DOCUMENTATIE

TEMA *1*

NUME STUDENT: Tîrcă Mihnea

GRUPA: 30224

# CUPRINS

[1. Obiectivul temei 3](#_Toc95297885)

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

[3. Proiectare 3](#_Toc95297887)

[4. Implementare 3](#_Toc95297888)

[5. Rezultate 3](#_Toc95297889)

[6. Concluzii 3](#_Toc95297890)

[7. Bibliografie 3](#_Toc95297891)

# Obiectivul temei

Obiectivul acestei teme este de a implementa un calculator de polinoame, capabil sa faca operatiile de adunare a doua polinoame, scadere a doua polinoame, inmultire a doua polinoame, impartire a doua polinoame, derivarea unui polinom si integrarea unui polinom.

Pentru a atinge acest obiectiv, a trebuit sa implementez:

● Interfata grafica

● Implementarea structurii de date de polinom folosind liste inlantuite

● Functie care converteste un String in structura de date de polinom, folosing biblioteca regex

● Operatiile mentionate anterior

● Diferite functii ajutatoare precum: impartirea a doua monoame, copierea unui polinom, formatare numar double, toString etc.

● Implementarea operatiilor in interfata grafica

# Analiza problemei, modelare, scenarii, cazuri de utilizare

Diagram

Description automatically generated

**Analiza:**

Un polinom reprezinta o expresie de forma:

Implementarea corecta operatiilor pe polinoame a fost cel mai important lucru. Am presupus ca valorile gradelor monoamelor sunt mereu valori intregi si ca utilizatorul introduce de la tastatura coeficienti intregi de asemenea, desi la integrare coeficientul rezultatului este afisat sub forma de numar real. Dupa efectuarea operatiei, termenii polinomului cu coeficientul 0 nu se afiseaza.

**Modelare:**

Structura de date “*Polynomial*” contine o lista inlantuita cu monoame.

Structura de date “*Monomial*” contine un *double* reprezentand coeficientul si un *int* reprezentand gradul.

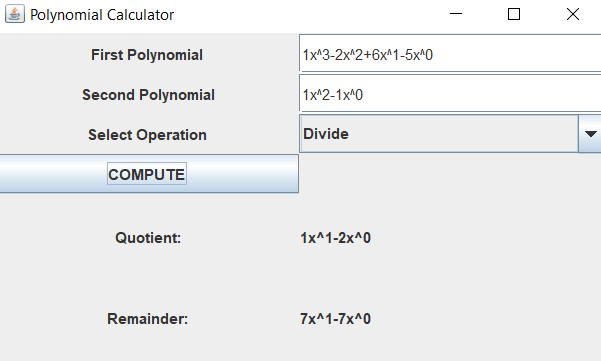
Pentru a opera mai usor cu structura de date de polinom, cand un String care reprezinta un polinom este introdus de la tastatura, programul il va ordona pe parcursul citirii polinomului, intr-o lista inlantuita, in sens crescator gradelor monoamelor. Acest lucru este de asemenea obligatoriu pentru a putea efectua operatia de impartire. La afisarea polinomului rezultat dupa operatie, acesta va fi afisat in ordine descrescatoare gradelor.

**Scenarii:**

Pentru corecta functionare a programului:

* Necunoscuta este ‘x’
* Fiecare termen va fi de forma: ax^b (in afara de termenii nuli, care nu trebuie scrisi)
* Nu se pune niciun spatiu

