

Calculator Polinoame

Documentație

Pentru: Tehnici de Programare

Studenți: Luncian Lidia, grupa 30227

Cuprins:

1. Obiectivul temei
2. Analiza problemei, modelare, scenarii, cazuri de utilizare
3. Proiectare
4. Implementare
5. Rezultate
6. Concluzii
7. Bibliografie

1. Obiectivul temei

Obiectivul acestei teme a fost proiectarea și implemetarea unui calculator de polinoame cu o interfață grafică ”User -Friendly”, cu ajutorul căreia utilizatorul poate introduce polinoame și poate selecta operația matematică (adunare, scădere, înmulțire, împărțire, derivare și integrare) care să fie aplicată polinoamelor.

Proiectarea va fi discutată la punctul 3 al documentației.

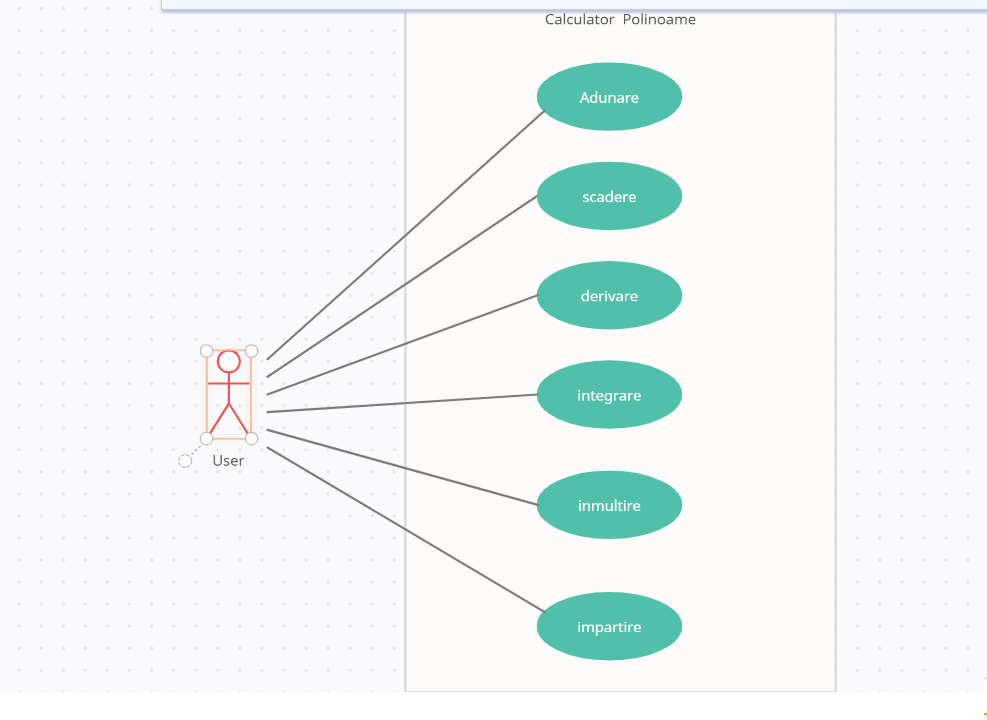
Implementarea va fi discutată la punctul 4 al documentației.

Interfața și operațiile matematice vor fi discutate atât la punctul 2 cât și la punctl 3 al acestei documentații.

2. Analiza problemei

Un polinom este implementat ca un ArrayList de monoame. Monomul este format dintr-un coeficient real și o putere întreagă și este de forma ax^b, a este coeficientul și b este puterea. Sub această formă trebuie și utilizatorul să introducă polinoamele pentru a avea rezultate corecte. Dacă coeficientul sau puterea sunt 1 sau 0 ele nu pot fi omise din scrierea polinomului. Regexul folosit pentru citirea si convertirea șirurilor, introduse de utilizator în Text Field-ul alocat fiecărui polinom, în polinoame, recunoaște doar șirurile de forma: (de exemplu) : ”ax^b+cx^d-...”. Dacă puterea sau coeficientul este omis, nu se mai citește corect. De asemenea dacă nu este introdus un text corespunzător, în urma apasării oricărui buton ce reprezintă o operație matematică, nu va apărea niciun rezultat, dar nici eroare. Singura avartizare care apare este dacă nu au fost introduse polinoamele.

Mai jos voi prezenta detaliat use-casul temei.



Pentru use-case-ul adunare, actorul este utilizatorul. Acesta introduce două polinoame folosind interfața grafică, apoi apasă butonul ”Adunare”. Dacă polinoamele au fost introduse corect, în Text Field-ul rezultat va apărea rezultatul corect al adunării celor două polinoame. Dacă niciunul din polinoame nu a fost introdus corect pe rezultat nu va apărea nimic. Dacă un polinom a fost introdus corect și celălalt nu, atunci cel introdus corect va apărea pe rezultat. Dacă nu a fost introdus nimic în Text Field-urile pentru polinoame, atunci pe rezultat va apărea mesajul ”Nu ati introdus polinoamele”.

