**DOCUMENTATIE**

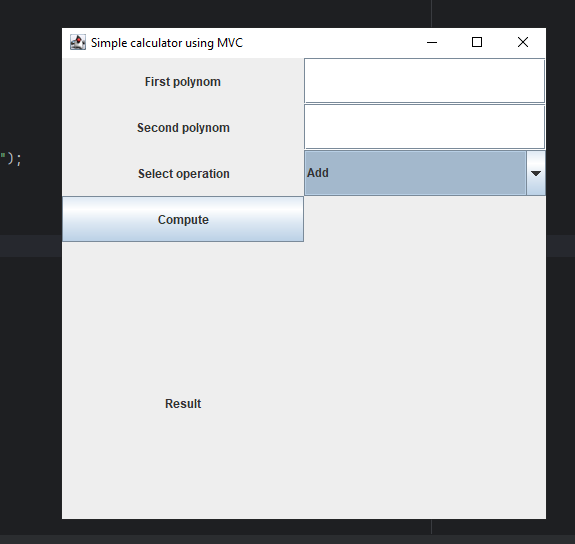
**TEMA 1**

NUME STUDENT: Ciupe Bogdan Adrian

GRUPA: 8

# **CUPRINS**

1. Obiectivul teme
2. Analiza problemei, modelare, scenarii, cazuri de utilizare
3. Proiectare
4. Implementare
5. Rezultate
6. Concluzii
7. Bibliografie



**1. Obiectivul temei**

(i) Obiectivul principal: Dezvoltarea unei aplicații Java pentru manipularea și efectuarea operațiilor asupra polinoamelor (adunare, scădere, înmulțire, împărțire, derivare).

(ii) Obiective secundare:

Analiza și proiectarea structurilor de date necesare reprezentării polinoamelor.

Implementarea interfeței utilizator pentru interacțiunea cu aplicația.

Crearea algoritmilor pentru operații polinomiale.

Testarea și validarea funcționalităților implementate.

**2. Analiza problemei modelare scenarii cazuri de utilizare**

Funcționalități:

Introducerea polinoamelor prin intermediul interfeței grafice.

Alegerea operației dorite (adunare, scădere, etc.).

Afișarea rezultatului operației.

Cazuri de utilizare:

Utilizatorul introduce două polinoame și selectează tipul operației.

Sistemul calculează rezultatul și îl afișează utilizatorului.

**3. Proiectare**

Arhitectura OOP a aplicației:

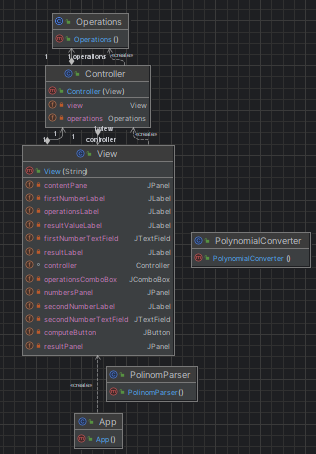
Operations - clasa care implementează operațiile polinomiale.

PolinomParser - clasa pentru parsarea string-urilor în polinoame.

PolynomialConverter - clasa pentru conversia polinoamelor în string-uri.

Interfața grafică View și controlerul Controller.

Diagrama UML:



**4. Implementare**

Descrierea claselor:

-Operations: Clasa OperatiiPolinoame conține metode pentru efectuarea operațiilor asupra polinoamelor, cum ar fi adunarea, scăderea, înmulțirea, împărțirea, derivarea si integrarea. Fiecare metodă primește ca argumente două polinoame și returnează un nou polinom care reprezintă rezultatul operației, mai putin derivarea si integrarea care primesc doar un polinom.

-PolinomParser: Clasa PolinomParser este responsabilă pentru analiza șirurilor de caractere și convertirea lor în obiecte TreeMap. Include metode care utilizează expresii regulate pentru a descompune șirurile de caractere în termeni individuali și pentru a extrage coeficienții și gradele acestora.

-PolynomialConverter: Clasa PolynomialConverter este utilizată pentru a converti obiectele Polinom înapoi în șiruri de caractere pentru afișare. Include metode care iau un Polinom ca intrare și construiesc un șir de caractere care reprezintă polinomul într-un format ușor de citit.

**5. Rezultate**

Testarea Funcționalității Clasei Operations

Testele unitare au fost scrise folosind JUnit pentru a valida corectitudinea metodelor implementate în clasa Operations. Aceste metode includ add, subtract, multiply, derivative, și integration. Rezultatele testelor sunt următoarele:

Testul Metodei add:

Scenariu testat: Adunarea a două polinoame "3x^2 + 2" și "4x + 3".

Rezultat așteptat: "3x^2 + 4x + 5".

Rezultat obținut: Testul a trecut cu succes, confirmând corectitudinea implementării adunării.

Testul Metodei subtract:

Scenariu testat: Scăderea a două polinoame "3x^2 + 2" din "4x + 3".

Rezultat așteptat: "3x^2 - 4x - 1".

Rezultat obținut: Testul a trecut cu succes, confirmând corectitudinea implementării scăderii.

Testul Metodei multiply:

Scenariu testat: Înmulțirea a două polinoame "3x^2 + 2" și "4x - 3".

Rezultat așteptat: "12x^3 - 9x^2 + 8x - 6".

Rezultat obținut: Testul a trecut cu succes, confirmând corectitudinea implementării înmulțirii.

Testul Metodei derivative:

Scenariu testat: Derivarea polinomului "3x^2 + 2x".

Rezultat așteptat: "6x + 2".

Rezultat obținut: Testul a trecut cu succes, indicând corectitudinea implementării derivării.

Testul Metodei integration:

Scenariu testat: Integrarea polinomului "3x^2 + 2x".

Rezultat așteptat: "x^3 + x^2".

Rezultat obținut: Testul a trecut cu succes, indicând corectitudinea implementării integrării.

Aceste teste asigură că metodele clasei Operations funcționează așa cum este așteptat și că pot gestiona diferite scenarii de intrare. Succesul acestor teste este esențial pentru validarea logicii matematice a operațiilor pe polinoame.

**6. Concluzii**

Acest proiect a oferit o experiență valoroasă în aplicarea conceptelor de programare orientată pe obiecte și în înțelegerea matematicii din spatele operațiilor cu polinoame. Printre principalele lecții învățate se numără:

Implementarea eficientă a algoritmilor: Dezvoltarea algoritmilor pentru operații polinomiale a necesitat o gândire logică și analitică detaliată.

Manipularea structurilor de date: Folosirea TreeMap pentru a stoca polinoamele a subliniat importanța alegerii structurii de date potrivite.

**7. Bibliografie**

Pentru realizarea acestui proiect, următoarele resurse și materiale au fost de un real folos:

1.GeeksforGeeks

2.Tutoriale de pe youtube

3.Stack Overflow: