DOCUMENTAȚIE

Tema 1

NUME STUDENT: Stroia Anton-Călin

GRUPA: 30225

CUPRINS

1.Obiectivul temei

Obiectivul principal al acestui proiect este de a implementa un calculator de polinoame care poate să execute diferite operații, precum: Adunare, Scăderea, împartirea, Înmulțirea, Derivarea și Integrarea.

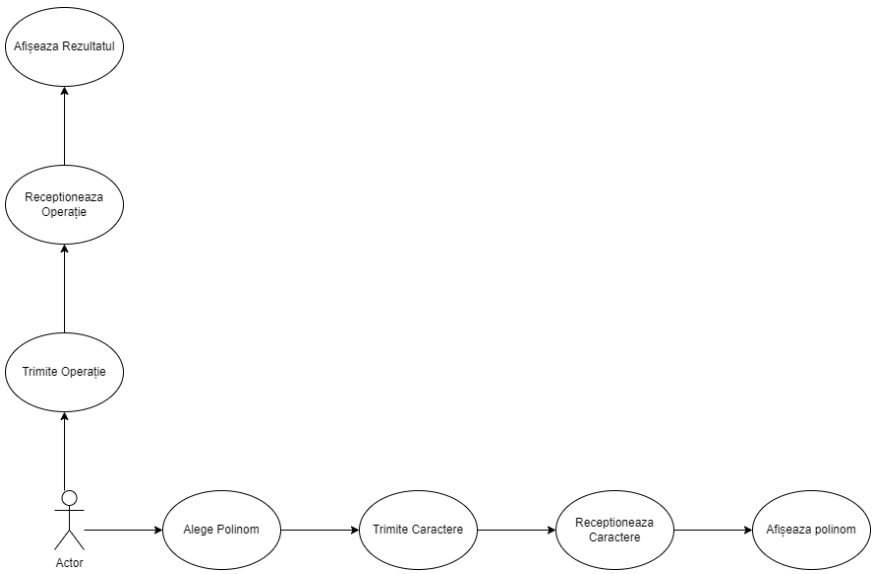
Pentru a se putea acest proiect, următorii pași au fost necesari de parcurs:

a. Reprezentarea unui polinom, cu ajutorul unei structuri de date predefinite in Java, și anume HashMap
b. Formarea unei clase specifice numită “Operații” pentru a putea declara și descrie operațiile posibile pe polinoame
c. Construirea unei clase specifice pentru reprezentarea, dacă e cazul unei fracții.
d. Implementarea unei interfețe grafice cu scopul de a face posibila interacțiunea cu un utilizator.

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

Acest calculator funcționează doar in cazul introducerii corecte a structurii unui polinom, de exemplu " 1*x^2+2*x^1+1*x^0 ". În urma introducerii adecvate a celor doua polinoame se poate apăsa butonul pentru efectuarea operațiilor dorite. Automat dupa apăsarea unui buton de operație se va afișa in câmpul specific rezultatul calculului. În cazul introducerii neadecvate a unui polinom, de exemplu "x^2+x^1+2" programul se va opri.

Utilizatorul este nevoit aleagă care polinom urmează să fie scris, apoi să introducă pe rând fiecare caracter al polinomului. Pe măsură ce utilizatorul apasă butoanele specifice caracterelor dorite, aceste caractere sunt recepționate și afișate pe rând în câmpul specific polinomului selectat anterior. După introducerea polinoamelor, utilizatorul are posibilitatea de a calcula diferite operații(Adunare, Scădere, Înmulțire, Împărtire, Derivare și Integrare), prin apăsarea butonului specific. De îndată ce butonul de operație este apăsat, operația este receptionată, afișând ulterior polinomul rezultat.



3.Proiectare

Pentru putea reprezenta cu ușurință un polinom a fost nevoie de folosirea unor structuri de date specifice, precum HashMap, ArayList. Introducerea cu ușurință a elementelor în HashMap, din polinomul introdus inițial sub formă de String, s-au luat pe rând coeficientul și puterea fiecărui element din polinom și s-a introdus în această ordine în listă. Ulterior se introduce în HashMap argumentele fiecărui element, pe prima pozitie aflându-se puterea iar pe a doua coeficientul.