Pentru use-case-ul scădere, actorul este utilizatorul. Acesta introduce două polinoame folosind interfața grafică, apoi apasă butonul ”Scadere”. Dacă polinoamele au fost introduse corect, în Text Field-ul rezultat va apărea rezultatul corect al scăderii celor două polinoame. Dacă niciunul din polinoame nu a fost introdus corect pe rezultat nu va apărea nimic. Dacă un polinom a fost introdus corect și celălalt nu, atunci cel introdus corect va apărea pe rezultat. Dacă nu a fost introdus nimic în Text Field-urile pentru polinoame, atunci pe rezultat va apărea mesajul ”Nu ati introdus polinoamele”.

Pentru use-case-ul înmulțire, actorul este utilizatorul. Acesta introduce două polinoame folosind interfața grafică, apoi apasă butonul ”Inmultire”. Dacă polinoamele au fost introduse corect, în Text Field-ul rezultat va apărea rezultatul corect al înmulțirii celor două polinoame. Dacă niciunul din polinoame nu a fost introdus corect pe rezultat nu va apărea nimic. Dacă un polinom a fost introdus corect și celălalt nu, atunci cel introdus corect va apărea pe rezultat. Dacă nu a fost introdus nimic în Text Field-urile pentru polinoame, atunci pe rezultat va apărea mesajul ”Nu ati introdus polinoamele”.

Pentru use-case-ul împărțire, actorul este utilizatorul. Acesta introduce două polinoame folosind interfața grafică, apoi apasă butonul ”Impartire”. Dacă polinoamele au fost introduse corect, în Text Field-ul rezultat va apărea rezultatul împărțirii celor două polinoame compus din cât și rest, dar în acest proiect nu este afișat întocmai corect. Primul polinom trebuie să aibă gradul primului monom mai mare decât gradul celui de a-l doilea polinom. Altfel împărțirrea nu va avea loc, și nici un mesaj de eroare nu va apărea, ci rezultatul va rămâne pur și simplu gol.

Pentru use-case-ul integrare, actorul este utilizatorul. Acesta intoduce un polinom fie în Text Field-ul primului polinom fie în celui de-al doilea. Dacă introduce câte un polinom în ambele Text Field-uri, primul polinom este integrat. Rezultatul corect al integrării apare în rezultat. Dacă nu au fost introduse polinoame apare mesajul ”Nu ati introdus polinoamele”.

La fel este și pentru derivare: actorul este tot utilizatorul. Acesta introduce cu ajutorul interfeței grafice unul sau două polinoame. Dacă introduce câte un polinom în ambele Text Field-uri, primul polinom este derivat. Rezultatul corect al derivării apare în rezultat. Dacă nu au fost introduse polinoame apare mesajul ”Nu ati introdus polinoamele”.

Pentru a face altă operație pe aceleași polinoame utilizatorul nu trebuie să le șteargi și să le reintroducă, ci apasă pe butonul a cărei opeație dorește rezultatu.

3. Proiectare

Am proiectat această temă folosind arhitectura MVC, clasa Model este reprezentată de clasa Operatii, clasa View de clasa View și clasa Controller de clasa Controller. În main sunt instanțiate cele trei clase componente. Tot în main clasa View este făcută vizibilă. Clasa Operatii implementeaza operațiile matematice care vor avea loc asupra polinoamelor.

În clasa View este construită interfața cu utilizatorul, sunt puse butoanele, etichetele, câmpurile de text și culoarea background-ului. Tot aici sunt create și ActionListener-ele pentru butoane. ActionListener-ul pentru fiecare buton este implementat în clasa Controller.

Clasa Monom definește atributele unui monom, getter-ele, setter-ele acestora preucm și metode de adunare, scădere, înmulțire și îmărțire a două monoame. Acestea vor fi folosite în implementarea metodelor pe polinoame.

Clasa Polynom definește atributele și metoda de construire a acestora.

Diagrama UML:

Diagram

Description automatically generated

Interfața grafică:

Graphical user interface

Description automatically generated with medium confidence

4. Implementare

Clasa Monom:





