DOCUMENTATIE

TEMA 1

NUME STUDENT: VLASE ELENA-RAFAELLA

GRUPA: 30225

# CUPRINS

[1. Obiectivul temei 3](#_Toc95297885)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](#_Toc95297886)

[3. Proiectare 4](#_Toc95297887)

[4. Implementare 6](#_Toc95297888)

[5. Rezultate 8](#_Toc95297889)

[6. Concluzii 10](#_Toc95297890)

[7. Bibliografie 11](#_Toc95297891)

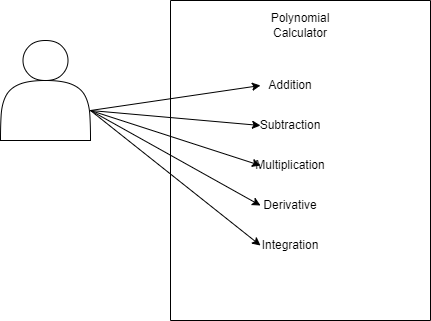
# Obiectivul temei

Obiectivul primei teme este implementarea unui calculator polinomial cu următoarele operații: adunare, scădere, înmulțire, împărțire a două polinoame, derivare și integrare a unui polinom.

Pentru a atinge acest obiectiv a fost nevoie de implementarea următoarelor:

* Interfață grafică (GUI)
* Structura de date Polinom folosind TreeMap
* Metode de convertire din String in Polinom, folosind regex
* Operațiile de adunare, scădere, înmulțire, derivare și integrare
* Funcții auxiliare precum toString etc.

# Analiza problemei, modelare, scenarii, cazuri de utilizare



Analiza:

Polinomul este o expresie de forma



Accentul se pune pe implementarea corectă a operațiilor pe polinoame. Am plecat de la premisa că gradele monoamelor din polinoame sunt valori întregi, iar coeficienții pot avea și valori reale. După efectuarea oricărei operații, termenii din polinom ce au coeficientul 0 nu vor fi afișați.

Modelare:

Structura de date “Polynomial” conține un TreeMap pentru polinoam, “polynomial, key este gradul și value este coeficientul.

Scenarii:

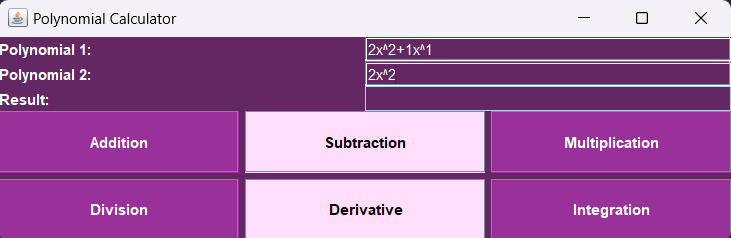
Pentru funcționare:

- Necunoscuta se notează cu “x”

