

# **DOCUMENTATIE**

TEMA NUMARUL 1

NUME STUDENT:CIONTE SERGIU IONUT

GRUPA:30228

# CUPRINS

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

## **1.OBIECTIVUL TEMEI**

Obiectivul principal al acestei teme este implementarea unui “calculator” care efectueaza operatiile principale pentru polinoame.

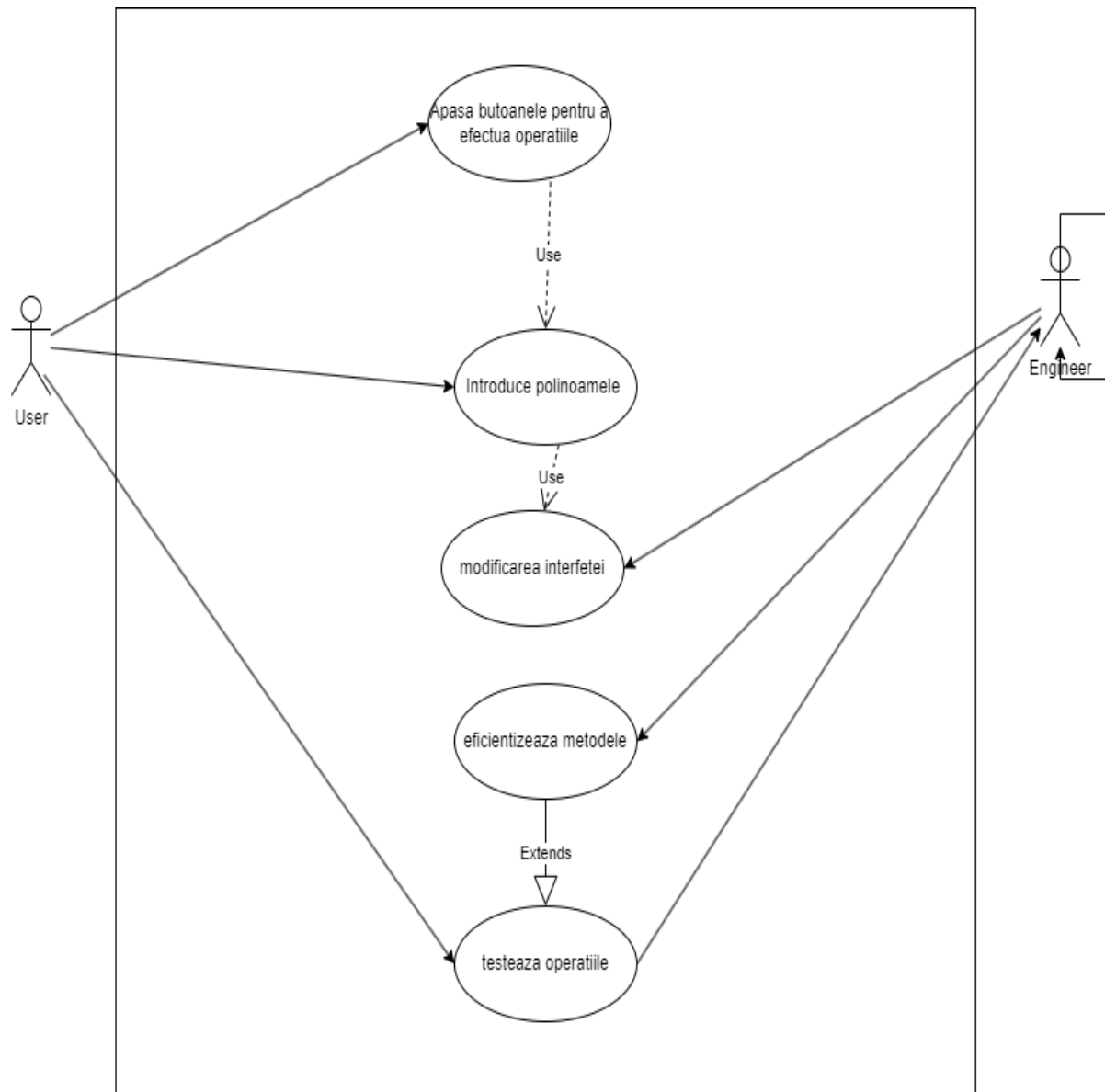
Obiectivele secundare sunt:

<b>1.Reprezentarea polinoamelor folosind structuri de tip Map</b>
<b>2.Implementarea individuala corecta a fiecarei operatii</b>
<b>3.Crearea unei interfete grafice cu utilizatorul</b>
<b>4.Adaugarea de functionalitati elementelor interfetei</b>
<b>5.Afisarea corespunzatoare a rezultatelor dorite</b>
<b>6.Testarea metodelor implementate folosind JUnit</b>

