

# **DOCUMENTATIE**

## **TEMA 1**

NUME STUDENT: Szabo Szilard  
GRUPA: 30229

# CUPRINS

<i>1. Obiectivul temei .....</i>	<i>3</i>
<i>2. Analiza problemei, modelare, scenarii, cazuri de utilizare .....</i>	<i>3</i>
<i>3. Proiectare .....</i>	<i>4</i>
<i>4. Implementare .....</i>	<i>5</i>
<i>5. Rezultate .....</i>	<i>6</i>
<i>6. Concluzii .....</i>	<i>6</i>
<i>7. Bibliografie .....</i>	<i>6</i>

## 1. Obiectivul temei

a) Obiectivul principal este de a ști să implementezi diferite operații cu polinoame și de a aplica principiile de OOP pentru a face proiectul să funcționeze conform cerinței și segmentarea în clase și unirea lor corespunzătoare.

b) Operațiile corespunzătoare sunt adunare, scădere, înmulțire, împărțire, derivare și integrare.

Primele 4 fiind între 2 polinoame, iar ultimele 2 fiind pe un singur polinom.

c) Tot programul trebuie să poată fi folosit cu ajutorul unei interfețe grafice implementat cu diferite componente Swing.

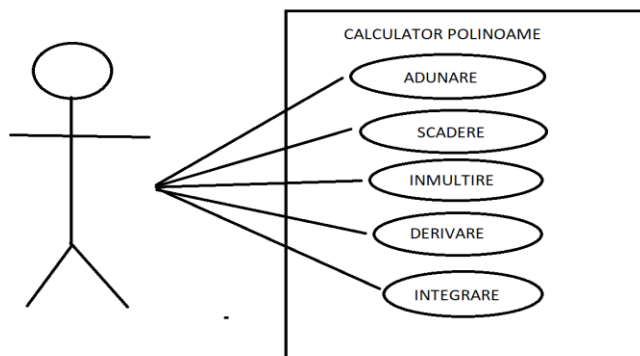
## 2. Analiza problemei, modelare, scenarii, cazuri de utilizare

Cum putem observa și în obiectivul temei, problema este rezolvarea operațiilor de adunare, scădere, înmulțire, împărțire, derivare și integrare aplicate pe polinoame/polinom.

Un polinom este format din mai multe monoame, separate prin semnul + sau -, fiecare monom are un coeficient și o putere, puterea fiind unică.

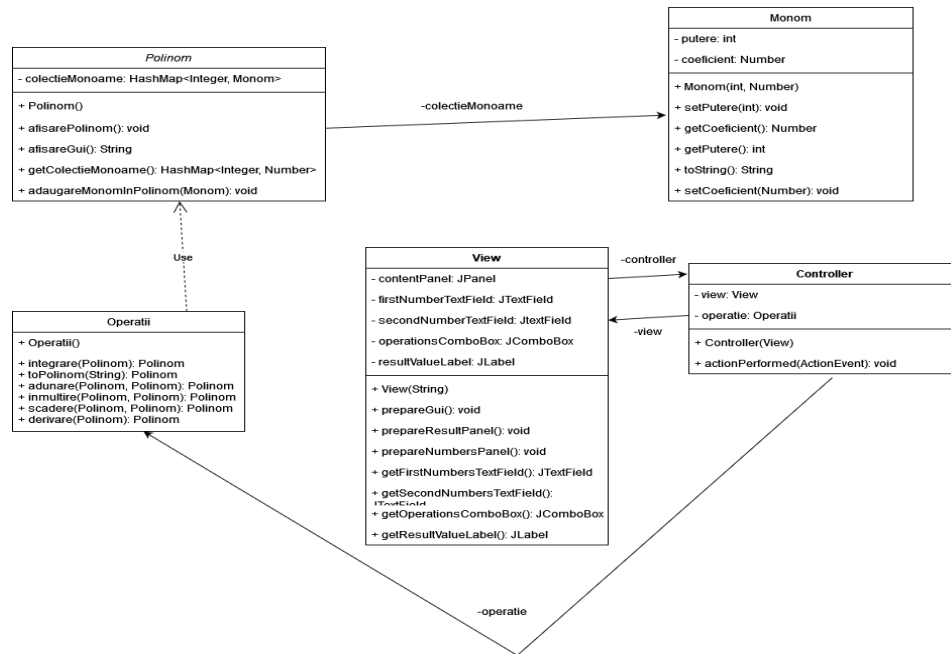
Utilizatorul va trebui să introducă un polinom sau două polinoame (depinde de operația pe care vor să o utilizeze), aceste polinoame se vor introduce în TextBox-uri (forma acceptată este **COEFICIENT** $x^{\text{PUTERE}}$  => ex:  $4x^3-2x^3+5x^1-7x^0$ , altfel calculatorul ori nu va returna nimic, ori o va face eronat), iar operația o vor selecta dintr-o listă de sub TextBox-uri, ulterior se va apăsa de "Confirm" și se va afișa polinomul rezultat în Panel-ul din dreapta jos, paralel cu textul "Rezultat".