- Termenii sunt de forma ax^b

- Nu se pun spații

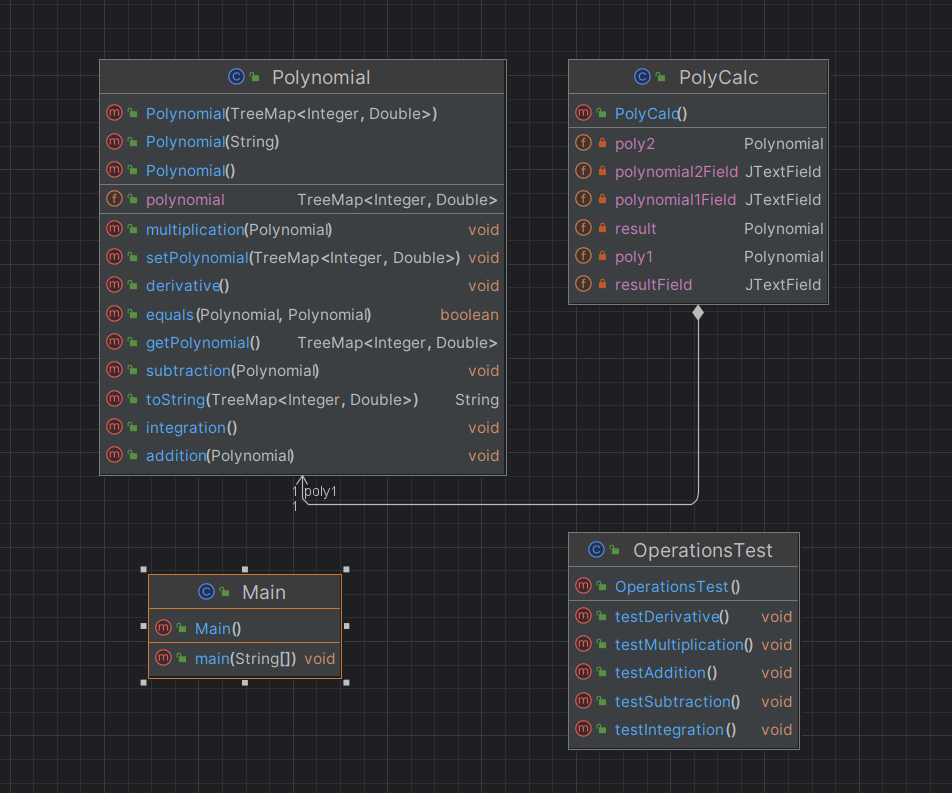
Use cases:



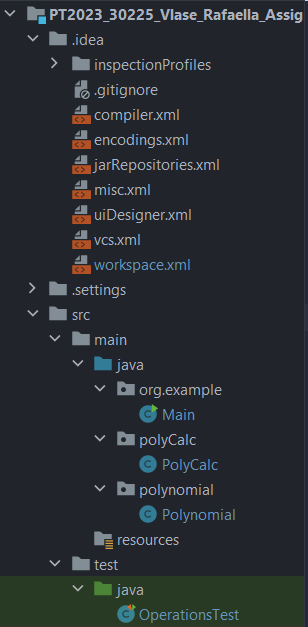
Interfața grafică este formată din două câmpuri de text pentru a insera polinoamele, un câmp unde va fi afișat rezultatul și șase butoane corespunzătoare fiecărei operații: adunare, scădere, înmulțire, împărțire, derivare, integrare. La apăsarea fiecărui buton va apărea în câmpul “Result” rezultatul operației alese prin apăsarea unui buton.

# Proiectare

Diagrama UML:



Ierarhia de fișiere a proiectului:



PolyCalc este clasa în care am implementat interfața grafică, iar în Main se apelează constructorul acesteia pentru instanțierea interfeței grafice.

# Implementare

Clasa Polynomial este folosită pentru a realiza operații pe polinoame. Aceasta are următorul atribut:

polynomial: un obiect TreeMap<Integer, Double> ce reține coeficienții polinomului, împreună cu gradele monoamelor din polinom.

Metodele clasei sunt urmatoarele:

getPolynomial(): returnează obiectul TreeMap<Integer, Double> al coeficienților polinomului.

setPolynomial(TreeMap<Integer, Double> polynomial): setter pentru obiectul TreeMap<Integer, Double> al coeficienților polinomului.

Polynomial(): constructor implicit care inițializează obiectul polynomial cu un TreeMap gol.

Polynomial(TreeMap<Integer, Double> polynomial): constructor care inițializeaza obiectul polynomial cu un TreeMap dat ca parametru.

Polynomial(String polynomialString): constructor care inițializează obiectul polynomial pe baza unei expresii de tipul String ce conține polinomul.

addition(Polynomial p): adună la polinomul curent un alt polinom dat ca parametru.

subtraction(Polynomial p): scade din polinomul curent un alt polinom dat ca parametru.

multiplication(Polynomial p): înmulțește polinomul curent cu alt polinom dat ca parametru.

derivative(): calculează derivata polinomului curent.

integration(): calculează integrala polinomului curent.

toString(TreeMap<Integer, Double> polynomial): transformă polinomul reprezentat de un TreeMap în formatul String corespunzător.

equals(Polynomial p1, Polynomial p2): verifică dacă două polinoame date ca parametri sunt egale.