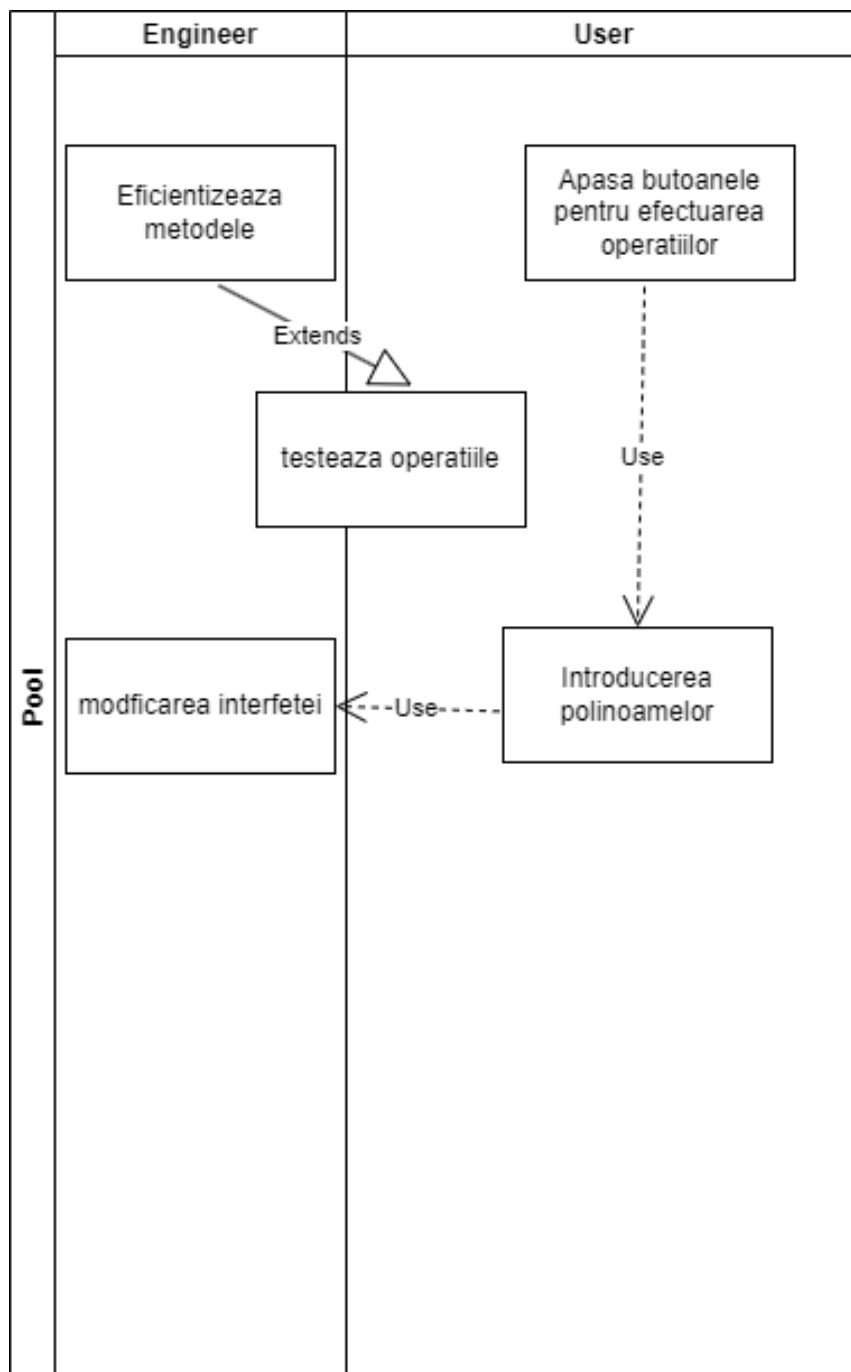
## **2.ANALIZA PROBLEMEI,MODELARE,SCENARII,CAZURI DE UTILIZARE**

<b>CERINTE FUNCTIONALE</b>	<b>CERINTE NONFUNCTIONALE</b>
1.IMPLEMENTAREA CORECTA A OPERATIILOR ELEMENTARE PENTRU POLINOAME	1.AFISAREA UNOR MESAJE CORESPUNZATOARE IN CAZUL INTRODUCERII AMBIGUE A POLINOAMELOR
2.CREAREA UNEI INTERFETE GRAFICE CU UTILIZATORUL	2.TRATAREA CAZURILOR SPECIALE
3.AFISAREA CORESPUNZATOARE A UNUI REZULTAT	3.CUSTOMIZAREA DESIGNULUI INTERFETEI GRAFICE

