



Calculator de Polinoame Documentație

Nume: Fărcaș Tudor Ioan
Grupa: 30225
AN: 2

Cuprins

1. Obiectiv
2. Studiul Problemei
3. Implementare
4. Concluzii
5. Bibliografie

1.Obiectiv

(i) Obiectivul principal al temei este de a crea un calculator polinomial care să ofere funcționalități precum adunarea, scăderea, înmulțirea, derivarea și integrarea polinoamelor.

(ii) Obiectivele secundare sunt următoarele:

Obiectiv secundar	Descriere
Implementarea clasei Polinom	Definirea și implementarea unei clase care să reprezinte un polinom.
Adunarea a doua polinoame	Definirea și implementarea funcției de adunare a două polinoame.
Scaderea a doua polinoame	Definirea și implementarea funcției de scădere a doua polinoame.
Înmulțirea a două polinoame	Definirea și implementarea funcției de înmulțire a doua polinoame.
Derivarea unui polinom	Definirea și implementarea funcției de derivare a unui polinom.
Integrarea unui polinom	Definirea și implementarea funcției de integrare a unui polinom.
Interfață grafică	Implementarea unei interfețe grafice pentru a oferi o modalitate mai intuitivă de utilizare a calculatorului polinomial.

2. Studiul Problemei

Un polinom este o expresie realizată din mai multe monoame (componente care au coeficienți și exponențiali). Din punct de vedere al graficii sau al codului, se poate realiza un calculator polinomial în foarte multe metode.

Pentru a prelua informațiile din interfață și pentru a le transforma în obiecte Java ce pot fi calculate, vom folosi metoda `Regex` care preia din `TextFields` String-ul de polinoame și îl face sub formă de `Polinom`.

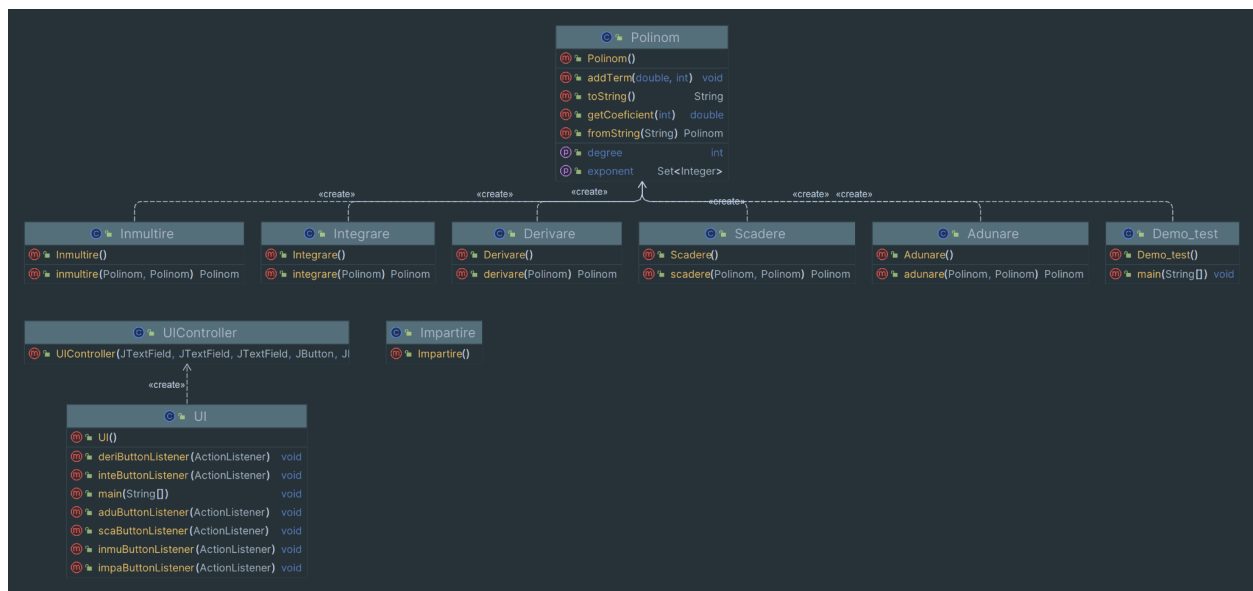
Metoda utilizată este una simplă, însă tind să spun că nu e foarte optimizată și pot interveni anumite erori.

