DOCUMENTATIE

TEMA *1*

Polynomial Calculator

NUME STUDENT: Chindris Alexandra – Maria

GRUPA: 30226

# CUPRINS

[1. Obiectivul temei 3](#_Toc95297885)

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

[3. Proiectare 7](#_Toc95297887)

[4. Implementare 11](#_Toc95297888)

[5. Rezultate 12](#_Toc95297889)

[6. Concluzii 13](#_Toc95297890)

[7. Bibliografie 14](#_Toc95297891)

# Obiectivul temei

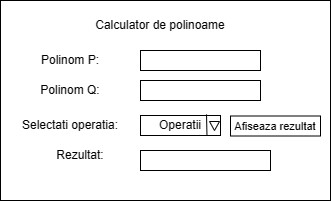
* **Obiectivul principal**

Proiectarea și implementarea unui calculator polinomial cu o interfață grafică corespunzatoare prin care utilizatorul poate insera polinoame, selecta operația matematică care trebuie efectuată și mai apoi sa poata vizualiza rezultatul.

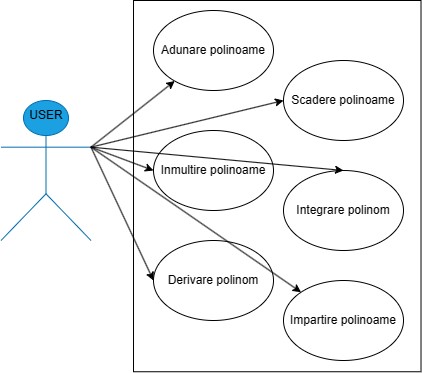
* **Obiectivele secundare**
* Analizarea problemei si identificarea cerintelor – determinam cerintele functionale si cele non-functionale, cat si operatiile matematice ce se pot implementa pentru un calculator polynomial. [Capitolul 2]
* Proiectarea unui calculator polinomial – determinam clasele de care avem nevoie impreuna cu atributele si metodele lor si le organizam in pachete. Determinam legaturile dintre pachete si clase.[Capitolul 3]
* Implementarea unui calculator polinomial – se realizeaza implementarea operatiile matematice pe polinoame si a interfetei grafice. [Capitoul 4]
* Testarea functionarii calculatorului polynomial – testam corectitudinea operatiilor matematice folosind testarea unitara. [Capitolul 5]

# Analiza problemei, modelare, scenarii, cazuri de utilizare

* Cerinte functionale:
  + Calculatorul de polinoame trebuie sa permita utilizatorului sa introduca polinoamele pe care se efectueaza operatiile matematice.
  + Calculatorul de polinoame trebuie sa permita utilizatorului sa selecteze operatia matematica dorita.
  + Calculatorul de polinoame trebuie sa poata aduna doua polinoame.
  + Calculatorul de polinoame trebuie sa poata scadea doua polinoame.
  + Calculatorul de polinoame trebuie sa poata inmulti doua polinoame.
  + Calculatorul de polinoame trebuie sa poata imparti doua polinoame.
  + Calculatorul de polinoame trebuie sa poata deriva un polinom.
  + Calculatorul de polinoame trebuie sa poata integra un polinom.
* Cerinte non-functionale:
  + Calculatorul de polinoame trebuie sa fie usor de folosit de catre utilizator.
  + Calculatorul de polinoame trebuie sa prezinte intuitiv functiile pe care le ofera.



Overall system design:



1. **Use case:** Adunare polinoame

**Primary actor:** User

**Main Success Scenario:**

1. Utilizatorul insereaza doua polinoame in interfata grafica.
2. Utilizatorul selecteaza operatia de “Adunare”
3. Utilizatorul da click pe butonul de “Afiseaza rezultat”
4. Calculatorul de polinoame efectueaza operatia de adunare a celor doua polinoame si afiseaza rezultatul in chenarul “Rezultat”
5. **Use case:** Scadere polinoame

**Primary actor:** User

**Main Success Scenario:**

1. Utilizatorul insereaza doua polinoame in interfata grafica. Primul polinom reprezinta descazutul, iar cel de-al doilea reprezinta scazatorul.
2. Utilizatorul selecteaza operatia de “Scadere”
3. Utilizatorul da click pe butonul de “Afiseaza rezultat”
4. Calculatorul de polinoame efectueaza operatia de scadere a celor doua polinoame si afiseaza rezultatul in chenarul “Rezultat”
5. **Use case:** Inmultire polinoame

