

# **DOCUMENTATIE**

## **TEMA 1**

NUME STUDENT: Bichilie Tudor Ovidiu  
GRUPA: 30224

# CUPRINS

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

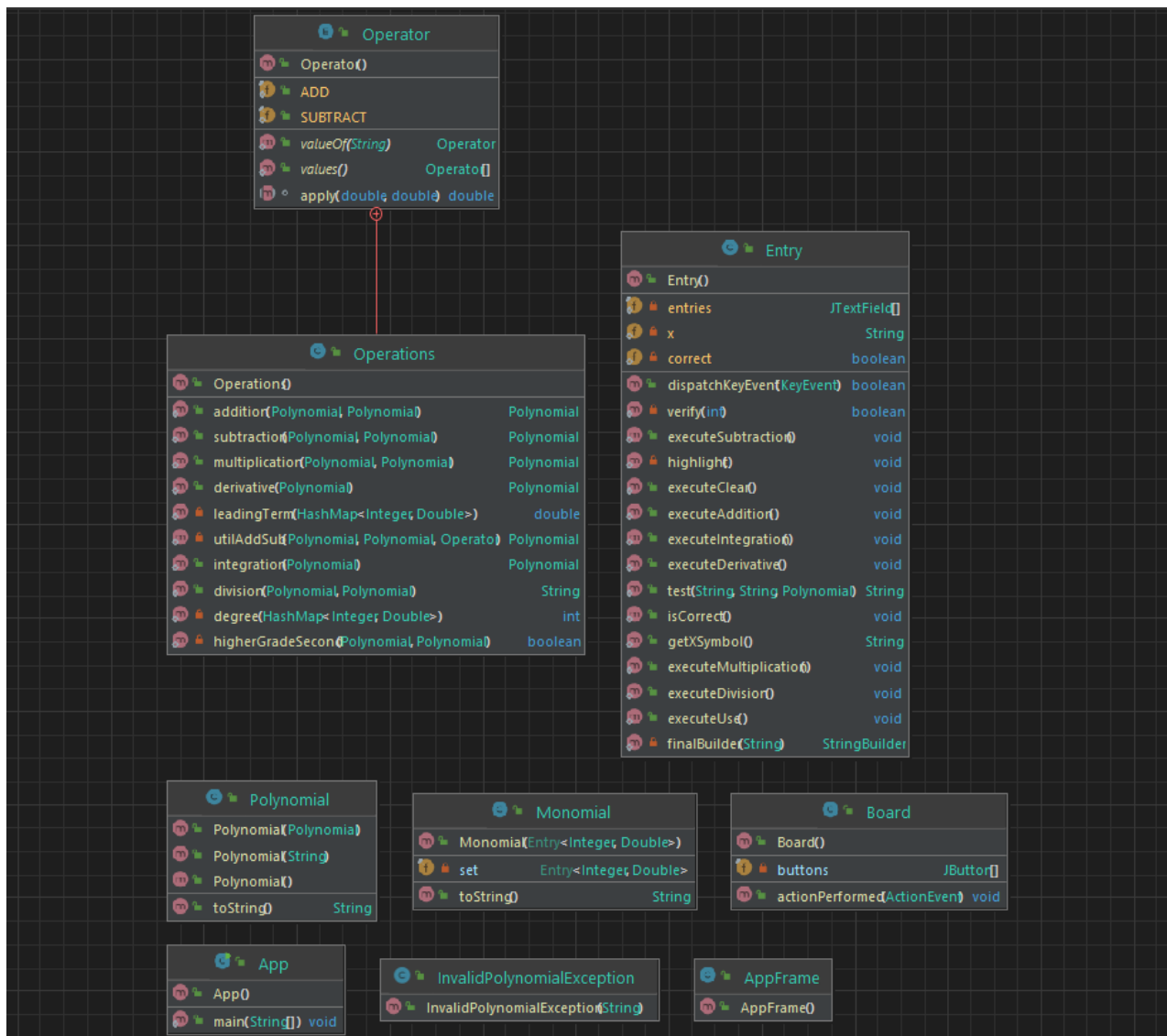
## 1. Obiectivul temei

- Calcul avansat cu polinoame
- Familiarizarea cu instructiuni complexe Java
- Lucrul cu Hash – Maps
- Introducere in Regex si Pattern Matching
- Structurarea organizata a codului

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

- Implementarea operatiilor de calcul
- Gandirea diferita a fiecarora
- Structurarea in Model – View – Controller
- Efectuarea rapida si comoda a operatiilor pe polinoame

## 3. Proiectare



Pentru proiectare s-a mers pe vizualizarea polinomului sub forma de Hash – Map.