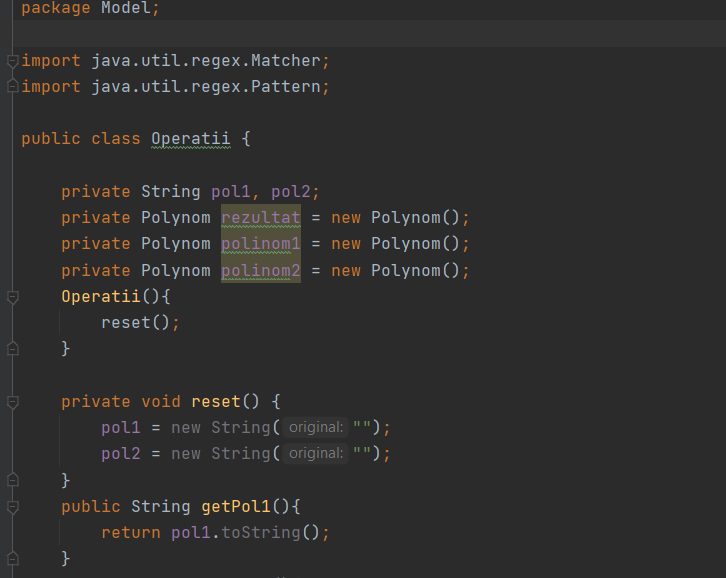
Această clasă are 2 atribute sugestiv denumite: un coeficien real și o putere întreagă. Am creat metode de get și set pentru fiecare atribut și un constructor pentru clasă care la crearea unui nou monom trebuie introduse coeficientul și puterea acestuia. Metoda toString este suprascrisă și afișează monomul în funcție de coeficient. Metodele adunareMonoame, scadereMonoame, inmultiremonoame, și impartireMonoame sunt implementeaza operații de adunare, scadere, inmultire si impartire pe monoame de aceeasi putere. Aceste metode au fost implementate pentru a usura implementarea metodelor pentru polinoame. In acest proiect am folosit dor metodele de inmultireMonoame si impartireMonoame in metoda de impartire din clasa Operatii. Celelalte metode pot fi folosite in metodele de adunare si scadere din clasa Operații.

Clasa Polynom:



În aceată clasă este definit polinomul ca un ArrayList de Monoame. Este implementat constructorul care creează o astfel de listă pentru fiecare polinom. Se adaugă în această listă monoame folosind metoda adaugaInLista. Metoda toString suprascrisă afișează un polinom.

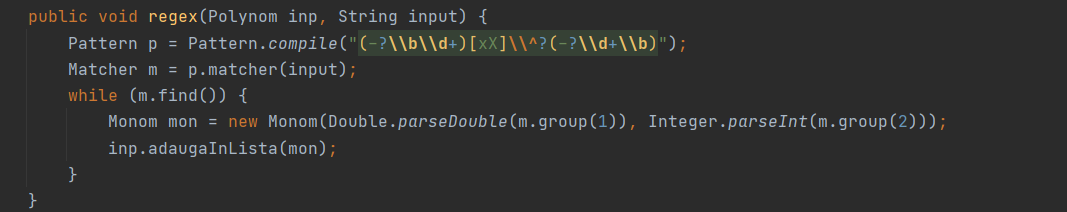
Clasa Operatii:



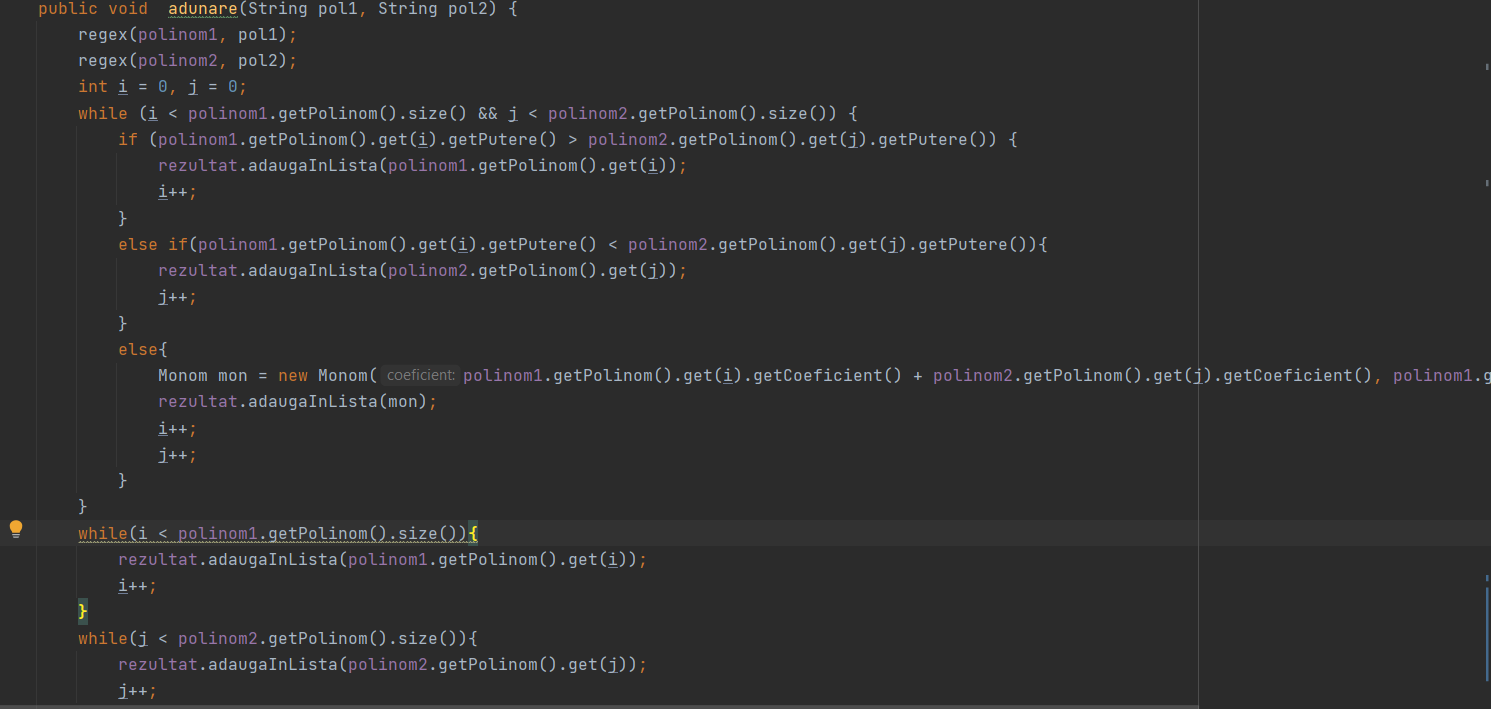
Clasa Operatii declară 2 String-uri ce reprezintă polinoamele ce vor fi date de utilizator prin intefața grafică ( pol1 și pol2). Acestea sunt convertite cu ajutorul regexului in polinoame de tipul Polynom (polinom1 și polinom2). Rezultatul va fi pus în polinomul de tip Polynom rezultat.



