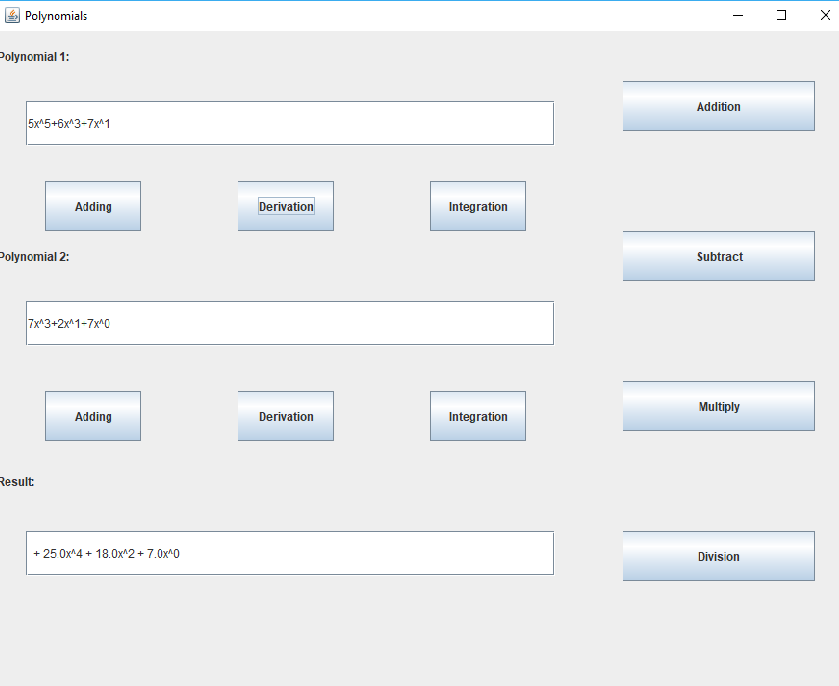
1. Inmultire



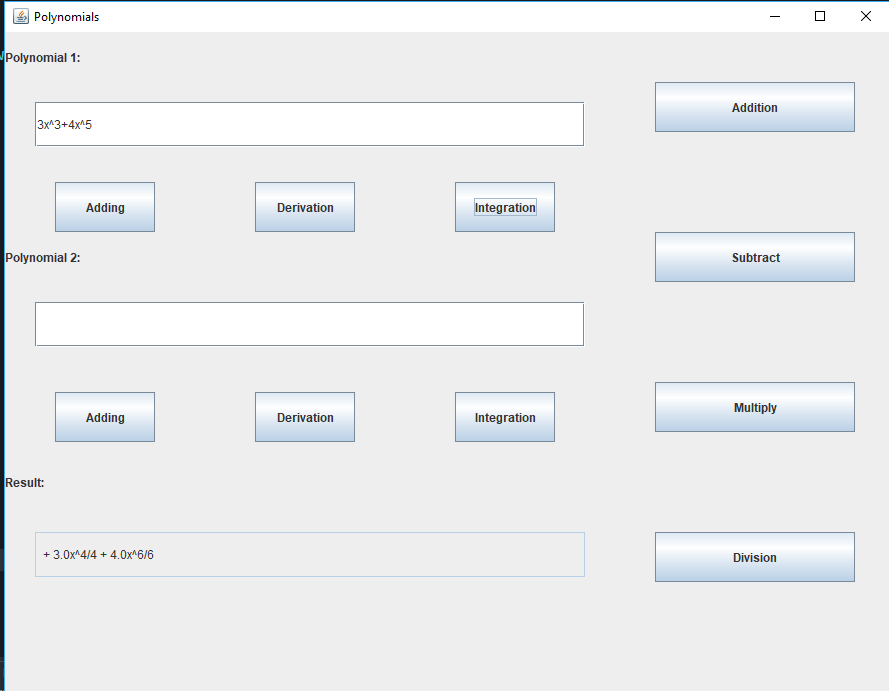
1. Impartirea



1. Derivarea



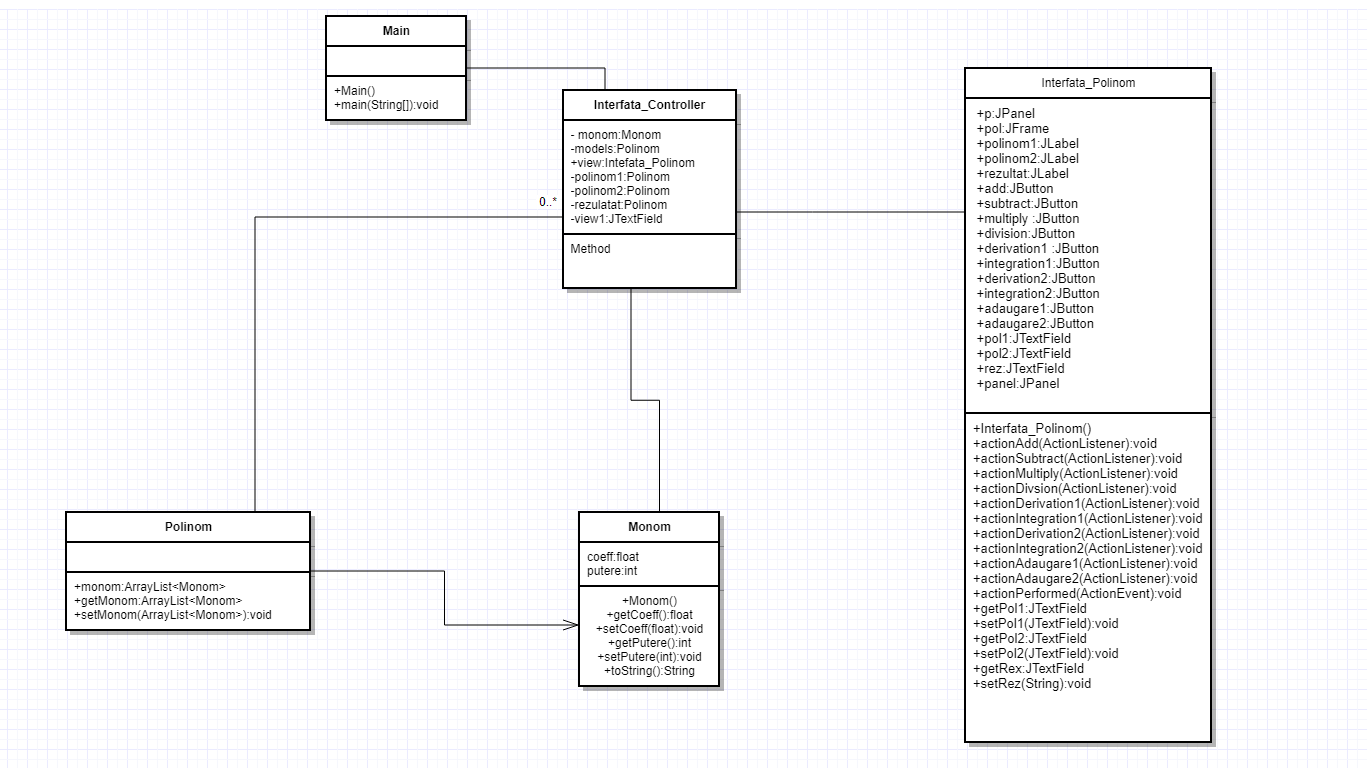
1. Integrare



Cazurile de utilizare au fost prezentate mai sus.Eroare apare doar in momentul in care utilizatorul introduce gresit polinomul, iar in acel caz utilizatorul trebuie sa scrie polinomul in forma corecta.Se presupune ca utilizatorul stie forma polinomului, prin urmare stie sa introduca corect acel polinom.

# 3. Proiectare(decizii de proiectare, diagram UML, structure de date, proiectare clase, interfete, relatii, packages, algoritmi, interfata utilizator, modul de tratare a erorilor)

O analiza mai amanuntita asupra aplicatie este prezentata de diagramele UML. UML(Unified Modelling Language) dupa cum ii spune si numele este un limbaj. Acestea sunt folosite pentru reprezentarea desfasurarii unei activitati in rularea programului. Acestea au in componenta urmatoarele elemente: numele clasei, atribute si metode. Diagrama UML pentru acest proiect este urmatoarea:



Dupa cum se observa avem 5 clase: Main, Polinom, Monom, Interfata\_Controller si Interfata\_Polinom. In clasa Main se se declara un obiect de tipul Interfata\_Controller. Clasa Interfata\_Controller este cea care controleaza