Hash – Map este o structura de date Java, care functioneaza similar cu Listele inlantuite doar ca fiecarei valoare introduce in sistem ii corespunde o cheie unica. Rularea din spate este implementata pe Arbore Rosu – Negru pentru a sporii eficienta si facand orice metoda din clasa respectiva de o complexitate maxima de  $O(\log n)$ .

Fiecare Monom reprezinta un Map.Entry asociat structurii, cheile, fiind unice si de tip intreg, au fost asociate exponentilor si valorile, de tipul double, au fost asociate coeficientilor.

S-a mers pe implementare cat mai eficienta a operatiilor, pentru majoritatea fiind strabatute polinoamele maxim o singura data.

Impartirea Polinoamelor reprezinta un caz mai special, aceasta avand nevoie de metode afiliate pentru a executa diferiti pasi. Trebuie facute verificari cum ar fi daca gradul polinomului impartitor este mai mic decat gradul celui deimpartit, etc. Ulterior aceasta extrage gradul maxim din fiecare polinom, creandu si un rest incomplet, executand scaderea dintre valorile obtinute si repetand cat timp gradul maxim al primului este mai mare decat celui de al doilea, in “util” salvandu se restul final.

## 4. Implementare

Interfata de utilizator consta in 3 Clase: AppFrame, Board si Entry.

- AppFrame este cea in care se face organizarea intre panelurile obtinute din celelalte doua. Este stabilita organizarea si forma aplicatiei.
- In Board este vorba de executarea operatiilor. Aceasta clasa are in subordine butoanele ce realizeaza legatura intre operatii si afisarea lor. Butoanele sunt organizate in 2 grupuri pentru estetica.
- Iar in Entry sunt introduse datele de la utilizator si apoi trecute prin filtrul de corectitudine. Am introdus diferite metode de Pattern Matching si Highlight pentru a asigura buna functionare si atentionare a utilizatorului astfel incat programul sa functioneze fara probleme.
- Cea mai importanta clasa a programului este cea Polynomial. In ea se executa pasii de recunoastere si salvare a monoamelor, de aceea este vital ca formatul String al intrarii sa aibe o forma optima si usor de inteles de catre program. Am introdus metode speciale pentru a putea recunoaste monoamele si cu sintaxa minima pentru a usura utilizarea.

## 5. Rezultate

Testarea unitara realizeaza cu un test correct si unul invalid pe fiecare operatie in parte.

Se introduce cate un exemplu sub forma de polinom, se efectueaza operatia corespunzatoare, se trece prin filtrul de verificare al Entry ului si apoi se compara rezultatele.

Testele invalide constau in introducerea unui exemplu in care programul sare in exceptia de InvalidPolynomialException, caz in care la testare metoda va returna un String “invalid”.

Toate testele au avut rezultat correct si au trecut cu success.

## 6. Concluzii

Am aprofundat:

- Junit testing
  - Regex, Pattern Matching
  - Highlight options in JEntry
  - Imbinarea claselor cu metode statice
- 

Pe viitor primul lucru care poate fii dezvoltat este calitatea interfete de utilizator, acordandu l un design mai prietenos si usor de inteles.

Se mai poate lucra la usurinta introducerii unui polinom si la cazurile acceptate.

## 7. Bibliografie

- *Testarea unitara cu Junit:*  
<https://junit.org/junit5/docs/current/api/org.junit.jupiter.api/org/junit/jupiter/api/Assertions.html>
- *Creearea Regexului:*  
<https://regexr.com/>
- *Highlight pe monoamele introduse gresit:*  
<https://stackoverflow.com/questions/6530105/highlighting-text-in-java>
- *Sprijin pentru operatiile aritmetice pe polinoame:*  
[https://dsrl.eu/courses/pt/materials/PT2023\\_A1\\_S1.pdf](https://dsrl.eu/courses/pt/materials/PT2023_A1_S1.pdf)