

# DOCUMENTATIE

## TEMA 1

NUME STUDENT: Salajan Adrian-Antonio  
GRUPA: 30222

# CUPRINS

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

## 1. Obiectivul temei

Obiectivul principal al temei este dezvoltarea unei aplicații care poate face calcule pe polinoame și dispune de o interfață grafică din care utilizatorul poate introduce polinoame de la tastatură și alege operația care urmează să fie executată.

Obiective secundare:

1. Proiectare unei interfețe grafice folosind Java Swing.
2. Citirea polinomului introdus de utilizator de la tastatură
3. Prelucrarea datelor citite astfel încât să putem efectua operații pe ele
4. Implementarea operațiilor pe polinoame:
  - a. Adunarea a două polinoame
  - b. Scaderea a două polinoame
  - c. Înmulțirea a două polinoame
  - d. Împărțirea a două polinoame
  - e. Derivarea și integrarea unui polinom
5. Afișarea rezultatelor obținute în interfața grafică sub forma de text.

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

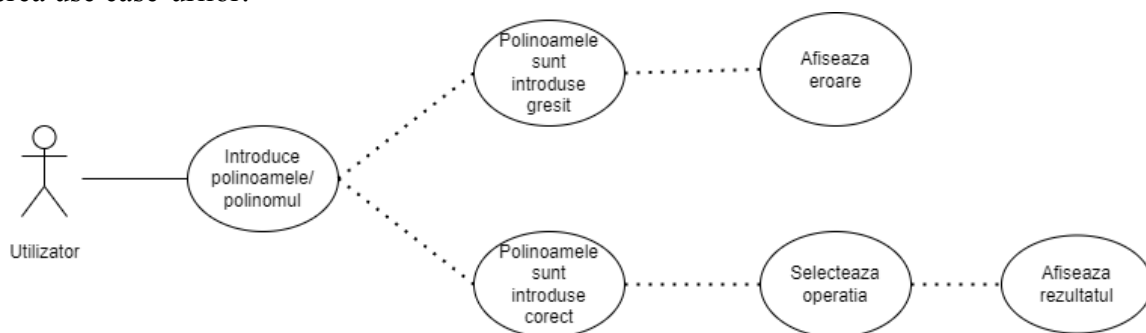
Cerinte functionale:

1. Implementarea unei interfețe de utilizator pentru aplicație.
2. Implementare operațiilor de : adunare, scadere, inmultire, impartire,derivare,integrare.

Cerinte non-functionale:

1. Folosirea unei abordări bazate pe principiile programării orientate pe obiect.
2. Folosirea unei structuri de tip Map pentru modelarea polinomului.
3. Folosirea pentru a itera structurile de date de foreach în loc de for(int i=0;...).
4. Implementare de clase cu maxim 300 de linii și metode cu maxim 30 de linii.
5. Folosirea convențiilor pentru nume în Java.
6. Documentatie calitativa.

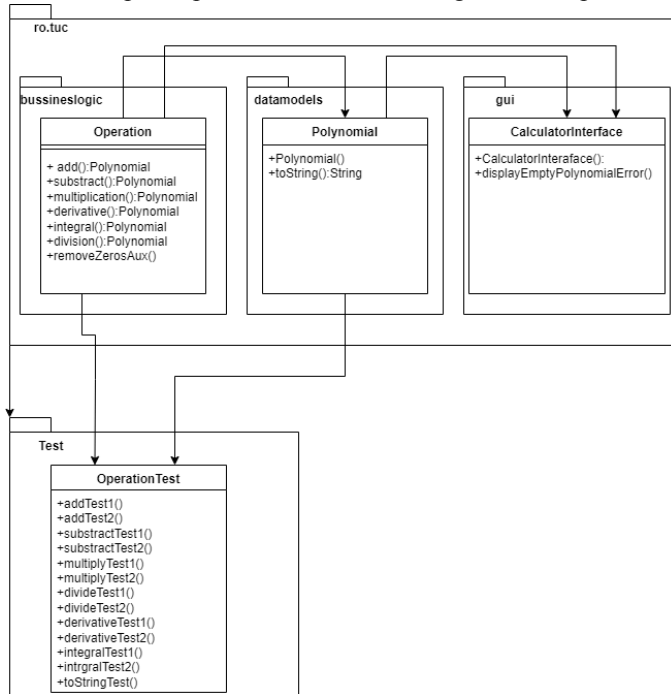
Descrierea use case-urilor:



Utilizatorul introduce unul sau două polinoame în funcție de operația pe care dorește ca programul să o performeze în cazul în care utilizatorul lasă un câmp necesar gol programul afișează un mesaj de eroare. Dacă utilizatorul introduce polinoamele corect și selectează informațiile la care dorește să afle rezultatul programul afișează rezultatul.

### 3. Proiectare

Programul este împărțit în doua pachete principale, unul care contine programul propriu-zis și cealalta care contine partea de testare, partea de program este și ea la randul ei impartita in 3 pachete fiecare ocupandu-se cu o parte esențială a programului, clasele modeleaza fiecare obiecte din lumea reala și respecta principiul incapsulari, mai multe detalii despre implementarea acestora se găsesc în capitolul 4.



### 4. Implementare

#### 1. Clasa Polynomial:

##### a. Campuri:

- i. map: campul map stochează reprezentarea polinomului sub forma unui TreeMap

##### b. Metode:

- i. Polynomial(), constructorul clasei, aceasta metoda primește ca și parametru un string din care extrage datele pentru construcția polinomului pe care le introduce într-o structură de tipul TreeMap.
- ii. toString(), metoda extrage din structura de tip TreeMap puterea și coeficientul pentru fiecare termen al polinomului modelează termenul cu ajutorul metodelor auxiliare și îl adaugă într-un String pentru a putea fi afișat în interfață.

#### 2. Clasa Operation:

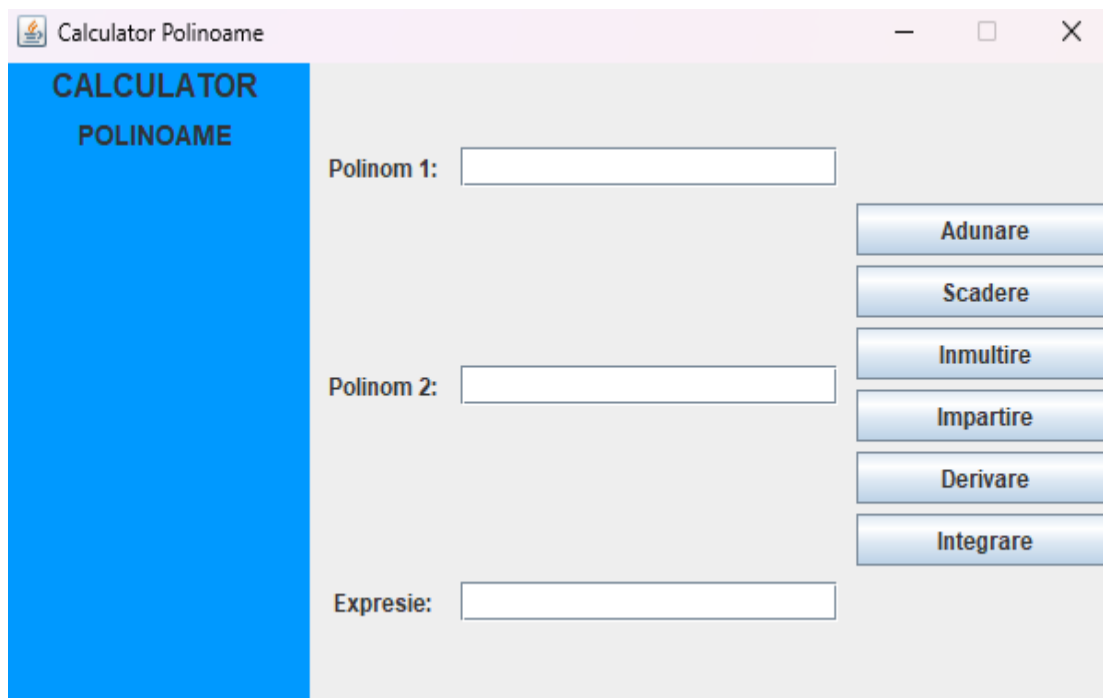
- a. Campuri: Clasa Operation nu are niciun camp deoarece folosim doar metodele statice pentru a returna rezultatul operației.