Implementarea operatiilor s-a realizat fiecare după un anumit algoritm specific.

Adunarea, prin adunarea fiecarui coeficient al celui de-al doilea polinom la primul prin asocierea fiecărui grad al elementelor.

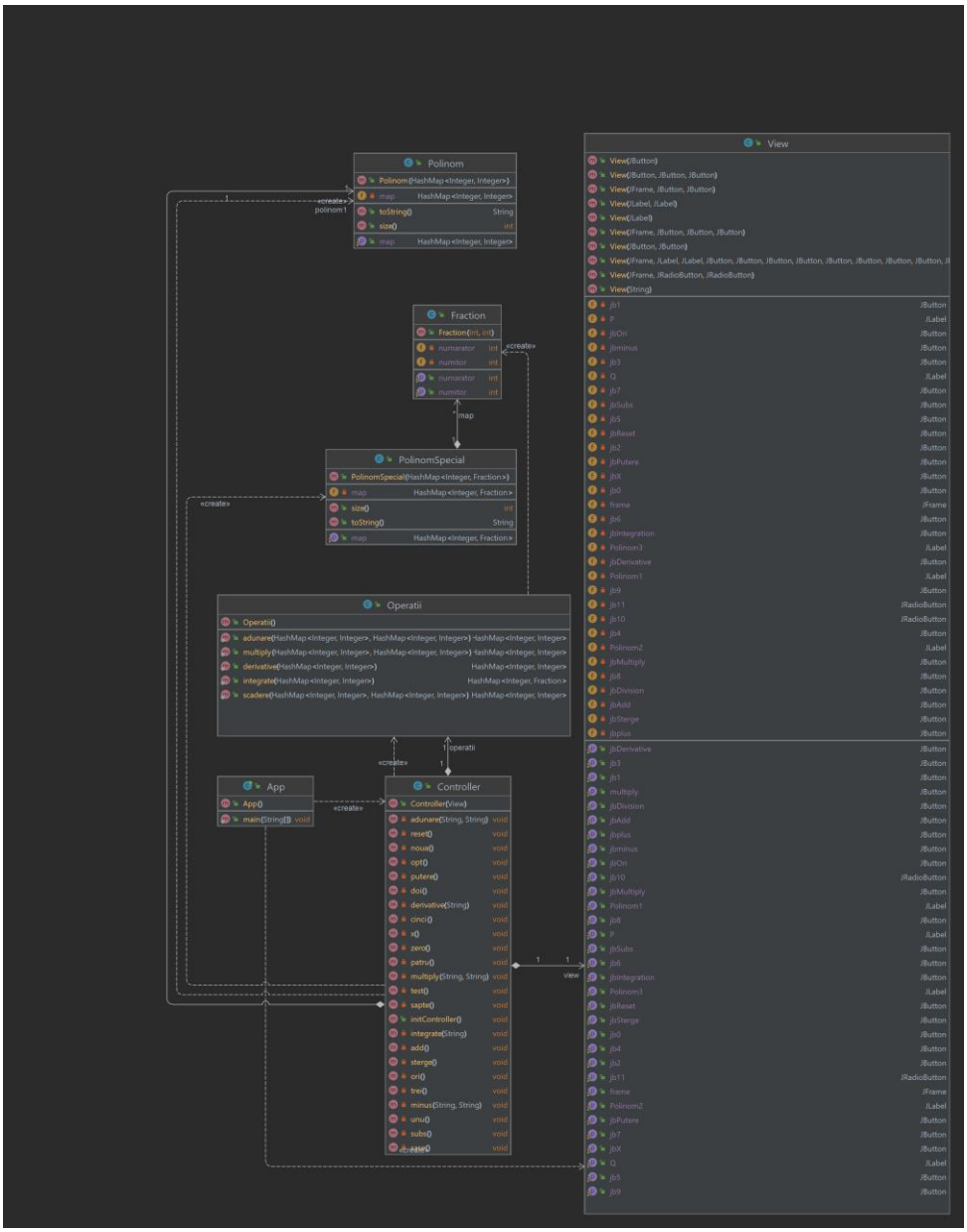
Scăderea, prin scăderea fiecarui coeficient al celui de-al doilea polinom la primul prin asocierea fiecărui grad al elementelor.