activitatea aplicatiei. Aceasta contine ActionEvent-uri pentru butoanele care efectueaza adunare, scadere, inmultire, impartire, derivare, integrare si adaugare. In momentul in care utilizatorul apasa butonul de adding se va adauga polinomul in lista si se va face parsarea, in caz contrar aplicatia nu va face nicio actiune cu polinoamele. Aceasta actiune se va face pentru ambele polinoame. Dupa acest pas polinoamele noastre sunt adaugate ca liste de monoame, deci noi vom lucra in continuare cu acele liste,care ne contin atat puterea, cat si coeficientul.Operatiile vor fi descrise mai amanuntit la subpunctul 4. Pentru acest proiect am folosit MVC(Model View Controller), adica am format pachete in care am introdus clase. Model reprezinta partea logica a aplicatie. Acesta cuprinde calasa Polinom si Monom. In clasa Polinom se declara o lista de Monoame. Pentru clasa Monoame am creat gettere si settere si am convertit in string polinomul nostru. In pachetul controller am declarat o clasa Interfata\_Controller . In aceasta clasa am declarata ActionListener pentru butoane, iar in momentul in care se apasa butonul se efectueaza operatia pe care o doreste utilizatorul. In pachetul View se afla clasa Interfata\_Polinom, in aceasta clasa avem codul pentru interfata noastra.