**Primary actor:** User

**Main Success Scenario:**

1. Utilizatorul insereaza doua polinoame in interfata grafica.
2. Utilizatorul selecteaza operatia de “Inmultire”
3. Utilizatorul da click pe butonul de “Afiseaza rezultat”
4. Calculatorul de polinoame efectueaza operatia de inmultire a celor doua polinoame si afiseaza rezultatul in chenarul “Rezultat”
5. **Use case:** Impartire polinoame

**Primary actor:** User

**Main Success Scenario:**

1. Utilizatorul insereaza doua polinoame in interfata grafica. Primul polinom reprezinta deimpartitul, iar cel de-al doilea reprezinta impartitorul.
2. Utilizatorul selecteaza operatia de “Impartire”
3. Utilizatorul da click pe butonul de “Afiseaza rezultat”
4. Calculatorul de polinoame efectueaza operatia de impartire a celor doua polinoame si afiseaza rezultatul in chenarul “Rezultat”

**Alternative Sequence:**

* 1. Utilizatorul insereaza polinomul de impartit avand gradul mai mic decat al polinomului la care se imparte.
  2. In chenarul “Rezultat” se va afisa un mesaj care sa indice utilizatorului greseala facuta “Grad P < Grad Q !”, unde P este deimpartitul si Q este impartitorul.
  3. Utilizatorul insereaza valoarea 0 pentru polinomul ce reprezinta impartitorul.
  4. In chenarul “Rezultat” se va afisa un mesaj care sa indice utilizatorului greseala facuta “Impartitorul nu poate sa fie 0 !”.

1. **Use case:** Derivare polinom

**Primary actor:** User

**Main Success Scenario:**

1. Utilizatorul insereaza un polinom in interfata grafica.
2. Utilizatorul selecteaza operatia de “Derivare”
3. Utilizatorul da click pe butonul de “Afiseaza rezultat”
4. Calculatorul de polinoame efectueaza operatia de derivare a polinomului si afiseaza rezultatul in chenarul “Rezultat”
5. **Use case:** Integrare polinom

**Primary actor:** User

**Main Success Scenario:**

1. Utilizatorul insereaza un polinom in interfata grafica.
2. Utilizatorul selecteaza operatia de “Integrare”
3. Utilizatorul da click pe butonul de “Afiseaza rezultat”
4. Calculatorul de polinoame efectueaza operatia de integrare a polinomului si afiseaza rezultatul in chenarul “Rezultat”

**Alternative Sequence:**

1. Utilizatorul insereaza un polinom egal cu 0 in interfata grafica.
2. Cum integral din 0 este constanta C, in chenarul “Rezultat” se va afisa “C – constanta”.

# Proiectare

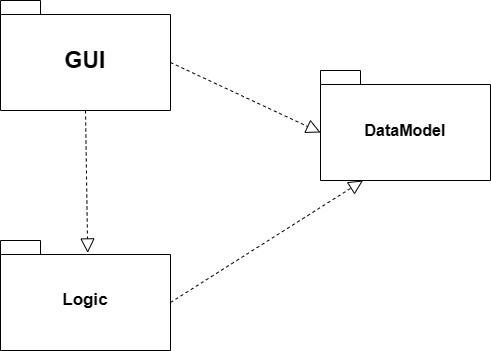
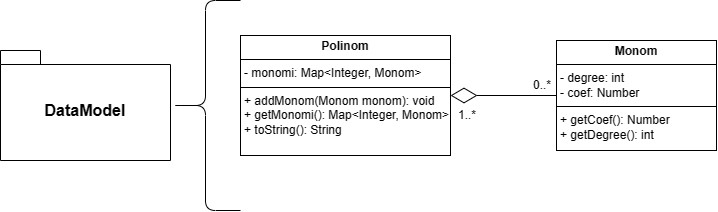
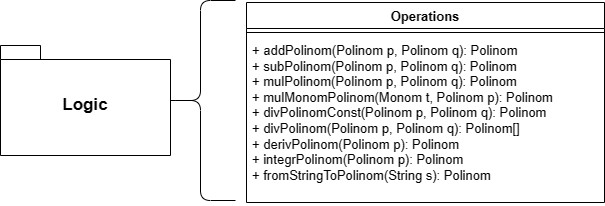