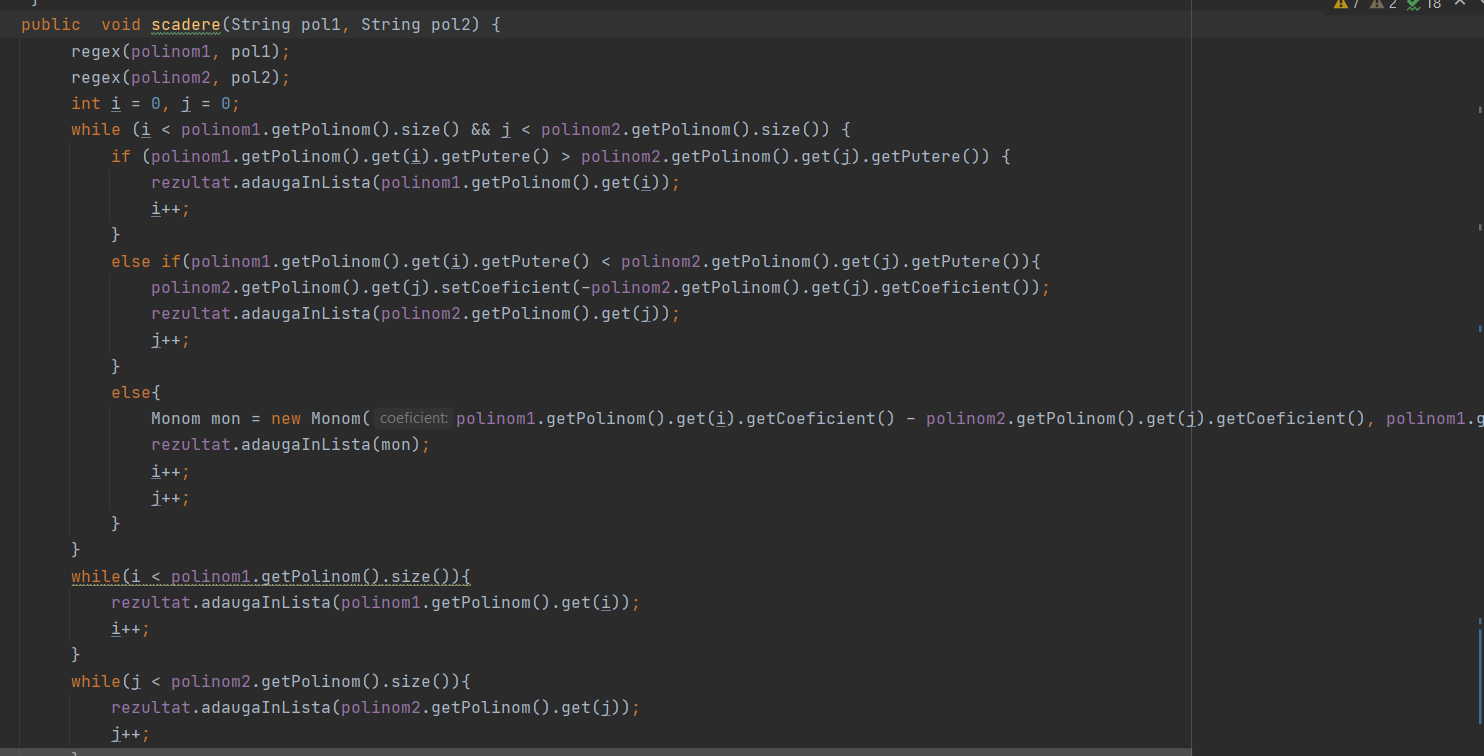
Metoda getRezultat este folosita in clasa Controller pentru a afisa rezultatul utilizatorului. Aceasta transmite rezultatul ca sir de caractere. Metoda setRezultat goleste toate cele trei polinoame, polinom1, polinom2, și rezultat pentru ca utilizatorul să facă altă operație folosind aceleași polinoame, fără ca rezultatul să se suprascrie sau să fie pus in continuarea celui vechi. Aceasta metodă este folosită tot în clasa Controller, în fiecare clasă implementată pentru ActionListener-ul fiecărui buton, pentru a reseta campurile de text a polinoamelor si a rezultatului.



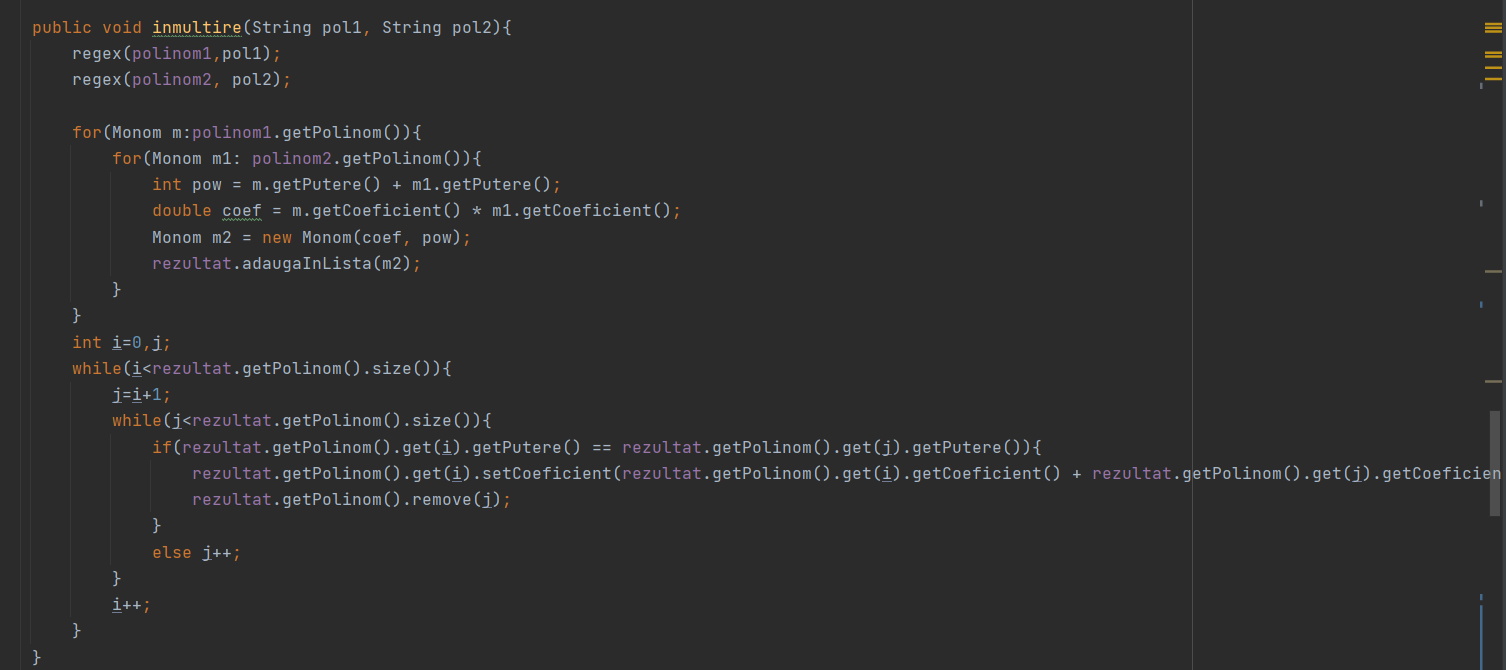
Metoda regex este folosită pentru a recunoaște un anumit pattern a intrărilor și doar cele corecte să ofere un rezultat. Această metodă caută un tipar de forma ”semn coeficient x^ putere intreaga” in sirul introdus ca parametru input. ( Acest input reprezintă inputul oferit de utilizator in Text Field-ul pentr polinomul 1 sau 2 din interfață  
). Daca este gasit un astfel de tipar, elementul din grupul 1 este parsat la un Double si este alocat coeficientului unui monom, iar elementul din grupul 2 este parsat la un Integer si este alocat puterii aceluiasi monom. Apoi acesta este adaugat polinomului inp. Dacă nu este gasit polinomul va fi gol și deci nu va avea ce să afișeze la rezultat orice buton am apăsa.



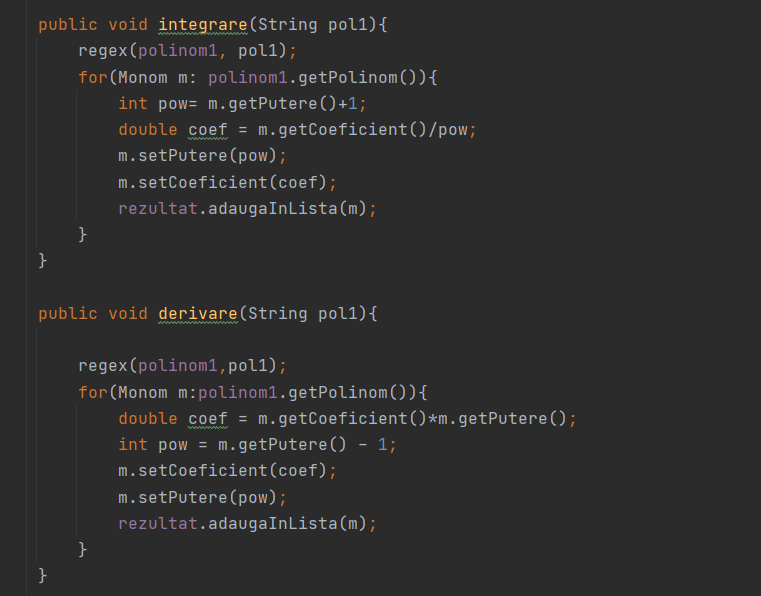
Metoda de adunare creeaza cele 2 polinoame folosind regex. Apoi cat timp i sau j este mai mic decat dimensiunea polinoamelor pe care le parcurg, in polinomul rezultat se adauga cate un monom in functie de coeficient si putere. Monoamele care au aceeasi putere li se adună coeficientii si puterea se pastreaza apoi e adaugat in rezultat. Altfel se adauga fara sa se modifice coeficientul. La finalul acestui while se verifica dacă s-a ajuns la finalul fiecărui polinom. Dacă indicele i este mai mic decât dimensiunea polinomului1, atunci acesta continua să crească și să adauge monoamele polinomului 1 în polinomul rezultat. La fel și pentru polinomul2.



Aceeași strategie se folosește și pentru scăderea a două polinoame. Singura diferentă este că atunci când două monoame au aceeași putere, se scad coeficienții (coeficientul monomului polinomului1 – coeficientul monomului polinomului2) si apoi se adauga monomul în rezultat.