- b. Metode: sunt toate metode statice care returnează rezultatul operației pe care o descriu

- i. add(): Implementează operația de adunare, itereaza peste primul polinom, dacă nu găsește puterea termenului în al doilea îl adaugă

în rezultat, altfel aduna coeficientii acestora. Când iterează peste al doilea polinom adăuga doar termenii care nu se găsesc în primul în polinomul rezultat, la final returnează rezultatul.

- ii. `subtract()`: Implementează operația de scădere, algoritmul este identic cu cel de adunare, dar operația dintre coeficienți este de scădere în loc de adunare. Rezultatul este returnat sub forma unui polinom.
  - iii. `multiplication()`: Implementează operația de înmulțire, programul înmulțește fiecare termen din primul polinom cu fiecare termen din al doilea polinom și pune rezultatul în mapa la adresa data de suma puterilor cu valoare înmulțirii coeficienților. Returnează un polinom care reprezintă rezultatul.
  - iv. `division()`: Folosind algoritmul de împărțire a doua polinoame funcția returnează un vector de polinoame de lungime 2 în care stocăm restul și rezultatul.
  - v. `derivative()`: Algoritmul iterează asupra fiecărei intrări din `TreeMap` și adaugă în polinomul rezultat derivarea fiecărui termen al expresiei.
  - vi. `integral()`: Similar cu algoritmul de derivare acesta iterează peste toată structura de date și adaugă în polinomul rezultat integrala din fiecare termen al polinomului, iar la final returnează rezultatul.
  - vii. `removeZerosAux()`: metodă este una auxiliara care doar facilitează afișarea rezultatului corect, iterează prin structura de date care definește polinomul și șterge toate intrările care au ca și valoare 0.
3. Interfața grafică pentru utilizator: Interfața de utilizator conține 3 text field-uri etichetate corespunzător utilizării lor, aditional avem 6 butoane, unul pentru fiecare operație pe care calculatorul o implementează. Fiecare buton are asociat un `ActionListener` care la apăsarea butonului citește polinoamele din campurile indicate, apelează operația aleasă și afișează rezultatul acesteia în câmpul denumit expresie. Excepție face operația de împărțire care atunci când este apelată apare un text field suplimentar în care se afișează restul operației.



## 5. Rezultate

Pe fiecare dintre operațiile pe polinoame se rulează cate 2 teste și una pentru afisarea polinomului. În cadrul testelor am acoperit 80% din plaja tuturor posibilitatilor. Toate testele ajung la valoarea așteptată ceea ce denota functionalitatea corecta a programului.

✓ OperationTest	1 sec 565 ms
✓ Substraction test 2	447 ms
✓ Derivative test 1	49 ms
✓ Derivative test 2	55 ms
✓ Division test 2	223 ms
✓ Integration test 1	38 ms
✓ Substraction test 1	67 ms
✓ Addition test 1	57 ms
✓ Addition test 2	59 ms
✓ Division test 1	369 ms
✓ Integration test 2	55 ms
✓ Multiplication test 1	65 ms
✓ Multiplicatyon 2	59 ms
✓ toString Method test	22 ms

## 6. Concluzii

1. *adaugarea unei buton de afișare a valorii pe care polinomul o primește într-un anumit punct*
2. *adaugarea unei buton care sa afiseze graficul pentru funcția data de un anumit polinom*
3. *posibilitatea de a face calcule și de a deriva sau integra și funcții precum sin sau cos.*

## 7. Bibliografie

*Exemplu:*

1. <https://www.baeldung.com>