Algoritmii utilizati pentru efectuarea operatiilor:

1.Adunarea

In actionAdd se va lua un vector si un contor in care vor fi stocata puterile comune ale celor doua polinoame. Vom lua doua for-uri pentru parcurgerea celor 2 liste si vom verifica daca puterile celor doua polinoame sunt egale, daca sunt egale vom adauga in lista rezultat coeficientii lor adunati si puterea. In acesta moment in lista rezultat avem operatia de adunare doar pentru polinoamele care au aceiasi coeficienti. Pentru a pune si monoamele care au puteri diferite vom parcurge pe rand listele cele doua polinoame si vom lua un ok=0, daca ok ramane tot 0 insemna ca polinoamele au puteri diferite, deci le vom adauga in lista rezultat,daca ok devine 1 inseamna ca au puterile egale. Pentru afisarea rezultatului in JTextField-ul rezultat o sa luam un string ,care initial nu are nimic in el si cu functia ‘concat’ vom pune forma corecta a stringu-ul ,deci prima data vom pune semnul de adunare + coefficient+ x^ + putere, aceasta afisare o vom folosi pentru toate operatiile.

2.Scaderea

Operatia de scadere foloseste acelasi algoritm ca si operatia de adunare singura diferenta fiind ca inainte de a efectua operatia de adunare vom lua un for pentru a parcurge a doua lista si vom face 0- polinom 2 pentru a schimba semnul tuturor coeficientiilor, Afisare este identica ca si pentru operatia de adunare.

3.Inmultire

In acest algoritm parcurgem ambele liste si vom inmulti coeficientii, iar puteriile le vom aduna. Afisarea este identica cu afisarea operatiilor de mai sus.

4. Impartirea

Algoritmul de impartire nu a fost implementat. Va aparea in rezultat un mesaj care semnaleaza faptul ca aceasta operatie nu a fost implementata, deci nu vom putea efectua nicio operatie de impartire.

5. Derivarea

Algoritmul de derivare poate fi aplicat atat pentru primul polinom, cat si pentru al doilea polinom, iar rezultatul va fi pus tot in lista rezultat. Algoritmul de derivare presupune parcurgerea polinomul si inmultirea coeficientului cu puterea dupa care se va face putere-1. In cazul in care puterea polinomul este 0 vom face face atat coeficientul cat si puterea 0, in cazul in care nu vom face acest lucru, in momentul in care utilizatorul introduce un singur monom care are puterea 0 nu vom avea niciu un rezultat in lista noastra rezultat, acest lucru se intampla din cauza ca inainte de a efectua operatiile pe coeficient si putere am pus conditia ca puterea sa fie mai mare de 0.

6. Integrarea

Algoritmul de integrare este aplicat atat pentru primul polinom, cat si pentru al doilea polinom. Prima data se aduna la putere 1, dupa care se va face o impartire intre coeficient si noua valoare a puterii.Rezultatul este adaugat in lista rezultat dupa care este aduga in JTextField-ul din interfata.

Avem doua butoane de adaugare, unul pentru primul polinom si unul pentru cel de al doilea polinom. In ActionEvent-ul acestor butoane se face parsarea polinomului in monoame si apoi formarea listelor cae contin acele monaome. Metoda de parsare este aceiasi pentru ambele polinoame, Intr-un string input eu copiez continutul din primul string ,respectiv al doilea string. Se ia un pattern in care se va trece forma generala a unui polinom, aceasta forma este data de functia regex, dupa care acestui pattern i se transmite matcher,iar in matcher avem poliomul nostru. Daca expresia noastra se potriveste cu expresia din pattern atunci o sa grupam polinomul in functie de coeficient si putere. Initial valorile mele inca sunt string-uri, deci trebuie sa imi convertesc coeficientul la float si puterea la integer, dupa care creem un monom nou in care adaugam coeficientul si puterea. Daca utilizatorul nu a introdus bine polinomul o sa ii apara un mesaj de eroare. Dupa fiecare operatie lista mea treuie sa o golesc, acest lucru se face cu functia removeAll.

Modul de testare se face prin aparitia unui nou Jframe in care o sa imi apara un mesaj si o informatie de cum ar trebui sa introduc functia.