3. Implementare

3.1 Diagrama UML

UML, sau Limbajul de Modelare Unificat, este o limbaj grafic care permite dezvoltatorilor de software să vizualizeze, să specifice, să construiască și să documenteze sisteme complexe. Este larg utilizat în industria software pentru modelarea sistemelor de software.

Diagramelor UML sunt utilizate pentru a reprezenta diferite aspecte ale unui sistem software. Acestea sunt compuse din elemente grafice, cum ar fi cutii, săgeți, linii și simboluri, care sunt utilizate pentru a reprezenta entități și relații între entități.



În poza de mai jos este realizata interacțiunea User-Interface cu sistemul realizat.

Pachetul Funcții

Este pachetul în care s-au implementat clasele de funcții ce le putem realiza în acest program.

Pachetul Interfață

Este pachetul în care s-a realizat interfața grafică destinată utilizatorului.

Pachetul Controller

Este pachetul în care s-a realizat legătura între butoanele din GUI și funcțiile programului.

Pachetul Polinom

Este pachetul în care s-a realizat clasa Polinom, în care avem funcțiile necesare pentru a implementa operațiile.

3.2 Metode utilizate în Pachetul Functii

— **Adunare:** În această clasă definim un static `adunare()` care primește două obiecte `polinoame` și returnează suma celor două polinomiale ca un nou obiect `Polinom`.

Metoda funcționează prin iterarea exponenților fiecărui termen în ambele polinoame, adăugarea coeficienților corespunzători și adăugarea termenului rezultat la noul polinom folosind metoda `addTerm()` a clasei `polinomiale`.

— **Scadere:** În această clasă am implementat o versiune modificată a adunării prin negarea coeficientului la al doilea obiect polinom din input.

— **Inmultire:** Metoda funcționează prin iterarea pe fiecare termen în ambele polinoame și înmulțirea coeficienților corespunzători și adăugarea termenului rezultat la noul polinom folosind metoda `addTerm()` a clasei `polinomiale`.

— **Derivare:** Metoda inițializează un nou obiect polinomial pentru derivat. Iterează pe fiecare termen în polinomul de intrare și calculează derivatul fiecărui termen folosind regula de putere a diferențierii. Derivatul unui termen $a \cdot x^n$ este $a \cdot n \cdot x^{(n-1)}$. Metoda adaugă fiecare termen derivat la polinomul derivat, sărind peste orice termeni cu coeficienți zero.

Metoda returnează polinomul derivat.

— **Derivare:** Metoda inițializează un nou obiect polinomial pentru integrală. Iterează pe fiecare termen din polinomul de intrare și calculează integrala fiecărui termen folosind regula de putere a integrării. Integrala unui termen ax^n este $(a/(n+1))x^{(n+1)}$. Metoda adaugă fiecare termen integral la polinomul integral.

Metoda returnează polinomul integral.

3.3 GUI

GUI este o interfață grafică care permite utilizatorilor să interacționeze cu sisteme de calcul și aplicații software prin elemente grafice precum butoane, meniuri, casete de dialog și ferestre. Este esențială în majoritatea sistemelor de operare și a aplicațiilor utilizate astăzi

Pentru GUI din calculatorul polinomial am folosit librariile Java AWT si Javax Swing. Interfata are urmatoarele componente:

--**Frame** = “rama” în care se adaugă toate elementele de care avem nevoie pentru buna funcționare a programului. E practice o fereastră care, după utilizare, apăsând butonul X din dreapta sus se poate închide (Exit_on_close).

--**Butoane** = avem 6 butoane în total, fiecare cu funcții diferite:

- Adunare: Adunarea dintre cele 2 polinoame
- Scadere: Scaderea dintre cele 2 polinoame
- Inmultire: Inmultirea dintre cele 2 polinoame
- Derivare: Derivarea primului polinom
- Integrare: Integrarea primului polinom

--**TextFields**: Un TextField este un element grafic care reprezintă un spațiu dreptunghiular în care utilizatorii pot introduce date de la tastatură. În plus, TextField-ul poate fi utilizat și pentru a afișa informații, fără a permite introducerea de date noi de la tastatură.