Este o interfață foarte sugestivă și simplă, deci utilizatorii nu vor avea probleme să o folosească.



### 3. Proiectare

Diagrama UML de clase este:



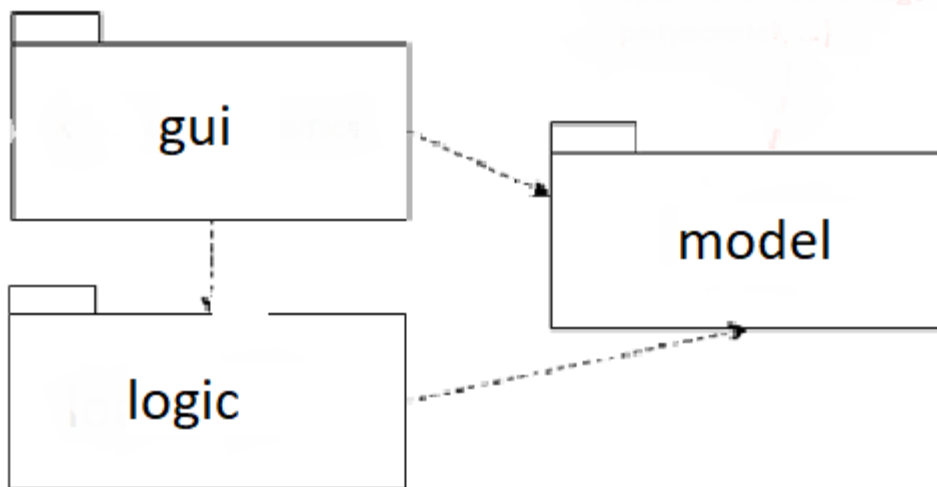
- *Definirea clasei polinom: Aceasta conține o listă de monoame stocate într-o structură de date de tip HashMap care are ca elemente puterea monomului și monomul propriu zis. HashMap este o structură de date eficientă pentru căutarea, inserția și ștergerea datelor, deoarece utilizează funcții hash pentru a accesa datele, ceea ce înseamnă că timpul de căutare și accesare a datelor nu depinde de dimensiunea colecției.*

- *Proiectarea interfeței grafice: Am utilizat Java Swing pentru a proiecta interfața grafică. De asemenea, clasa Controller implementează interfața ActionListener prin metoda actionPerformed pentru a verifica interacțiunea utilizatorului cu programul.*

*Pentru aceasta temă am folosit mai multe pachete. Pachetul gui cuprinde clasele specifice interfeței grafice, pachetul model conține clasele care modelează obiectele specifice programului, iar pachetul logic, clasa care dezvoltă algoritmi pentru operațiile matematice ale calculatorului de polinoame.*

*Paradigma OOP (Orientată pe obiecte) este un concept de programare care se concentrează pe modelarea obiectelor și relațiilor dintre acestea. Java este un limbaj de programare care a fost conceput pentru a fi orientat pe obiecte, ceea ce înseamnă că majoritatea codului scris în Java utilizează principiile OOP. Principiul de bază pe care m-am axat a fost încapsularea datelor prin definirea câmpurilor claselor private.*

Diagrama UML de pachete este:



*Model contine clasele Polinom si Monom*

*Logic contine clasa in care se efectueaza operatiile matematice*

*Gui contine clasele View si Controller pentru interfata grafica*

## 4. Implementare

*Clasa Monom are ca si campuri puterea sub forma de intreg si coeficientul declarat de tipul clasei invelitoare Number. Metodele din aceasta clasa sunt reprezentate de catre un constructor potrivit, metode de obtinere si setare ale atributelor.*