Clasa PolyCalc reprezintă un calculator de polinoame cu o interfață grafică utilizator (GUI). Clasa conține câteva câmpuri și metode care permit utilizatorilor să introducă două polinoame și să efectueze diferite operații, precum adunarea, scăderea, înmulțirea, derivarea și integrarea lor.

Clasa PolyCalc extinde clasa JFrame și are următoarele câmpuri de date:

polynomial1Field: câmpul text pentru primul polinom

polynomial2Field: câmpul text pentru al doilea polinom

resultField: câmpul text pentru rezultat

poly1: obiectul Polynomial asociat primului polinom

poly2: obiectul Polynomial asociat al doilea polinom

result: obiectul Polynomial asociat rezultatului

Metoda constructor a clasei inițializează toate câmpurile de date și creează o interfață grafică cu două panouri: unul în partea de sus pentru introducerea polinoamelor și afișarea rezultatului, și unul în partea de jos cu butoane pentru efectuarea diferitelor operații cu polinoamele.

Clasa PolyCalc definește, de asemenea, o serie de metode private care sunt apelate când utilizatorul apasă pe un buton corespunzător. Aceste metode creează obiecte Polynomial pe baza inputului utilizatorului și apelează metodele corespunzătoare ale obiectelor pentru a efectua operațiile aritmetice necesare. Rezultatul este apoi afișat în resultField.

Metodele clasei includ:

"PolyCalc()": Constructorul clasei care inițializează interfața grafică și câmpurile necesare pentru a efectua operațiile.

"addButton.addActionListener()": O metodă care adaugă un ascultător la butonul de adunare. Când butonul este apăsat, această metodă crează două instanțe ale clasei Polynomial, le adaugă împreună și afișează rezultatul în "resultField".

"subButton.addActionListener()": O metodă care adaugă un ascultător la butonul de scădere. Când butonul este apăsat, această metodă crează două instanțe ale clasei Polynomial, le scade și afișează rezultatul în "resultField".

"mulButton.addActionListener()": O metodă care adaugă un ascultător la butonul de înmulțire. Când butonul este apăsat, această metodă crează două instanțe ale clasei Polynomial, le înmulțește și afișează rezultatul în "resultField".

"derButton.addActionListener()": O metodă care adaugă un ascultător la butonul de derivare. Când butonul este apăsat, această metodă creează o instanță a clasei Polynomial și calculează derivata polinomului, apoi afișează rezultatul în "resultField".

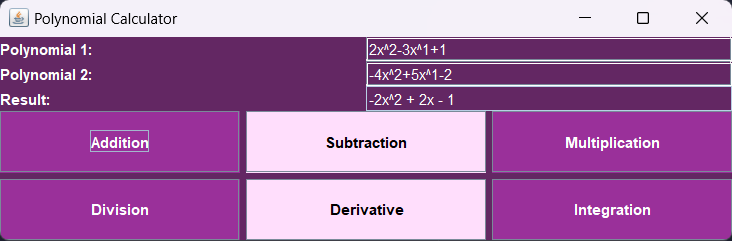
"intButton.addActionListener()": O metodă care adaugă un ascultător la butonul de integrare. Când butonul este apăsat, această metodă creează o instanță a clasei Polynomial și calculează integrala polinomului, apoi afișează rezultatul în "resultField".

Clasa Main reprezintă punctul de intrare (entry point) al programului. Ea conține metoda "main", care instantiază clasa "PolyCalc", declanșând astfel deschiderea aplicației calculatorului de polinoame.

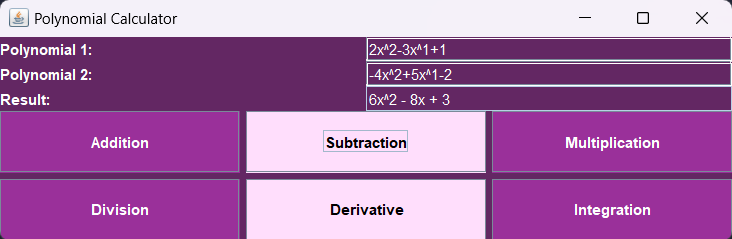
# Rezultate

Testarea a fost realizată în clasa OperationsTest, folosind JUnit, cu ajutorul metodei assertTrue() și a metodei auxiliare equals(), implementată în clasa Polynomial.