**Cazuri de utilizare**

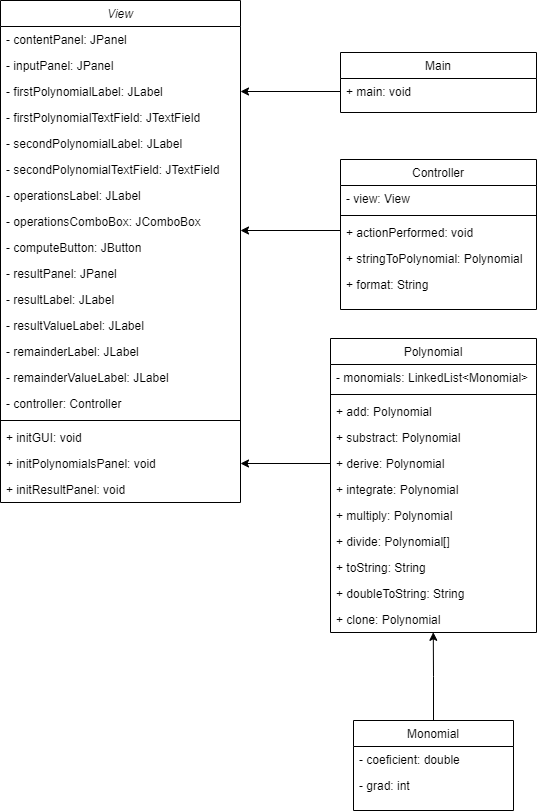


Interfata grafica contine doua campuri de text pentru inserarea polinoamelor, marcate corespunzator. Un drop-down menu permite alegerea operatiei de catre utilizator. Operatiile disponibile sunt: adunare, scadere, inmultire, impartire, derivare, integrare. Utilizatorul poate apoi apasa pe butonul “COMPUTE”, iar rezultatul va fi afisat pe jumatatea de jos a ferestrei.

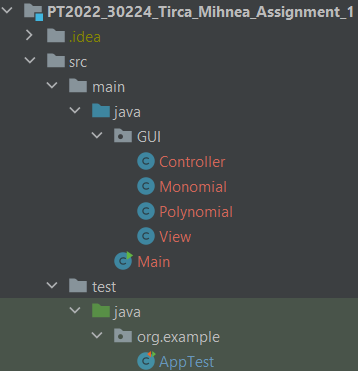
# Proiectare

Atributele claselor Controller, Monomial si Polynomial sunt declarate implicit package-private, iar atributele clasei View sunt declarate private. Metodele de operatii si alte metode sigure sunt declarate publice. Anumite metode care se folosesc doar in clasa respectiva si care nu au nevoie sa fie publice, sunt declarate private.

Diagrama:



Ierarhia de fisiere a proiectului:



Proiectul este se bazeaza pe structura Model View Controller:

Model: Monomial.java; Polynomial.java

View: View.java

Controller: Controller.java

Din main se apeleaza constructorul clasei View, care mosteneste clasa JFrame, astfel instantiindu-se interfata grafica.

# Implementare

**Clasa** *Monomial:*

Utilizata pentru stocarea unui monom.

**Campuri:**

double coeficient;  
int grad;

Un camp **double** care tine minte valoarea coeficientului si un camp **int** care tine minte valoarea gradului.

**Metode**:

public Monomial divide(Monomial)

Primeste ca parametru un monom si returneaza rezultatul impartirii monomului **this** la monomul primit prin parametru. Aceasta functie este folosita in functia de impartire a 2 polinoame.

**Clasa** *Polynomial*:

Utilizata pentru stocarea unui polinom.

**Campuri**:

LinkedList<Monomial> monomials;

O lista inlantuita cu monoame. Este ordonata in ordine crescatoare in functia stringToPolynomial(String);

**Metode:**

public Polynomial add(Polynomial)

Primeste ca parametru un polinom si returneaza suma polinomului **this** si polinomul primit ca parametru.

Deoarece polinomul este tinut minte in ordine crescatoare a gradului, pentru a aduna polinoamele, se interaclaseaza 2 siruri ordonate de grade a polinoamelor. Daca gradele sunt egale, se aduna coeficientii.

public Polynomial subtract(Polynomial)

Primeste ca parametru un polinom si returneaza diferenta intre polinomul **this** si polinomul primit ca parametru.

public Polynomial derive()

Returneaza derivata polinomului **this**.

public Polynomial integrate()

Returneaza integrata polinomului **this**.

public Polynomial multiply(Polynomial)

Primeste ca parametru un polinom si returneaza produsul intre polinomul **this** si polinomul primit ca parametru.

public Polynomial[] divide(Polynomial)

Primeste ca parametru un polinom si returneaza catul si restul (intr-un vector de polinoame) al impartirii polinomului **this** si polinomul primit ca parametru.

@Override  
public String toString()

Returneaza reprezentarea polinomului **this** in String, cu coeficientii cast in **int**.

public String doubleToString()

Returneaza reprezentarea polinomului **this** in String.

@Override  
public Polynomial clone()

Returneaza o copie a polinomului **this**.