*Clasa Polinom are ca si camp o structura de date care stocheaza monoamele din polinom. In aceasta clasa metodele implementate efectueaza adăugarea unui monom intr-un polinom si transformarea acestuia intr-un obiect de tipul clasei String folosita la afisarea rezultatului in interfata grafica.*

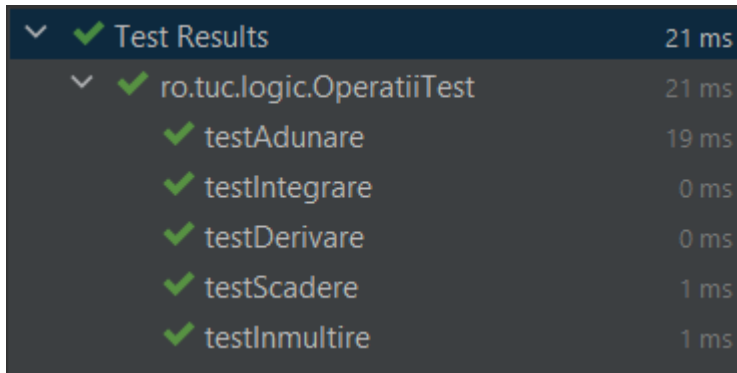
*Clasa Operatii cuprinde metodele de adunare, scadere, inmultire, derivare si integrare a polinoamelor. In plus aici am creat metoda care transforma un obiect de tipul clasei String intr-un polinom.*

*Clasa View se ocupa de crearea interfetei grafice. Aceasta contine metode care pregatesc componentele vizuale, care ii ofera utilizatorului oportunitatea de a introduce 2 polinoame, de a selecta operatia dorita si de a vizualiza rezultatul. Interfata grafica este alcatuita din 2 casete Text pentru datele de intrare, un ComboBox pentru calculul acestora si un buton care genereaza raspunsul in caseta Text corespunzatoare datelor de iesire.*

*Clasa Controller preia selectia utilizatorului, iar la apasarea butonului acesta seteaza textul aferent rezultatului operatiei matematice.*

## 5. Rezultate

Am creat o clasa `OperatiiTest` care se foloseste de testarea unitara pentru a verifica algoritmi utilizati pentru implementarea operatiilor matematice. Cu ajutorul instructiunii `assertEquals` m-am asigurat ca acestia se comporta conform asteptarilor. Totodata am testat metoda care converteste un polinom intr-un rezultat sub forma de text. Dupa ce JUnit a aprobat succesul testelor, m-am convins ca aplicatia functioneaza in parametri optimi.



✓	Test Results	21 ms
✓	ro.tuc.logic.OperatiiTest	21 ms
✓	testAdunare	19 ms
✓	testIntegrare	0 ms
✓	testDerivare	0 ms
✓	testScadere	1 ms
✓	testInmultire	1 ms

## 6. Concluzii

Am invatat sa utilizez arhitectura de modelare `ModelViewController` si cum sa impart codul creat de mine in cele 3 pachete corespunzatoare.

Fiind constrans ca o metoda sa nu fie mai lunga de 30 de linii am reusit sa gandesc mai eficient modul de implementare si cum sa imi organizez codul.

Pentru a manipula un sir de caractere am invatat cum sa folosesc clasele `Pattern` si `Matcher` astfel incat sa il pot transforma intr-un polinom.

O imbunatatire ulterioara ar fi implementarea operatiei de impartire a 2 polinoame sau aplicarea unor algoritmi folosindu-ma de operatiile de baza deja dezvoltate. De exemplu implementarea de derivate si de un ordin superior.

## 7. Bibliografie

Referintele pe care le-am consultat pe parcursul implementarii temei sunt:

1. <https://dsrl.eu/courses/pt/>
2. <https://www.baeldung.com/java-hashmap>
3. [https://users.utcluj.ro/~igiosan/teaching\\_poo.html](https://users.utcluj.ro/~igiosan/teaching_poo.html)
4. <https://www.geeksforgeeks.org/java/>