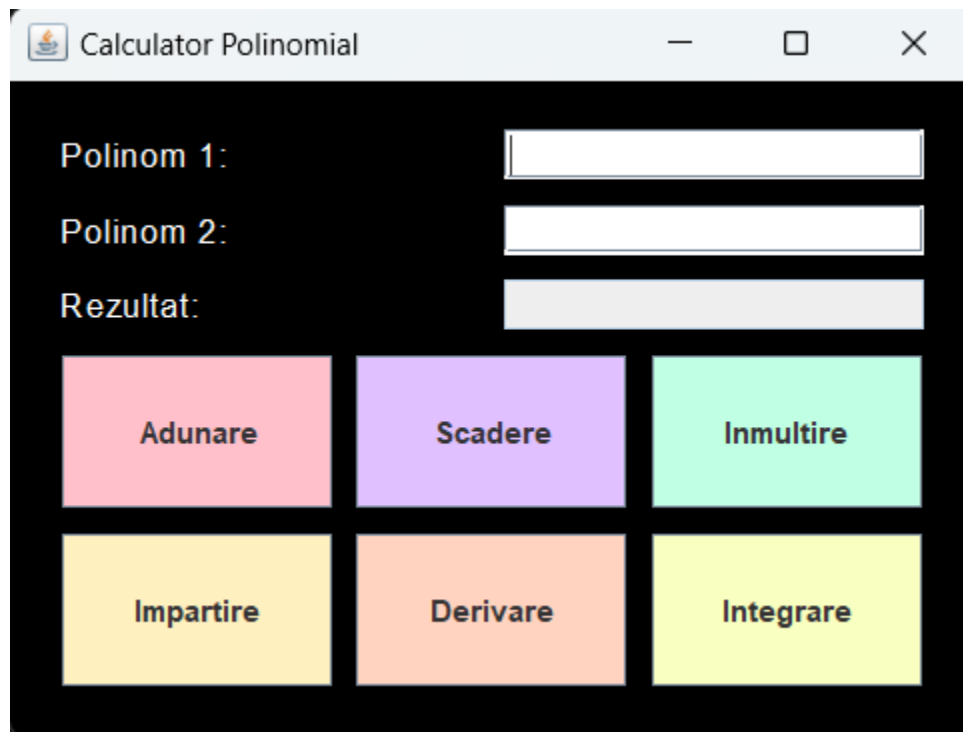
- Text1: datele pentru primul polinom
- Text2: datele pentru al doilea polinom
- textResult: rezultatul operației selectate

--**Label**: Un Label reprezintă de fapt o etichetă care poate fi utilizată pentru a indica informații, titluri sau indicații într-o aplicație software.

Pentru fiecare buton, a fost creată o metodă nouă care utilizează un obiect de tipul `ActionEvent`. Astfel, când are loc o acțiune asupra unui buton (de exemplu, acesta este apăsat), obiectul `ActionEvent` transmite informația către un `ActionListener` care așteaptă astfel de informații. În urma acestei acțiuni, au loc evenimentele asociate cu metoda respectivă (de obicei, se creează un nou polinom numit "rezultat", la care se atribuie rezultatul operației selectate).

Mai jos voi arata o schita si rezultatul final:

```
-----Calculator Polinomial-----
-                                     -
-                                     -
-   Polinom 1:                       -
-   Polinom 2:                       -
-   Rezultat:                        -
-                                     -
-                                     -
-       Adunare      Scadere      Inmultire      -
-                                     -
-       Impartire     Derivare     Integrare      -
-                                     -
-                                     -
-----
```



4. Concluzii

În concluzie, consider că acest proiect mi-a consolidat cunoștințele în domeniul limbajului Java, implementarea paradigmei POO și crearea unei aplicații cu interfață grafică.

5. Bibliografie

<https://stackoverflow.com/questions/8405540/java-regular-expression>

https://www.youtube.com/playlist?list=PLDN4rrl48XKoYR1H6l19hvF_8SMHGdPvk

[https://en.wikipedia.org/wiki/Swing_\(Java\)](https://en.wikipedia.org/wiki/Swing_(Java))