Modul de tratare a erorilor: in aplicatia mea am folosit afisarea de mesaje,insa putem sa avem si de tipul try si catch, pentru a-mi arunca o exceptie in caz de eroare. Pentru derivare am luat in considerare cazul in care avem x^0, iar printr-o ramura if am testat acest caz si am facut atat coeficientul cat si puterea 0.Am luat in considerare acest caz deoarece daca eu introduce doar un monom care are puterea 0 nu o sa imi afiseze nimic, insa avem un dezavantaj deoarece la polinoamele cu mai multe monoame in care avem inclus si acest monom o sa imi afiseze 0.0x^0, nu e tocmai practic, dar insa este mai bine asa decat sa nu imi apara nimic in rezultat.La integrare puteam sa lasam sub forma de fractie coeficientul pe putere+1 insa avand in vedere ca am lucrat cu coeficienti de tip float nu a trebuit sa fac acest lucru si am facut acea impartire.

# 4. Implementare

Dupa cum am mentionat si la subpunctele anterioare eu am 5 clase incluse in 4 pachete. Proiectul este de tipul MavenProject, pentru a putea realize JUnit Test pentru fiecare operatie. Eu nu am facut acest lucru insa o sa prezint pe scurt ce presupune un JUnit Test. Vom lua 2 polinoame(cum vrem noi) si vom scrie rezultatul asteptat de noi, dupa care se face un assertEquals intre rezultatul nostru si rezultatul generat de aplicatia implememntata de mine. Inainte de a scrie aceasta metoda vom scrie @Test si vom rula cu JUnit, daca raspunsul este true inseamna ca a trecut testul.

Intefata grafica este create in pachetul View. Aceasta are in componenta ei :un frame, un panel, 3 label-uri, 10 butoane si 3 TextField-uri. In frame am adaugat panel-ul, care are aceiasi dimensiune cu frame-ul, acest lucru este mai practic, dupa care noi am lucrat cu panel, pe acesta am pus butoanele, label-urile si textfield-ul.

Pentru a nu primi mesaje de eroare voi specifica in continuare modurile de utilizare ale aplicatiei:

1.Se va introduce polinomul sub forma 1x^3+4x^2+5x^0, acesta este un exemplu luat random, am vrut sa evidentiez faptul ca trebuie sa avem intotdeauna la coeficient un numar, chiar daca acesta este 1 . Desi din punct de vedere matematic nu este ceva obligatoriu, aici trebuie sa facem acest lucru. Un alt aspect este legat de ultimul termen care trebuie sa aiba neaparat scris x^0(in cazul in care acesta este ultimul termen), aceasta regula se aplica si pentru x^1. Motivul pentru care trebuie introduse polinoamele in acest mod sunt: am uitlizat functia regex, care se aplica doar pentru polinoamele de acesta forma. Nu este tocmai practic acest lucru, insa se vor putea face schimbari in aplicatie.

2.Intotdeauna dupa introducerea polinomului se apasa butonul de “Adding” de la primul polinom, respective de la cel de-al doilea.

3. Daca ati urmat pasi de mai sus, acum se vor putea efectua toate operatiile pentru acele polinoame(execeptie impartirea,insa si daca se apasa pe butonul de impartire o sa apara un mesaj care imi spune ca nu functioneaza impartirea).

# 5. Testare

Partea de testare nu a fost realizata, insa am explicat la subpunctul anterior ce presupune partea de testare.

# 6. Rezultate

La sectiunea 2.3 am introdus screenshot-uri in care avem efectuata fiecare operatie. De remarcat cateva observatii:

1. Polinoamele mele daca nu sunt in ordine descrescatoare, adica de la cea mai mare putere la cea mai mica putere, rezultatele nu vor fi puse in ordine descrescatoare. La adunare si scadere nu sunt puse in ordine din motivul ca sunt cautate puterile comune care se aduaga prima data in lista ,dupa care se adauga elementele din polinomul 1 si dupa aceea elementele din polinomul 2. Aceasta problema va fi rezolvata pe parcurs prin adaugarea unei metode de sortare.
2. Rezultatele obtinute se scriu sub forma de numere de tipul float, chiar daca acestea sunt intregi va aparea de exemplu 2.0. Aceasa problema de asemenea poate fi modificata.

# 7. Cocluzii

Dupa realizarea acestui proiect am aprofundat partea cu ArrayList si am reusit sa le folosesc pe tot parcursul proietului listele, am reusit sa scriu un proiect de la 0 fara a folosi WindowBuilde(o aplicatie care te ajuta sa iti creezi interfata folosind doar drag and drop). In momentul in care te apuci de implementat acest proiect ai cam o singura idee de cum va arata si cum vei implementa operatiile , insa pe parcurs o sa realizezi faptul ca de fapt ai o multime de idei.

# 8.Bibliografie

<https://stackoverflow.com/questions/36490757/regex-for-polynomial-expression>

<https://docs.oracle.com/javase/7/docs/api/java/util/ArrayList.html>

<https://www.youtube.com/watch?v=sNEcpw8LPpo>