DOCUMENTATIE

TEMA 1

CURCANU SAMUEL-ADRIAN

GRUPA:30227

# 

# CUPRINS

[1. Obiectivul temei 3](file:///C:\Users\Samuel-Adrian\Downloads\PT2023_Template_Documentatie_RO%20(1).doc#_Toc95297885)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](file:///C:\Users\Samuel-Adrian\Downloads\PT2023_Template_Documentatie_RO%20(1).doc#_Toc95297886)

[3. Proiectare 3](file:///C:\Users\Samuel-Adrian\Downloads\PT2023_Template_Documentatie_RO%20(1).doc#_Toc95297887)

4. Implementare…………………...……………………………………………………………………………..4

5. Rezultate………………………………………………………………………………………………………5

6. Concluzii……………………………………………………………………………………………………...5

7. Bibliografie………………………………………………………………………………………………...…5

**1.Obiectivul temei**

Tema are ca scop principal creearea unui calculator care poate sa faca anumite operatii pe baza unor polinoame.

• Utilizarea un design de programare orientat pe obiecte (detaliata in punctul 3)

• Utilizare Map pentru modelarea polinomului.(detaliata in punctul 4)

• Implementare interfață grafică cu utilizatorul utilizând Java Swing sau JavaFX (detaliata in punctul 4)

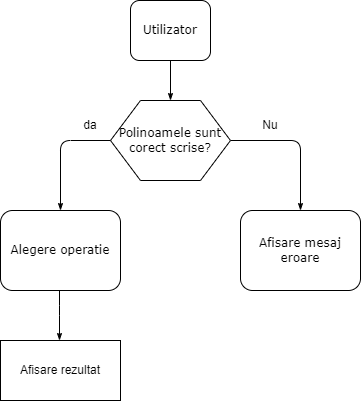
• Implementare operațiile (detaliata in punctul 4)

• Utilizare Junit pentru testare (detaliata in punctul 5)

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

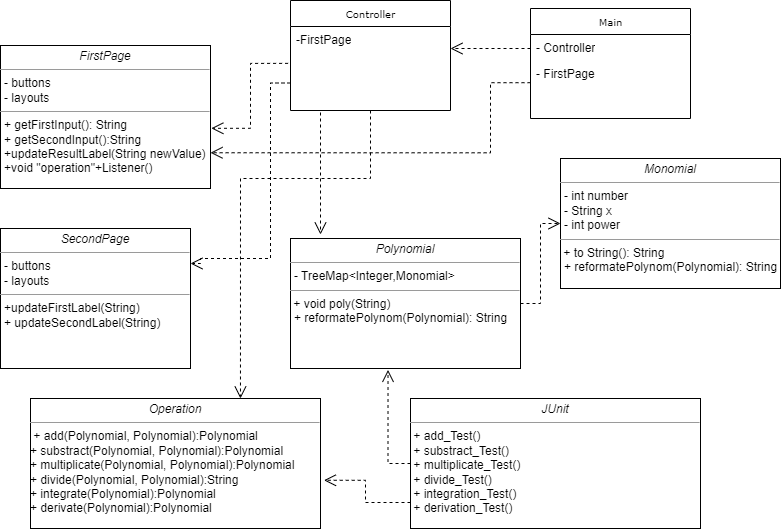
Pentru functionarea corecta a fiecarei operatii a trebuit sa tin cont de mai multe lucruri:

forma polinomului primit ca si string sa fie corecta (coeficientul, un string format dintr-un singur caracter “x”, si puterea lui x, ex: 2x3 se citeste 2x la puterea 3), la anumite operatii trebuie sa verificam daca al doilea polinom este 0 sau daca este null etc.



**3.Proiectare**

Tema este creeata in stil OOP, variabilele sunt toate private putand fi accesate doar prin metode de get si set. Pentru o vizualizare cat mai buna clasele sunt in pachete diferite avand un nume corespunzator, pentru a ajuta utilizatorul sa gaseasca portiuni de cod dorite cat mai usor.



**4.Implementare**

In clasa “Monomial” avem 3 variabile private : number, power si x; pentru care avem si get-er si set-er. De asemenea in clasa aceasta avem si o metoda “toString” care returneaza o iesire cat mai frumoasa a unui polinom sub forma de string.

In clasa “Polynomial” avem o singura variabila care este un TreeMap avand ca si key un intreg si ca si valoare un Monomial. Metoda “poly” primeste ca paramentru de intrare un string pe care trebuie sa il imparta in monoame si sa le adauge in TreeMap. Metoda parcurge string-ul si in functie de pozitia semenelor “+” si “-“ creeaza substringuri pe care le stocheaza intr-un tablou. Apoi parcurge tabloul de substringuri si compara fiecare substring cu pattern-urile corespunzatoare. Creeaza apoi Monoame in functie de caracterele fiecarui substring; le adauga apoi in TreeMap.

Clasa “Polynomial” mai are si o functie specifica “reformatePolynom” care primeste ca parametru de intrare un “Polynomial” si returneaza un string, intr-o forma cat mai lizibila a polinomului(de la puterea cea mai mare la cea mai mica).

In clasa “FirstPage” implementam interfata paginii principale a programului , pe interfata avem 6 butoane care corespund fiecarei operatii, doua campuri de text in care putem introduce cele doua polinoame si un camp pentru a afisa rezultatul operatiei dorite. Aceasta clasa are metode pentru a extrage textul din cele doua campuri si metode pentru a da update la campul de result, de asemenea am adaugat metode pentru ascultatori pentru fiecare buton.

In clasa “SecondPage” avem doua campuri care isi vor da update de fiecare data la apasarea butonului corespunzator. Aceasta clasa am folosit-o pentru doua operatii: derivare si integrare, pentru a afisa amandoua polinoame derivate sau integrate.

In clasa “Operation” sunt prezente toate operatiile pe care le poate face calculatorul.

Fiecare operatie are metoda ei separata in care este implementata.

Clasa “Controller” primeste ca variabila un atribut de tipul “FirstPage” si uneste toate componentele programului, de la operatii la ascultatori de butoane si toate cele necesare pentru functionarea fiecarei optiuni a programului.

Clasa “JUnit” contine testele fiecarei operatii in parte, fiecare Test preia niste date pentru cele doua polinoame din niste fisiere text si compara rezultatul operatiei respective cu rezultatul asteptat.

In final clasa “Main” creaza un obiect de tip “FirstPage” pe care il da mai departe unui constructor al unui obiect de tip “Controller” si seteaza visibilitate paginii principale.

**5.Rezultate**

Fiecare operatie preia polinomul 1 dintr-un fisier text numit “Polinom1”, polinomul 2 dintr-un fisier text numit “Polinom2” si apoi face operatia respectiva si o compara cu rezultatul asteptat care este gasit tot intr-un fisier text respectiv fiecarei operatie(ex: “Add.txt” , “Substract.txt” etc).

Fiecare operatie va efectua la randul ei 6 operatii de acelasi tip cu date de intrare diferite si cu rezultate corespunzatoare fiecarei dintre acestea.

**6.Concluzii**

Tema a fost una destul de complexa punand la testare cunostintele de java, de asemenea aceasta tema se mai poate dezvolta ulterior pentru a o face mult mai complexa, cum ar fi ridicari la putere, limite etc.

**7.Bibliografie**

1. FUNDAMENTAL PROGRAMMING TECHNIQUES LABORATORY GUIDE - <https://dsrl.eu/courses/pt/>

2.Regerx -https://regexr.com/