## DIAGRAMA USE CASE

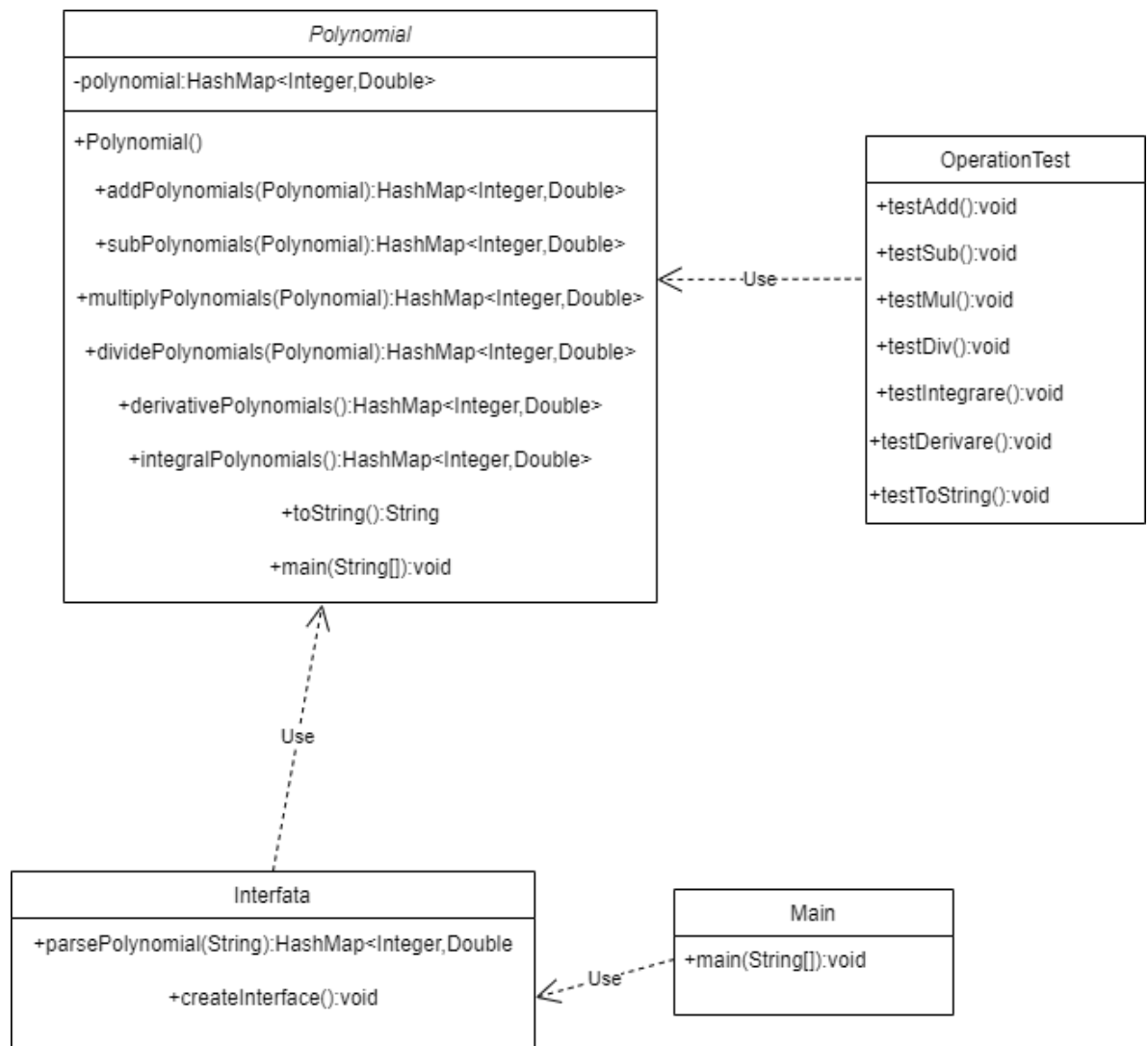


## FLOW CHART CARE DESCRIE USE-CASE-URILE

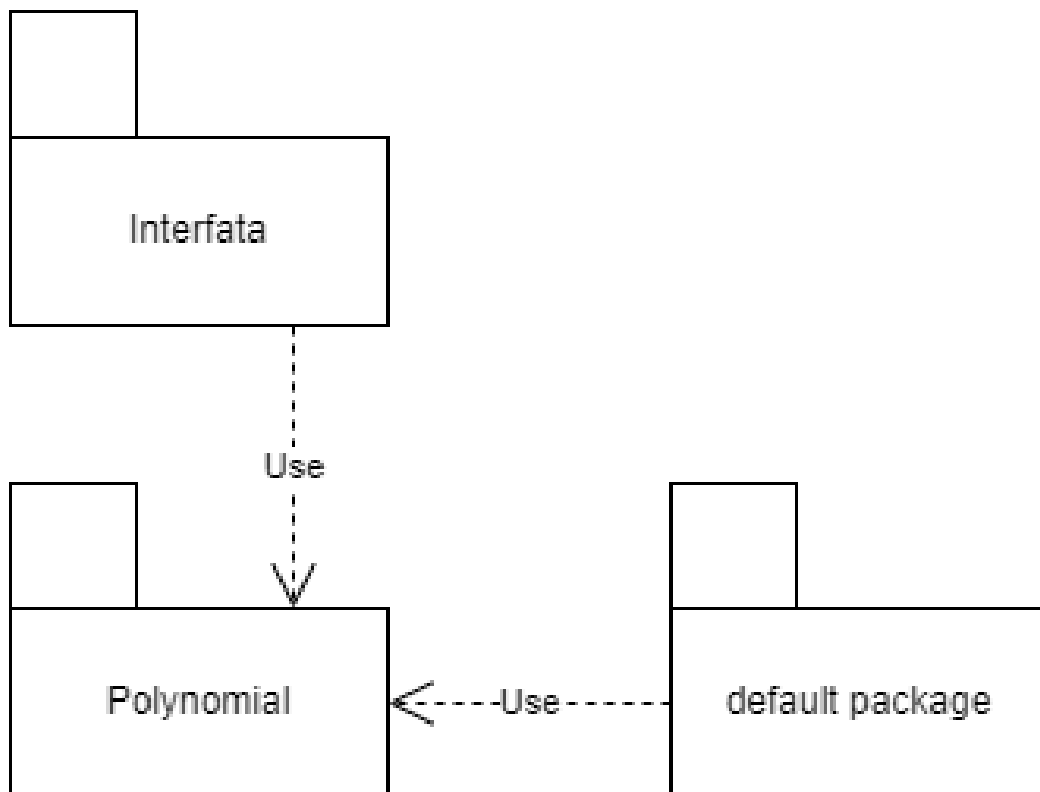


### 3.PROIECTARE

#### DIAGRAMA DE CLASE



## DIAGRAMA DE PACHETE



Structurile de date folosite sunt HashMap pentru crearea polinomului si am ales aceasta structura deoarece operatiile simple au complexitate  $O(1)$ , iar cautarea este de asemenea eficienta si rapida. Ca si algoritmi in principal au fost folositi cei obisnuiti de efectuare a operatiilor pe polinoame. S-a folosit programarea orientata pe obiecte bazata pe clase,obiecte care contin attribute si metode. S-au respecta paradigmele programarii orientate pe obiecte



## **4.IMPLEMENTARE**

S-au implementat 4 clase:

### 1.Clasa Polynomial:

Ca si attribute avem un HashMap cu o cheie de tip integer si valoare de tip double,cheia reprezentand gradul polinomului, iar valoarea coeficientul.

Se defineste un constructor pentru clasa Polynomial;

Dupa aceea in continuare vor fi definite metodele pentru operatiile cu polinoame dupa cum urmeaza:

1.addPolynomials:care va aduna 2 polinoame, adunand fiecare coeficient din primul polinom cu coeficientul gradului corespunzator din al doilea polinom.

2.subPolynomials:care va scadea 2 polinoame analog ca si la adunare doar ca operatia va fi de scadere.

3.multiplyPolynomials:care va inmulti 2 polinoame inmultind fiecare coeficient din primul polinom cu fiecare coeficient din al doilea polinom.

4.dividePolynomials:care va imparti 2 polinoame facand o bucla care va imparti polinomul cu gradul cel mai mare la polinomul cu gradul cel mai mic , impartgind dupa aceea primul monom din P cu primul monom din Q ca sa obtinem catul si dupa aceea inmultim catul cu Q si scadem rezultatul multiplicarii cu P, astfel obtinand restul dupa care repetam procedura.

5.derivativePolynomials:care va deriva un polinom conform regulii de derivare care spune ca  $X^n$  derivat este egal cu  $n \cdot X^{(n-1)}$ .

6.integralPolynomials:care va integra un polinom conform regulii de integrare care spune ca  $X^n$  integrat este egal cu  $X^{(n+1)}/(n+1)$

7.toString care suprascrie metoda toString din clasa Object pentru a converti un polinom (in cazul nostru un HashMap) la un String.