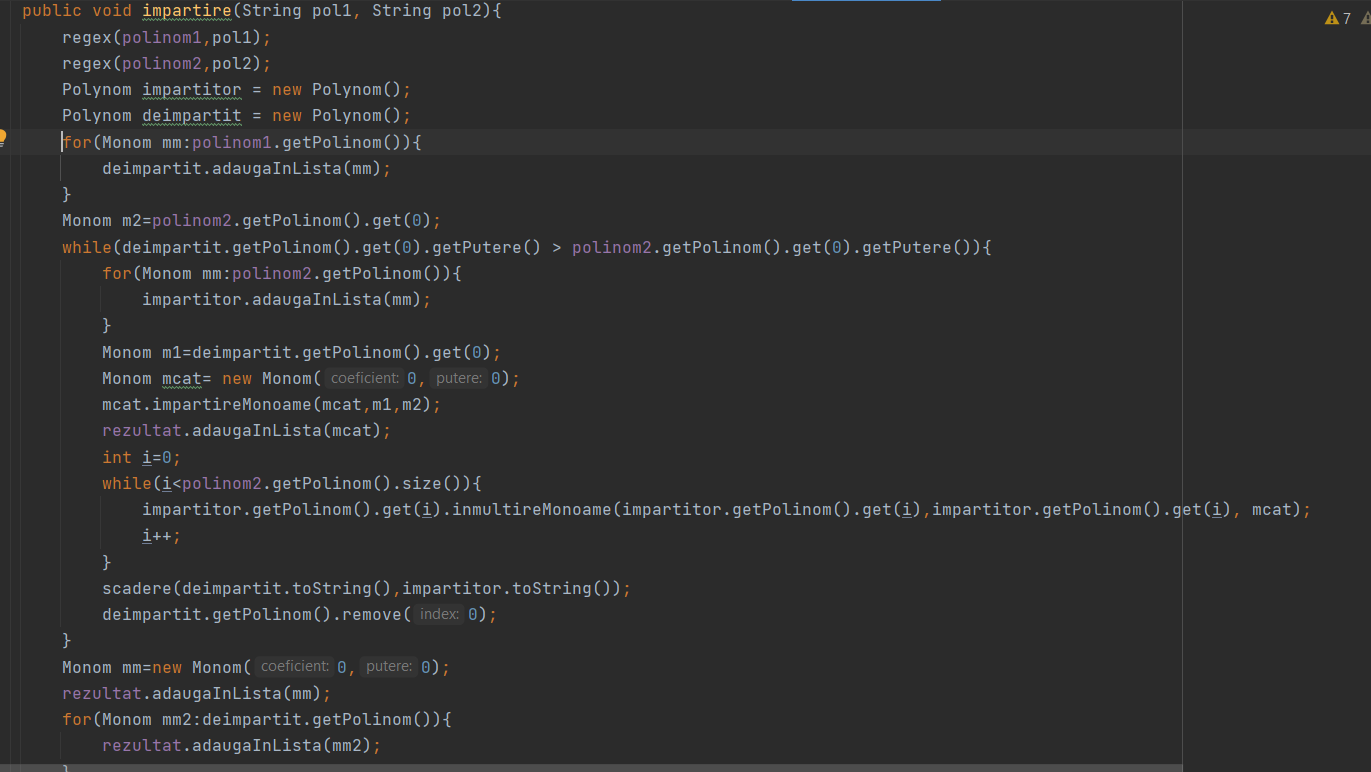


Metoda de înmulțire folosește următoarea strategie: se înmulțește fiecare monom al polinomului 1 cu fiecare monom al polinomului2. Coeficientii celor două monoame se înmulțesc, iar puterile celor două monoame se adună. Apoi se parcurge polinomul rezultat pentru a aduna monomaele care au ajuns să aibă aceeași putere și să nu apară de mai multe ori în rezultat aceeași putere. Realizăm acest lucru parcurgând polinomul rezultat cu un while în while. Când găsește două monoame cu aceeași putere, se adaugă coeficientul celui de-al doilea monom la coeficientul primului și cel de-al doilea monom este eliminat din rezultat.



Metoda de integrare parcurge polinomul 1 monom cu monom folosind un foreach. Se calculează noua putere adunând 1 la cea veche, și noul coeficient impărtind coeficientul la noua putere. Se actualizează monomul și apoi se adauga in polinomul rezultat.

Metoda de derivare presupune parcurgerea polinomului 1 monom cu monom folosind un foreach. Se calculeaza noul coeficient al monomului înmulțind vechiul coeficient cu puterea. Apoi puterea se scade cu 1. Se reactualizează monomul polinomului 1 și se adaugă în polinomul rezultat.



Metoda de împărțire creează 2 polinoame denumite impărțitor și deîmpărțit, în ccare se adaugă cele două polinoame de intrare. Cat timp gradul deîmpărțitului e mai mare decât puterea primului monom al celui de-al doilea polinom introdus de utilizator se foloseste metoda de impartireMonoame pentru a determina fiecare termen al rezultatului. Acest cat este adaugat in rezultat. Cu el se inmultește fiecare monom al polinomului 2. Apoi din deîmpărtit se scade polinomul 2 si se sterge monomul de pe indexul 0 care va avea ca coeficient pe 0.

Dupa ce gradul deimpartitului e mai mic decat al polinomului 2 restul impartirii se adauga in rezultat dupa un monom 0 0.

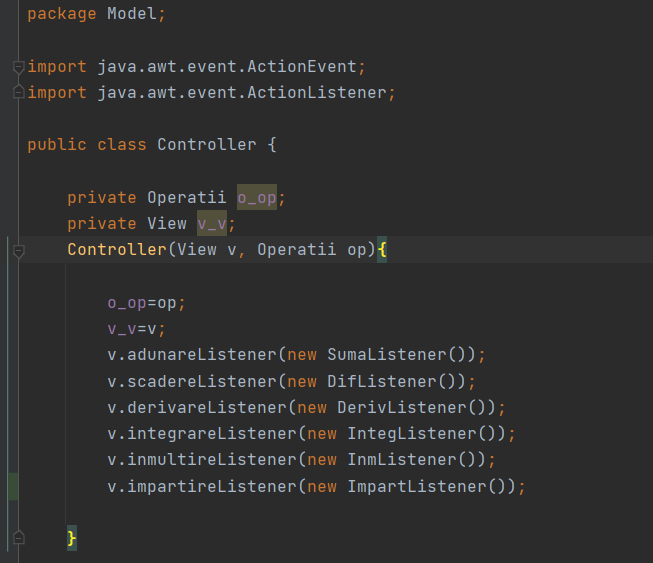
Clasa View:





În clasa View se creează interfața grafică cu utilizatorul. Interfata e alcatuita din 4 panel -uri. Panel 1 pentru polinomul1, panel 2 pentru polinomul 2, panel 3 pentru afisarea rezultatului si panel 4 pentru butoane.

