DOCUMENTATIE

TEMA *1*

NUME STUDENT: Rosca Andrei Bogdan

GRUPA: 30226

# 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)

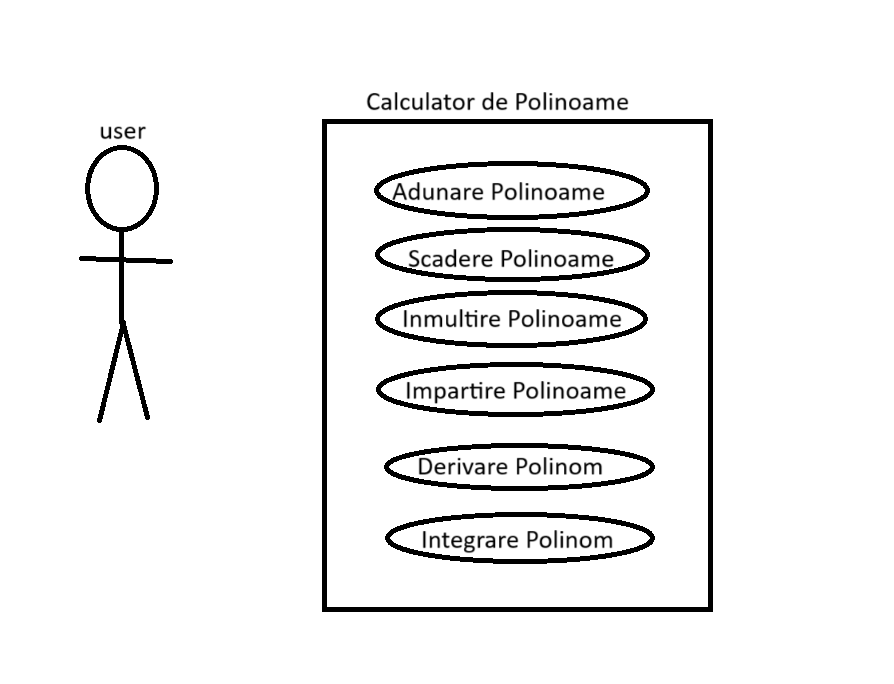
[7. Bibliografie 3](#_Toc95297891)

# Obiectivul temei

Obiectivul principal al temei este implementarea unui calculator de polinoame in Java ,folosind principiile de programare POO.Obiectivele secundare sunt implementarea operatiilor ce pot fi efectuate cu calculatorul, creearea unei interfete grafice , testarea codului si organizarea acestuia

# Analiza problemei, modelare, scenarii, cazuri de utilizare

Scopul acestei teme este să ofere un calculator de polinoame cu o interfață grafică intuitivă, care să permită utilizatorului să citească doua polinoame de la tastatura, să efectueze operații asupra lor și să vizualizeze rezultatele. În același timp, se pune accent pe asigurarea unei performanțe optime și a unei experiențe de utilizare ușoare și plăcute pentru utilizator.



Caz de utilizare 1: înmultire de polinoame

Actor: user

Scenariu de utilizare ideal:

1.Utilizatorul introduce doua polinoame in interfata grafica

2.Utilizatorul selecteaza operatia de înmultire

3.Calculatorul efectueaza operatia de înmultire si rezultatul este afisat in interfata grafica

Caz de utilizare 2: derivarea primului polinoam

Actor: user

Scenariu de utilizare ideal:

1.Utilizatorul introduce un polinom in interfata grafica

2.Utilizatorul selecteaza operatia de derivare

3.Calculatorul efectueaza operatia de derivare si rezultatul este afisat in interfata grafica

Se procedeaza la fel si pentru celelalte cazuri de utilizare: adunarea, scaderea , împartirea si integrarea de polinoame