**Clasa** *View*

Aceasta clasa mosteneste clasa JFrame

**Campuri:**

Campurile acestei clase reprezinta fiecare componenta din interfata grafica, adica cele 2 Label-uri de polinoame, Text Field-urile, Drop-Down Menu, Result Label si Label-ul de valoare rezultata.

**Metode:**

public void initGUI()

Aceasta metoda initializeaza interfata grafica. Ea apeleaza de asemenea metodele initPolynomialsPanel si initResultPanel.

public void initPolynomialsPanel()

Aceasta metoda initializeaza primul panel. Acesta reprezinta locul unde utilizatorul insereaza polinoamele, drop-down menu si butonul de calculat.

public void initResultPanel()

Aceasta metoda initializeaza al doilea panel. Acesta contine doar Label-urile de afisaj al rezultatului.

Clasa mai contine si metode de get, specifice limbajului OOP.

**Clasa** *Controller*

Aceasta clasa implementeaza interfata *ActionListener* si este de aceea folosita pentru controlul comportamentului aplicatiei in functie de input-urile utilizatorului.

**Campuri**:

private View view;

Campul de tip **View** este folosit pentru legarea clasei Controller cu clasa View, care este responsabila cu interfata grafica.

**Metode**:

public static String format(double)

Aceasta metoda este folosita pentru a formata numerele **double** cand afisez rezultatul unei integrari. Formatarea este de fapt scrierea numerelor **double** cu doar 2 cifre dupa punctul zecimal.

public static Polynomial stringToPolynomial(String)

Aceasta metoda foloseste libraria java.util.regex pentru a citi polinomul din tip String in structura de date creata de mine. Aceasta functie permite scrierea polinomului doar sub forma **ax^b**. Coeficientul **a** trebuie scris chiar daca are valoarea 1, iar gradul **b** trebuie scris chiar daca are valoarea 1 sau 0. Necunoscuta poate fi doar **x**.

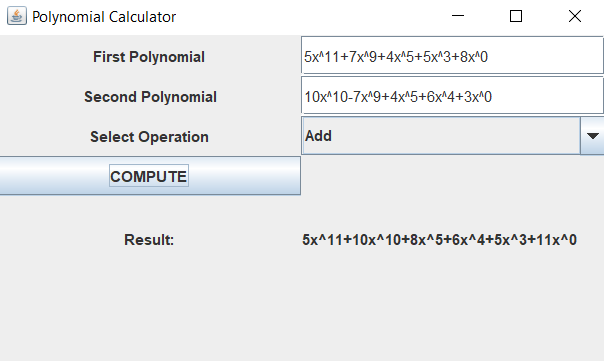
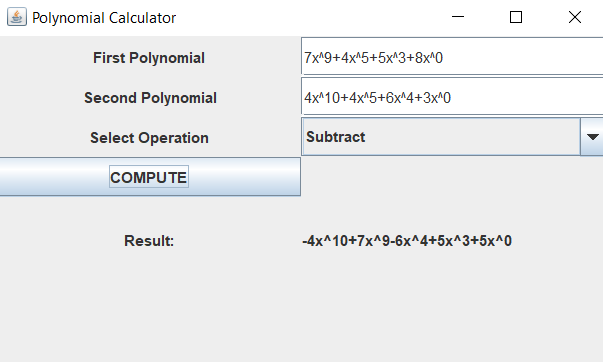
public void actionPerformed(ActionEvent)