Clasa Controller:



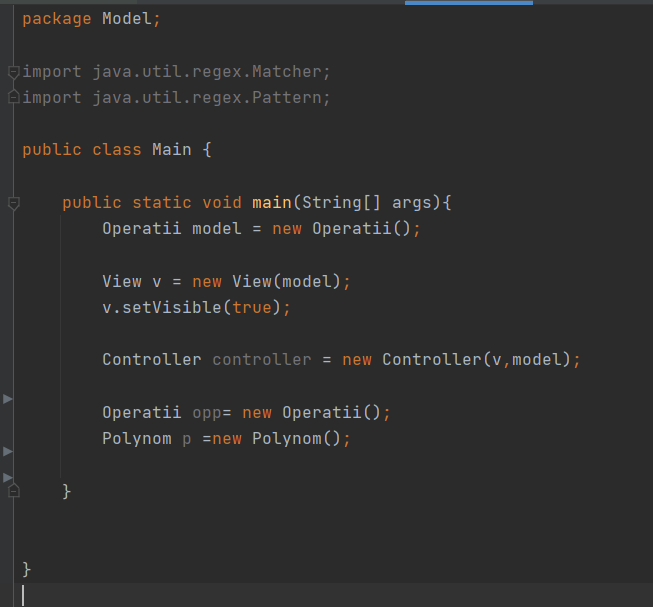






Clasa Controller implementeaza ActionListener pentru fiecare buton care declanșează o operație matematică. In fiacare clasa se declară doua String-uri in care se citesc intrarile interfetei, daca nu sunt nule se realizează operația, iar rezultatul este afisat utilizatorului.

Clasa Main:



În clasa main sunt instanțiate clasele arhitecturii MVC, iar clasa View este pusă pe modul vizibil.

Toate clasele fac parte din pachetul numit Model.

5. Rezultate

Am incercat să fac testarea folosind Junit. Ceva nu merge, nu pot testa metodele ce implementeaza operatiile. Metodele au fost testate manual pe mai multe exemple si conform calculelor mele, acestea sunt corecte. Chiar si impartirea face corect, dar nu afiseaza rezultatul bine.

Au fost introduse exemple înainte de folosirea regex-ului pentru confirmarea corectitudinii codului. Apoi am testat introducand in interfata grafica cat mai multe exemple pentru a observa dacă regexul converteste corect polinoamele. Acesta e corect, daca au fost introduse doar monoame valide. Altfel acesta nu functionează pentru operații.

6. Concluzii

În urma acestui proiect am reușit să înțeleg mai bine modelul MVC, deși chiar și acest proiect poate fi îmbunătățit și simplificat mult. Am aflat de regex și Junit. Am înțeles cum și la ce se folosesc acestea. Acest proiect se poate îmbunătăți, se poate modifica regex-ul ca utilizatorul să nu trbuiască sa mai intoducă coeficienții 0 sau 1. Pot fi puse mai multe mesaje de eroare. Desi ”tacerea” (lipsa mesajului) e în sine un mesaj, nu e tocmai cel mai potrivit in toate situatiile. :)). Am inteles la ce foloseste Junit, chiar daca in proiect nu mi-a ieșit. Operatia de împărțire poate fi îmbunătățita astfel incat să afiseze corect rezultatul, nu doar partial.

7. Bibliografie

<https://code-it.ro/calculator-de-polinoame-in-java/> , Mar 14, 2021

<https://stackoverflow.com/questions/49788045/polynomial-calculator-using-linked-list>, Mar 14, 2021

<https://stackoverflow.com/questions/36490757/regex-for-polynomial-expression>, Mar 14, 2021

<https://www.youtube.com/watch?v=-sTWutdffY4&ab_channel=MarkLewis> , Mar 14, 2021

<https://stackoverflow.com/questions/32844417/java-parsing-a-polynomial-with-complex-coefficients-with-regex>, Mar 14, 2021

<https://crunchify.com/what-is-regex-pattern-regular-expression-how-to-use-it-in-java-example-attached/>, Mar 14, 2021

<https://topic.alibabacloud.com/a/use-a-regular-expression-font-colorredregexfont-to-match-polynomials-polynomial-regexpolynomial_1_31_32668196.html>, Mar 14, 2021

<https://social.msdn.microsoft.com/Forums/en-US/f17d9919-52d3-4ebc-b67e-44f6e6e73501/regex-for-polynomial?forum=regexp>, Mar 14, 2021

<https://stackoverflow.com/questions/28859919/java-regex-separate-degree-coeff-of-polynomial>, Mar 14, 2021

<https://www.javatpoint.com/java-regex>, Mar 14, 2021

<https://www.wyzant.com/resources/answers/627688/how-do-i-pass-in-a-polynomial-function-in-java>, Mar 16, 2021

<https://app.creately.com/diagram/p9hYhUegR2r/edit>, Mar 17, 2021