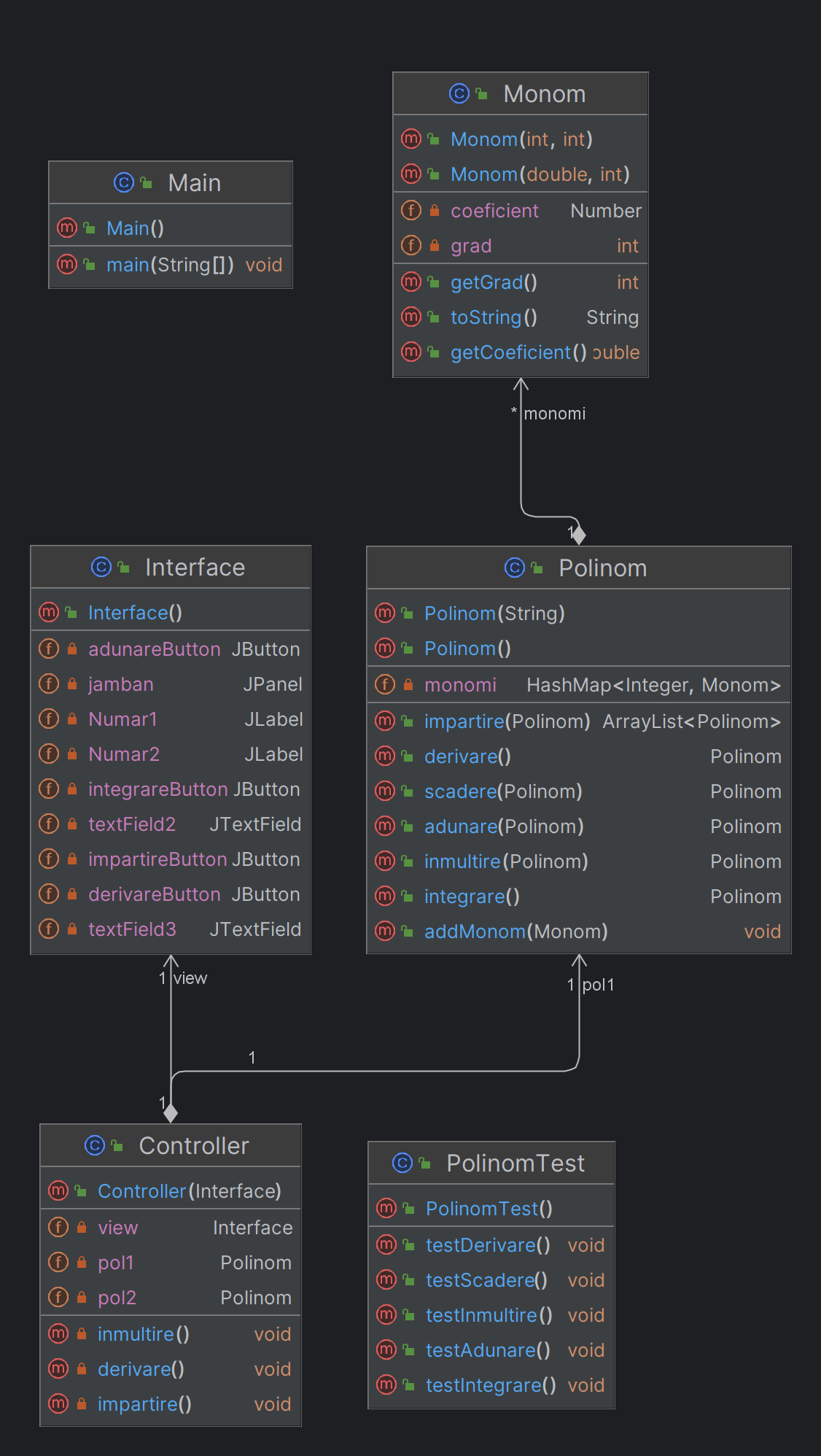
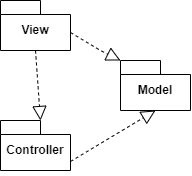
Probleme ce pot fi intalite :

Utilizatorul introduce gresit un polinom lucru care va duce la un mesaj

Utilizatorul împarte un numar cu 0 lucru care va duce la mesaj

In aceste cazuri ne reintoarcem la pasul 1.

1. Proiectare

Diagrama de clase Diagrama de pachete 

**Î**n diagrama de clase se pot observa clasele folosite in acest proiect , acestea fiind organizate in 3 pachete in functie de functionalitatea lor.Pe langa aceastea mai exita o clasa PolinomTest care se ocupa de testarea operatiilor.

Pentru proiectarea calculatorului am ales sa folosesc o clasa polinom si o clasa monom . **Î**n clasa polinom retin într-o colectie de tip HashMap toate monoamele din care acesta este format . Pe polinoame se pot aplica operatii care sunt implementate prin metode.

# **Implementare**

Clasa Polinom permite crearea, manipularea și efectuarea de operații matematice pe polinoame, cum ar fi adunare, scădere, înmulțire, împărțire, derivare și integrare.  
Aceasta contine urmatoarele elemente:

**Atribute:**  
 - monomi: Un obiect de tip HashMap<Integer, Monom> care stochează monomii polinomului, unde cheia este gradul monomului, iar valoarea este monomul însuși.

**Constructori:**

- Polinom(): Inițializează un polinom gol, fără monomi.  
 - Polinom(String s): Inițializează un polinom pe baza unui șir de caractere dat ca parametru, parsând șirul ,cu ajutorul expresilor regulate ,pentru a extrage monomii și adăugându-i la polinom.

**Metode**:  
 - addMonom(Monom x): Adaugă un monom dat ca parametru la polinom.  
 - adunare(Polinom x): Realizează adunarea polinomului curent cu un alt polinom dat ca parametru și returnează rezultatul adunării.  
 - scadere(Polinom x): Realizează scăderea polinomului curent cu un alt polinom dat ca parametru și returnează rezultatul scăderii.  
 - inmultire(Polinom x): Realizează înmulțirea polinomului curent cu un alt polinom dat ca parametru și returnează rezultatul înmulțirii.  
 - impartire(Polinom x): Realizează împărțirea polinomului curent cu un alt polinom dat ca parametru și returnează un ArrayList conținând câtul și restul împărțirii.  
 - derivare(): Calculează derivata polinomului și returnează rezultatul.  
 - integrare(): Calculează integrala polinomului și returnează rezultatul.  
 - toString(): Returnează o reprezentare sub formă de șir de caractere a polinomului.

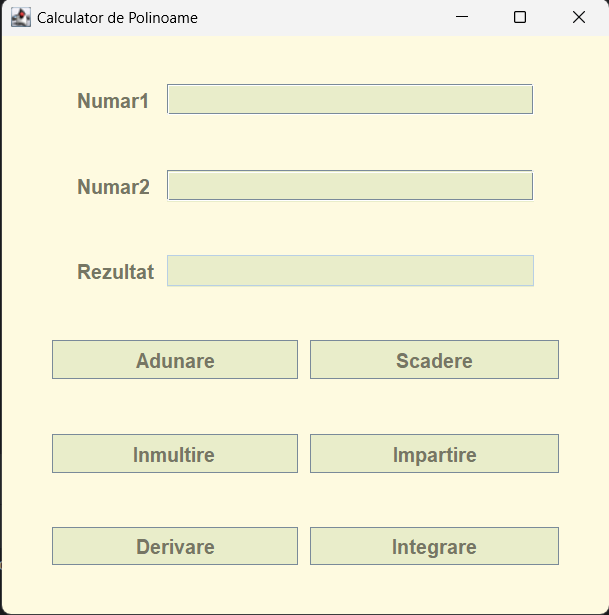
Clasa Monom este utilizată pentru a reprezenta monomii în cadrul polinoamelor.  
Aceasta conține următoarele elemente:

**Atribute**:  
 - coeficient: Un obiect de tip Number care stochează coeficientul monomului , se foloseste Number deoarece coeficientul poate fi si int si double.  
 - grad: Un număr întreg care reprezintă gradul monomului.

**Constructor**i:  
 - Monom(double coeficient, int grad): Inițializează un monom cu coeficientul de tip double și gradul specificate ca parametri.  
 - Monom(int coeficient, int grad): Inițializează un monom cu coeficientul de tip int și gradul specificate ca parametri.

**Metode**:  
 - getCoeficient(): Returnează coeficientul monomului sub formă de double.  
 - getGrad(): Returnează gradul monomului sub formă de int.  
 - toString(): Returnează o reprezentare sub formă de șir de caractere a monomului, în care coeficientul și gradul sunt incluse conform cerințelor. Această reprezentare poate fi utilizată pentru afișarea monomului într-o formă ușor de înțeles pentru utilizator.

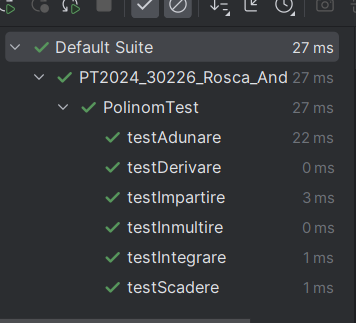
Clasa Interface este o componentă a interfeței utilizator pentru un calculator de polinoame. Aceasta utilizează framework-ul Swing din Java pentru a crea elementele interfeței utilizator și a le lega de acțiunile utilizatorului prin intermediul unor ascultători de evenimente.

**Componente**:  
 - JPanel principal: Panoul principal al interfeței, care conține toate celelalte componente.  
 - JTextField textField1, textField3: Câmpuri de text în care utilizatorul poate introduce coeficienții și gradele polinoamelor și rezultatul operațiilor.  
 - JTextField textField2: Câmp de text în care utilizatorul poate sa vada rezultatul operațiilor  
 - JLabel Numar1, Rezultat, Numar2: Etichete pentru a indica scopul fiecărui câmp de text.  
 - JButton adunareButton, scadereButton, inmultireButton, impartireButton, integrareButton, derivareButton: Butoane pentru a permite utilizatorului să efectueze operații de adunare, scădere, înmulțire, împărțire, integrare și derivare a polinoamelor.  
**Constructor**:  
 - Interface(): Inițializează interfața și afișează fereastra principală a aplicației.  
**Metode**:  
 - Metodele getButonAdunare, getButonScadere, getButonInmultire, getButonImpartire, getButonDerivare, getButonIntegrare: Returnează butoanele corespunzătoare operațiilor pentru a le lega de ascultători de evenimente în controller  
 - Metodele tf1, tf2, tf3: Returnează câmpurile de text pentru a permite accesul asupra lor din controller  
 

Clasa Controller este responsabilă pentru gestionarea logică a interacțiunii dintre interfața utilizator și modelele de date în cadrul aplicației de calculator de polinoame.  
Aceasta contine urmatoarele elemente:  
**Atribute**:  
 - view: Reprezintă interfața grafică a aplicației.  
 - pol1, pol2: Reprezintă polinoamele pe care se efectuează operațiile.  
**Constructor**:  
 - Controller(Interface view): Inițializează controller-ul și primește interfața utilizator ca parametru.  
**Metode**:  
 - adunare(), scadere(), inmultire(), impartire(), derivare(), integrare(): Aceste metode adaugă ascultători de evenimente pentru butoanele corespunzătoare din interfața și definesc operatiile care trebuie facute atunci când aceste butoane sunt apăsate. Fiecare metodă preia datele introduse de utilizator din câmpurile de text, creează obiecte Polinom corespunzătoare și efectuează operația specificată utilizând metodele din clasa Polinom. Rezultatul este afișat în câmpul de text corespunzător.

# Rezultate

Pentru verificarea rezultatelor am folosit un calculator online pentru polinoame in care am introdus un exemplu pentru fiecare operație iar mai apoi am introdus in clasa Polinom test in metodele de testare a fiecarei operatii aceleasi date de intrare si am verificat daca iesirea este la fel.



# Concluzii

Din aceasta temă am învățat cum se realizează o aplicație Java de la 0 ,cum se folosește o structură de tip MVC și cum este de folos testarea programului. Am învățat și cum se utilizează Java Swing ,pentru a ușură realizarea interfeței grafice și cum se folosește un HashMap și în ce cazuri este util să-l folosim. O alta chestie noua învățată în acest proiect sunt expresiile regulate pe care le am folosti pentru a transforma un string în monoame.

Că și dezvoltări ulterioare a acestui calculator ar putea fi extinderea operațiilor realizate pe polinoame,cum ar fi aflarea rădăcinilor . Se poate dezvolta și interfață grafica într-o interfață mai complexă care iți permite să tastezi numerele direct din butoane.

# Bibliografie

1. https://dsrl.eu/courses/pt/