Înmulțirea se realizează prin parcurgerea primului polinom și extragerea fiecărui grad si coeficient, urmând ca prin parcurgerea celui de-al doilea polinom să se adune gradele intre ele, iar coeficienții să se înmulțească. Înainte de introducerea coeficienților se verifică dacă la gradul respectiv se află deja un coeficient in polinomul rezultat. Dacă se află aceștia se adună intre ei.

Derivarea unui polinom se realizează prin parcurgerea fiecarui element din polinom și tot odată introducerea în polinomul rezultat gradul nou prin scăderea cu unu si coeficientul specific prin înmulțirea coeficientului cu gradul.

Integrarea se realizează prin parcurgerea fiecărui element din polinom si tot odată introducerea în polinomul rezultat gradul nou prin adăugarea cu unu si coeficientul specific prin împărțirea coeficientului cu gradul.

Diagrama UML:



4.Implementare

Pentru realizarea acestui proiect s-au creat următoarele clase:

1. Polinom -> această clasă are scopul de a putea reprezenta cele două polinoame folosite în proiect. Reprezentarea se realizează cu ajutorul unei structuri de dată "HashMap", având doua argumente de tip Integer pentru stocarea puterii și coeficienților fiecărui element.

```
Map<Integer,String> map = Map.of(1, "A", 2, "B", 3, "C");
```

key	value
1	A
2	B
3	C

De asemenea sunt formate metode specifice "ToString" pentru a putea afișa polinom sub forma dorită și "size" pentru a putea determina dimensiunea structurii de date create.

2. Operații -> această clasă are scopul de a implementa metodele specifice calculatorului(adunarea, scăderea, înmulțirea, derivarea și integrarea). Aceste operații încep prin declararea unui nou HashMap, urmând ca în interiorul structurii să fie introduse elemenele in urma efectuării operațiilor, conform algoritmilor prezentați în capitolul "Proiectare".

3. Fraction -> pentru a putea reprezenta o fractie sub forma "a/b" a fost creată această clasă, având ca si proprietăți numărătorul si numitorul fracției.

4.PolinomSpecial -> de asemenea această clasă a fost creată pentru a putea reprezenta polinoamen ce au argumente sub forma de "a/b".

Implementarea interfeței grafice

Interactiunea utilizatorului cu calculatorului se realizează printr-o interfață grafică, creată cu ajutorul metodei MVC (Model View Controller). Această metodă presupune punerea împreună a 3 secțiuni pentru a putea reprezenta cu ușurintă această interfață.

1. View -> această clasă reprezintă ceea ce vede utilizatorul. Aici sunt prezentate fiecare componentă a interfeței și locul specific al acesteia.