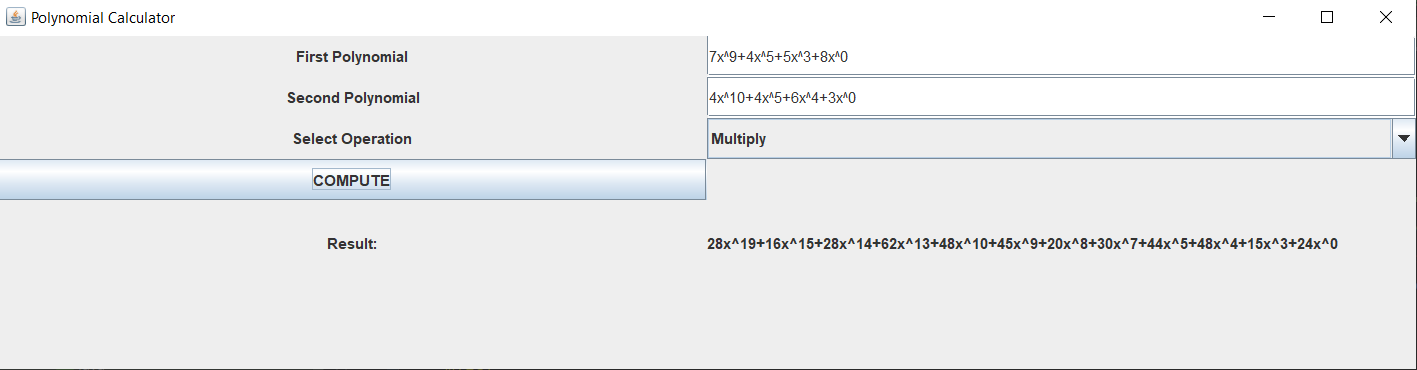
Aceasta metoda este folosita pentru a computa operatiile alese de utilizator, la apasarea butonului *COMPUTE*.

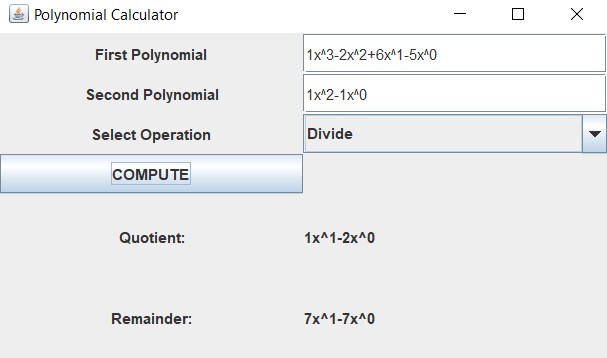
# Rezultate

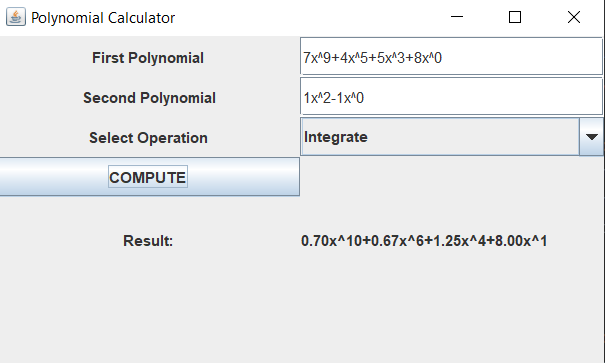
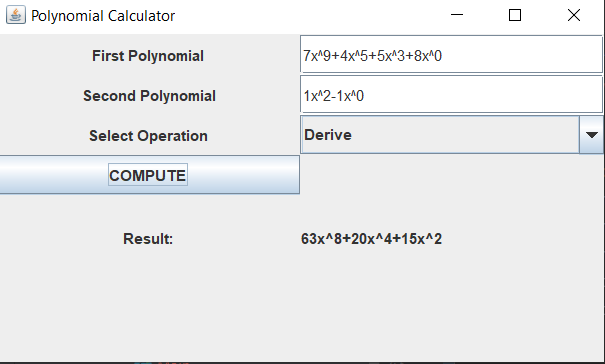
*Se vor prezenta scenariile pentru testare cu Junit sau alt framework de testare.*

Testarea a fost realizata in Junit, cu ajutorul metodei Assertions.AssertEquals()

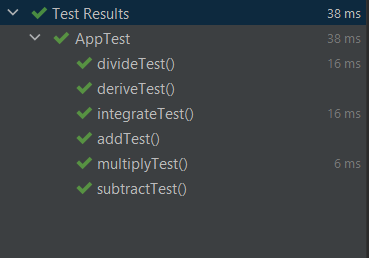
**

**





Rezultatele testarii:



# Concluzii *.*

Aceasta tema m-a ajutat sa imi aprofundez cunostintele in Swing. De asemenea, m-a invatat sa lucrez cu git, fiind foarte important in locurile de munca. Am lucrat de asemenea pentru prima data cu Maven si JUnit si am invatat sa imi fac propriile teste.

# Bibliografie

<https://docs.oracle.com/javase/7/docs/>

<https://geeksforgeeks.org/>