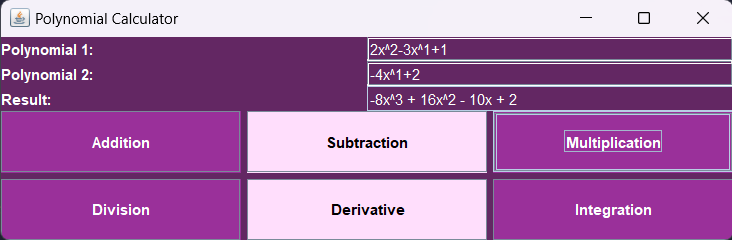
Adunare:



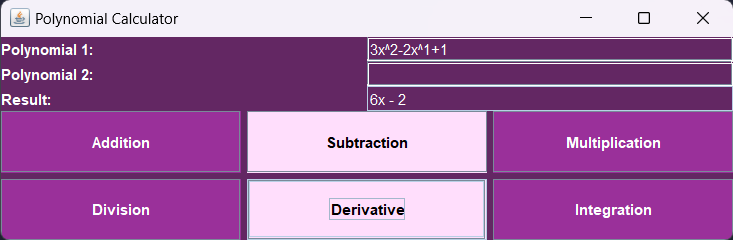
Scădere:



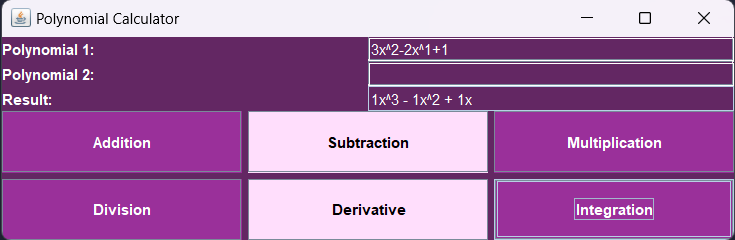
Înmulțire:



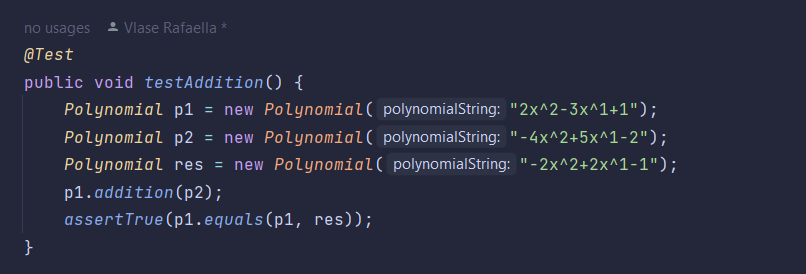
Derivare:



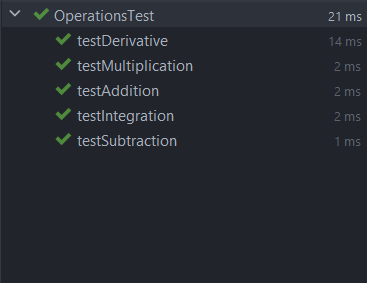
Integrare:



Exemplu de test în JUnit:



Rezultatele testării în JUnit:



# Concluzii

Cu ajutorul acestei teme am învățat să lucrez cu git și tot acum am folosit pentru prima data Maven.

O posibilă dezvoltare ulterioară a calculatorului de polinoame este implementarea metodei de împărțire sau implementarea unei funcționalități ce permite afișarea unui istoric al rezultatelor operațiilor efectuate anterior.

# Bibliografie

<https://www.geeksforgeeks.org/java-swing-jpanel-with-examples/>

<https://docs.oracle.com/javase/tutorial/uiswing/index.html>

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

<https://topic.alibabacloud.com/a/use-a-regular-expression-regex-to-match-polynomials-polynomial-regexpolynomial_1_31_32668196.html>

<https://dsrl.eu/courses/pt/>