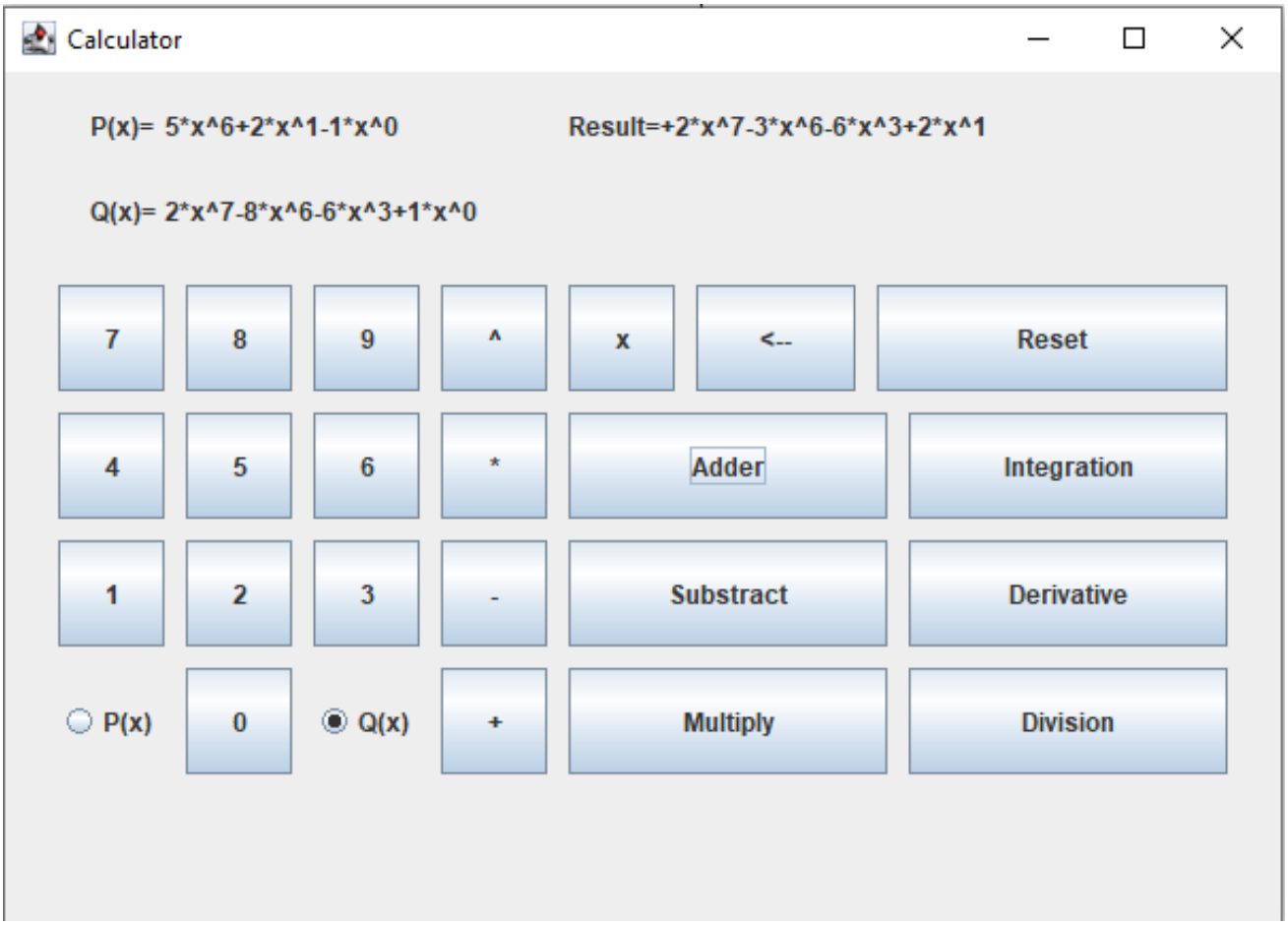
2. Controller -> această clasă reprezintă “creierul” aplicației. Are ca și proprietăți cele două polinoame pe care se realizează operațiile, View și Operații. În interiorul acestei clase se specifică funcționalitatea fiecărui buton. De asemenea pentru a se putea realiza operațiile declarate anterior, în metodele specifice butoanelor de operații, cu ajutorul funcției regex, se realizează spargerea Stringului într-un Array de numere intregi.

3. Model -> reprezintă clasele polinom si operații, fiind de fapt materialul pe care controller-ul îl manipulează.

4. App -> aceasta reprezinta main-ul aplicației unde se realizează inițializarea și instanțierea aplicației.

5.Rezultate

Adunarea a două polinoame:



6.Concluzii

Acest proiect a avut scopul aprofundării reprezentării interfeței grafice cu ajutorul metodei MVC. De asemenea utilizării structurilor de date și implementarea algoritmilor de operații a adus un beneficiu în dezvoltarea ulterioară a unor noi proiecte. Utilizarea funcției regex a determinat introducere ușoara a elementelor în structura de date HashMap.

Ca și posibile dezvoltări ulterioare, s-ar putea îmbunătății aspectul interfeței, și introducerea polinomelor sub forma "x^2+x^1+3".

7.Bibliografie

1 https://www.itcsolutions.eu/ro/2011/01/02/tutorial-java-6-6-clase-si-obiecte/

2. https://www.javatpoint.com/java-hashmap

3.https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm

4. https://github.com/

5. https://stackoverflow.com/questions/4736/learning-regular-expressions