## 2.Clasa Interfata

Aici au fost implementate 2 metode:

Prima se numeste parsePolynomials care parseaza un polinom de la un String

A doua se numeste createInterface care modeleaza interfata, interfata la care am adaugat butoane pentru fiecare metoda in parte respectiv JTextField pentru a putea introduce polinoamele, dupa care mai tarziu am adaugat functionalitati butoanelor prin addActionListener.

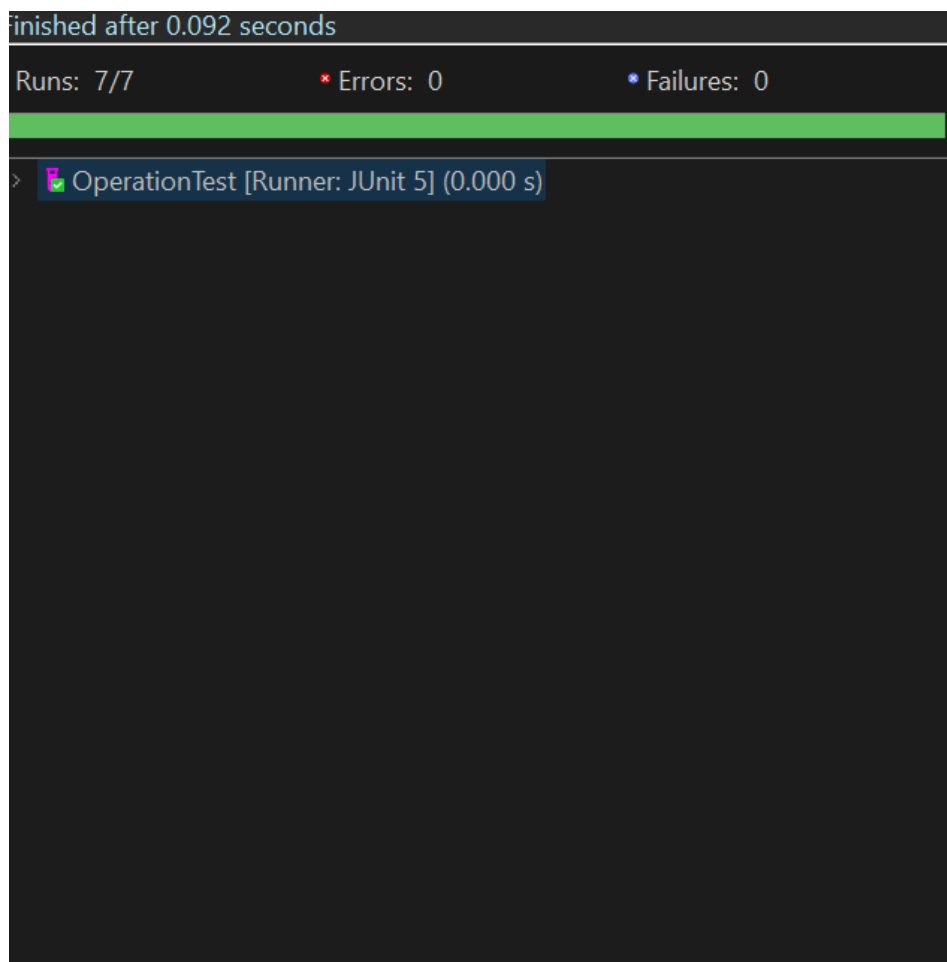
## 3.Clasa Main

Aici practic tot ce facem este sa avem o metoda de tip main care creeaza un obiect de tipul Interfata;

## 5.REZULTATE

### **Clasa Operation Test si testarea**

In clasa Operation Test s-a efectuat testarea cu JUnit pentru fiecare metoda din clasa Polynomials. Testele s-au efectuat utilizand metoda assertEquals comparand rezultatul fiecarei metode din clasa Polynomial cu rezultatul care ar trebui sa fie corect.



## **6.CONCLUZII**

In aceasta tema s-a invatat testarea cu JUnit, precum si lucrarea cu Maven, aprofundarea cunostintelor despre interfete grafice, utilizarea HashMap-urilor precum si dezvoltarea operatiilor pentru polinoame

## **7.BIBLIOGRAFIE**

1.Java HashMap-

[https://www.w3schools.com/java/java\\_hashmap.asp](https://www.w3schools.com/java/java_hashmap.asp)

2.Polynomial Division-

<https://www.geeksforgeeks.org/division-algorithm-for-polynomials/>