Diagrama UML de pachete:

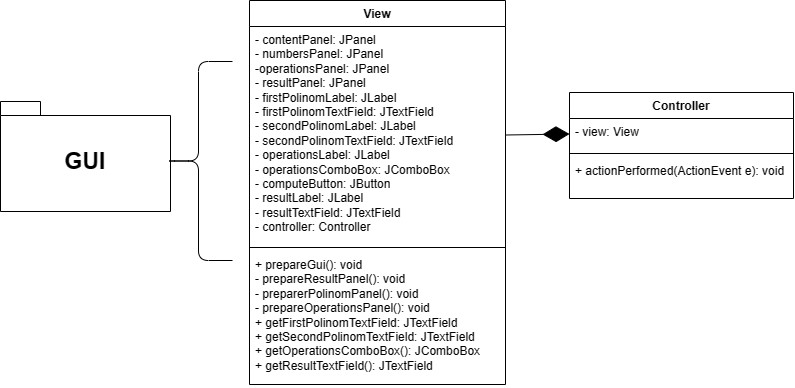
Continutul pachetului DataModel impreuna cu diagrama de clase:



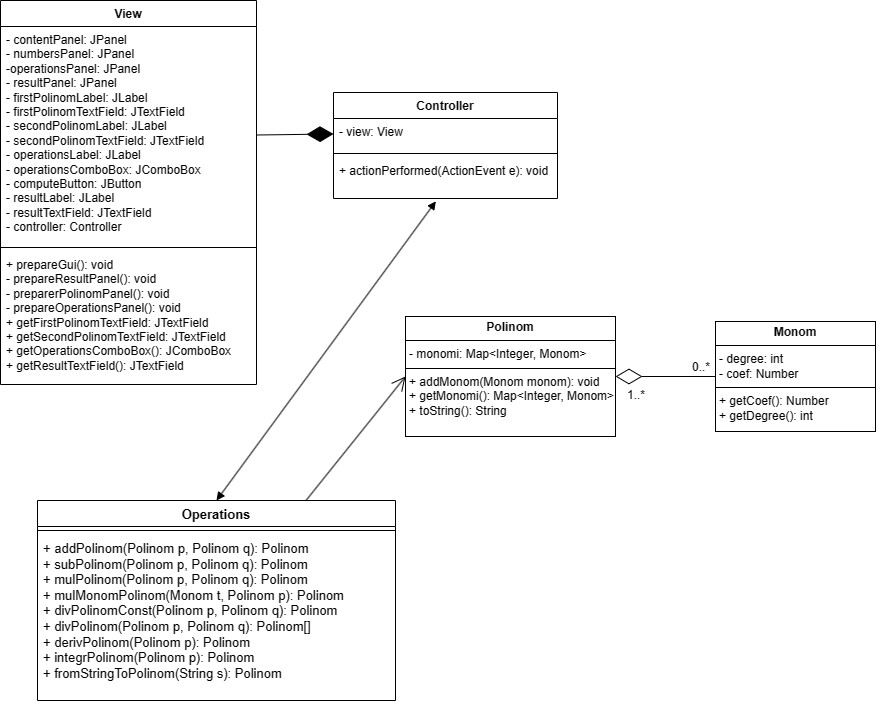
Continutul pachetului Logic:



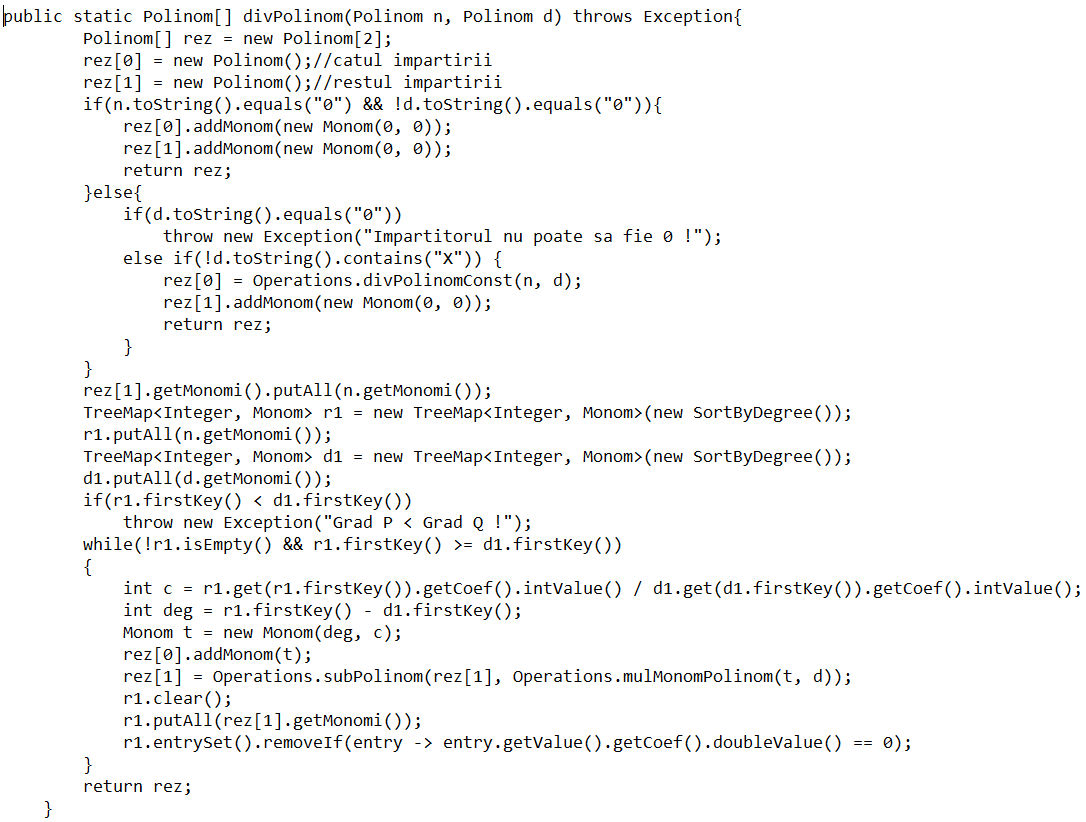
Continutul pachetului GUI:



**Diagrama de clase:**



* Structuri de date folosite:
* Map<Integer, Monom>
* TreeMap<Integer, Monom> pentru care am folosit un Comparator “SortByDegree” astfel incat elementele sa fie ordonate descrescator dupa gradul monomului (Integer).
* Algoritmul de impartire a doua polinoame:

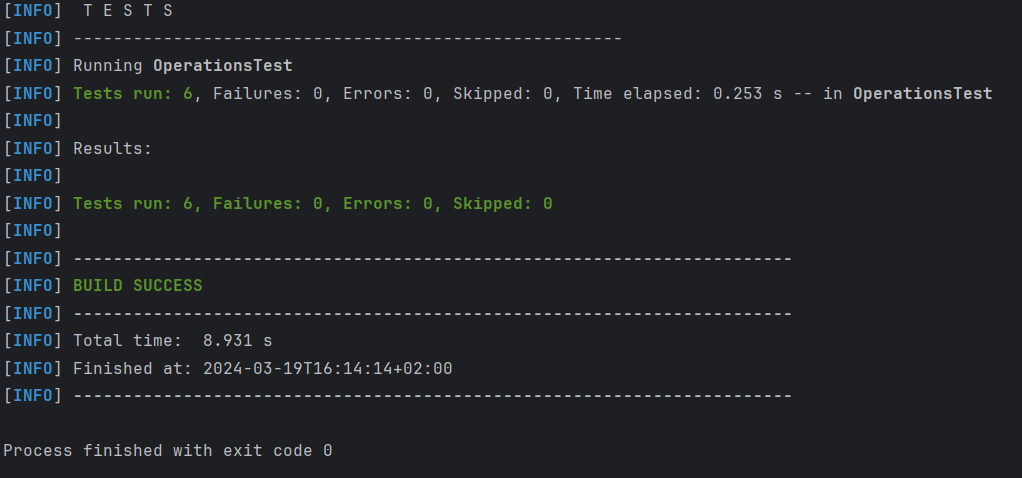


# Implementare

* Clasa Polinom:
  + Contine o structura de date de tip Map<Integer, Monom> unde se pastreaza monomii polinomului.
  + Contine o metoda care adauga monomi la polinom, astfel creandu-se polinomul.
* Clasa Monom:
  + Contine un grad si un coeficient.
* Clasa Operations:
  + Se regasesc metodele pentru operatiile matematice cu polinomi: adunarea, scaderea, inmultirea, impartirea, derivarea si integrarea.
  + Metodele acestei clase sunt statice.
* Clasa View:
  + Implementeaza interfata grafica accesata de utilizator.
  + Contine TextField-uri pentru inserarea polinoamelor si pentru afisarea rezultatelor.
  + Contine un ComboBox pentru a se putea alege operatia care sa se execute.
  + Contine un buton la apasarea caruia controllerul verifica ce fel de operatie s-a selectat pentru a putea seta rezultatul correct.
  + Aceasta are 3 metode pentru a pregati interfata si mai exact pentru partea de introducere a polinoamelor, pentru partea de alegere a operatiei si pentru partea de rezultat.
* Clasa Controller:
  + Controllerul preia informatia si comenzile oferite de utilizator si continua prin a obtine rezultatele dorite prin intermediul clasei Operations. Ca mai apoi aceste informatii sa fie trimise in View.
  + O metoda importanta este “public void actionPerformed(ActionEvent e)” care verifica daca s-a apasat butonul de “Afiseaza rezultat” si in caz afirmativ continua cu verificarea operatiei matematice alese de utilizator si obtinerea rezultatului.

*.*

# Rezultate



* Testele efectuate:

(4 \* X^5 -3 \* X^4 + X^2 -8 \* X + 1) + (3 \* X^4 - X^3 + X^2 + 2 \* X -1) = 4 \* X^5 - X^3 + 2 \* X^2 -6 \* X

(4 \* X^5 -3 \* X^4 + X^2 -8 \* X + 1) - (3 \* X^4 - X^3 + X^2 + 2 \* X -1) = 4 \* X^5 -6 \* X^4 + X^3 -10 \* X + 2

(3 \* X^2 - X + 1) \* (X -2) = 3 \* X^3 -7 \* X^2 + 3 \* X -2

(X^3 -2 \* X^2 + 6 \* X -5) / (X^2 -1) = X -2 rest 7 \* X -7

(X^3 -2 \* X^2 + 6 \* X -5)’ = X^3 -2.0 \* X^2 + 6.0 \* X

Integrala(3 \* X^2 -4 \* X + 6) = X^3 -2.0 \* X^2 + 6.0 \* X

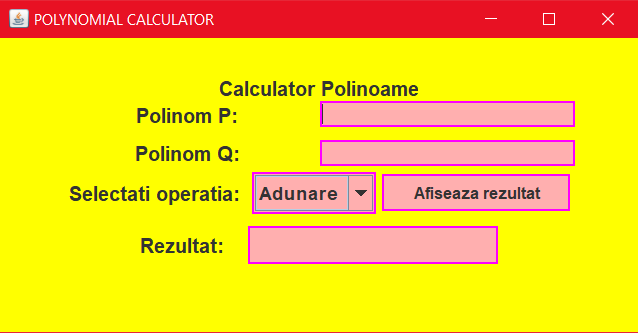
# Concluzii

Calculatorul polinomial realizat poate efectua adunarea, scaderea, inmultirea, impartirea, derivarea si integrarea pe polinoame cu success.

Din aceasta tema am invatat cum sa-mi organizez codul mai bine si in ce ordine trebuie sa lucrez pentru a putea lega informatiile intre ele.

Posibile dezvoltari:

* + In functie de operatia aleasa sa se afiseze unul sau doua campuri de introducere a polinoamelor pentru ca la derivare si integrare este necesar un singur polinom.
  + Interfata grafica sa fie mai prietenoasa cu utilizatorul, de exemplu s-ar putea adauga imagini.
  + In loc sa se introduca polinoamele de la tastatura, s-ar putea realiza chiar o tastatura in interfata grafica, astfel utilizatorul trebuie sa dea click pe butoanele respective.



# Bibliografie

1. [Lesson: Regular Expressions (The Java™ Tutorials > Essential Java Classes) (oracle.com)](https://docs.oracle.com/javase/tutorial/essential/regex/index.html)
2. [Quantifiers in Java - GeeksforGeeks](https://www.geeksforgeeks.org/quantifiers-in-java/)
3. [regex101: build, test, and debug regex](https://regex101.com/)
4. [Trail: Creating a GUI With Swing (The Java™ Tutorials) (oracle.com)](https://docs.oracle.com/javase/tutorial/uiswing/)