

DOCUMENTAȚIE

TEMA 1 – Calculator Polinomial

NUME STUDENT: **BÎZGĂ ANA-MARIA**
GRUPA: **30221**

CUPRINS

1. Obiectivul temei	3
2. Analiza problemei, modelare, scenarii, cazuri de utilizare.....	3
3. Proiectare	5
4. Implementare	7
5. Rezultate	7
6. Concluzii.....	8
7. Bibliografie	8

1. Obiectivul temei

- (i) **Obiectivul principal al temei** : implementarea unei aplicații Java pentru manipularea și operarea cu polinoame.
- (ii) **Obiective secundare** :
 - proiectarea aplicației folosind principii ale programării orientate pe obiecte (capitolul 3)
 - implementarea operațiilor de bază pentru polinoame (capitolul 4)
 - testarea și validarea corectitudinii funcțiilor implementate (capitolul 5)
 - proiectarea interfeței grafice (capitolul 4)

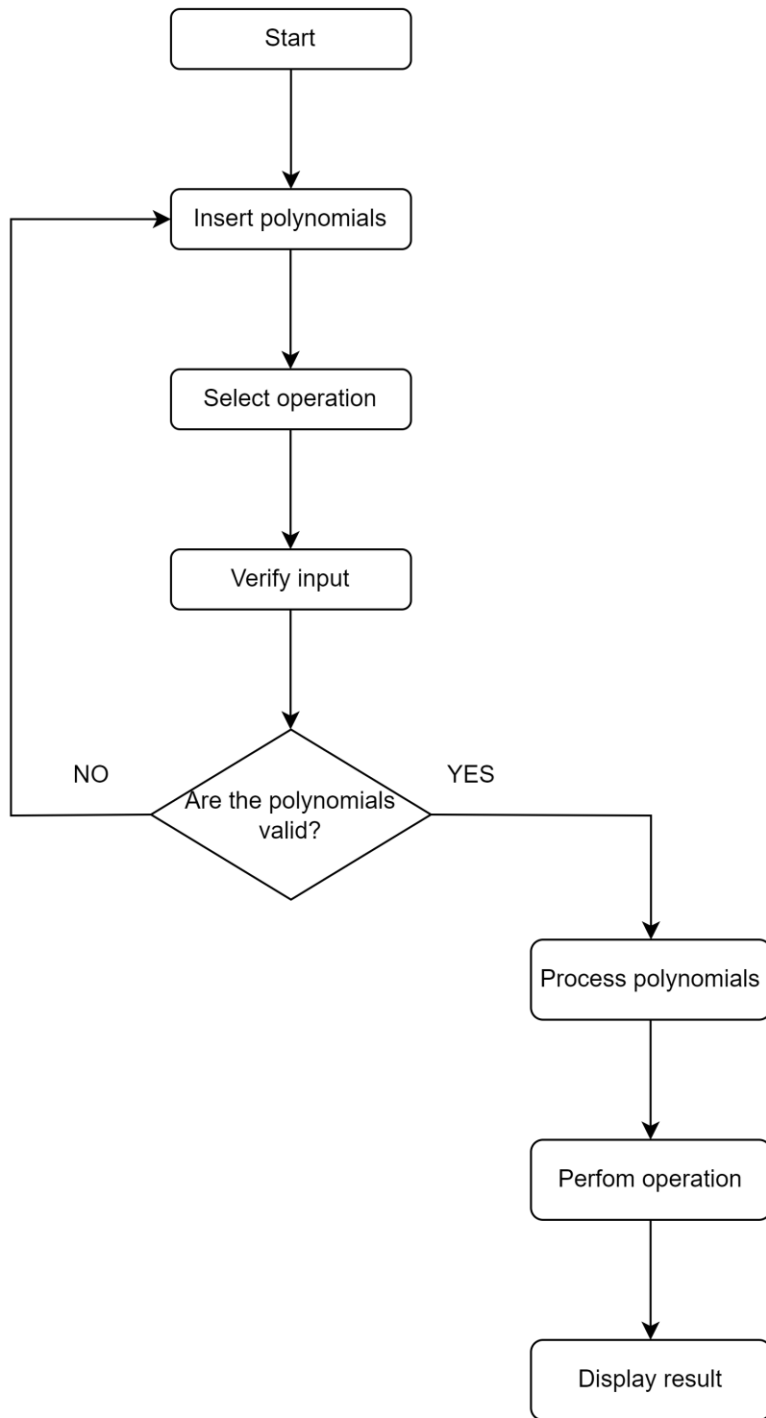
2. Analiza problemei, modelare, scenarii, cazuri de utilizare

Cadrul de cerințe funcționale și non-funcționale :

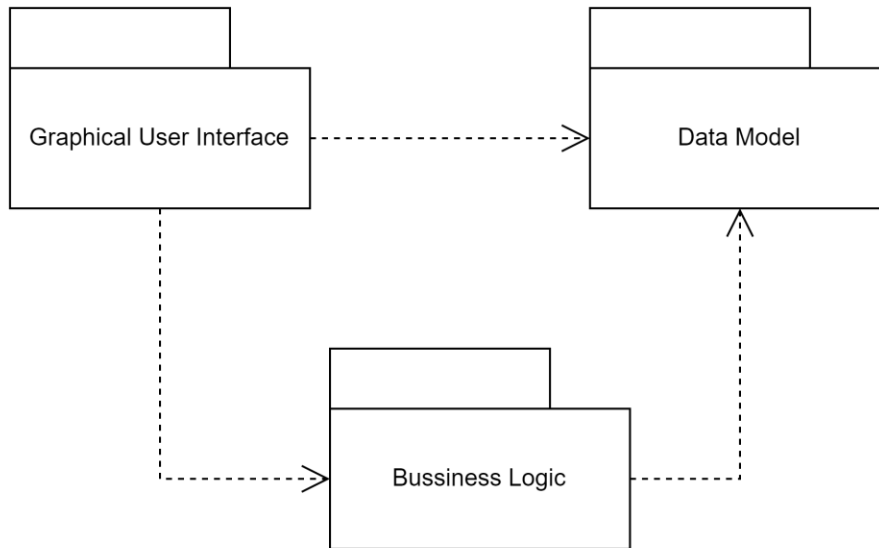
- **Funcționale** : se permite utilizatorului să efectueze operații aritmetice pe polinoame (adunare, scădere, înmulțire, împărțire) , inclusiv operații avansate precum derivarea și integrarea, gestionarea anumitor excepții ce pot apărea pe parcurs și testarea anumitor scenarii.
- **Non-funcționale** : aplicația nu este eficientă în execuție.

Cazuri de utilizare :

- **Operarea cu polinoame** : utilizatorul introduce două polinoame, alege operația dorită și se va afișa rezultatul.
- **Operarea cu polinoame nule** : se verifică corectitudinea operațiilor când unul sau ambele polinoame sunt nule. Aplicația va atenționa utilizatorul în această situație.



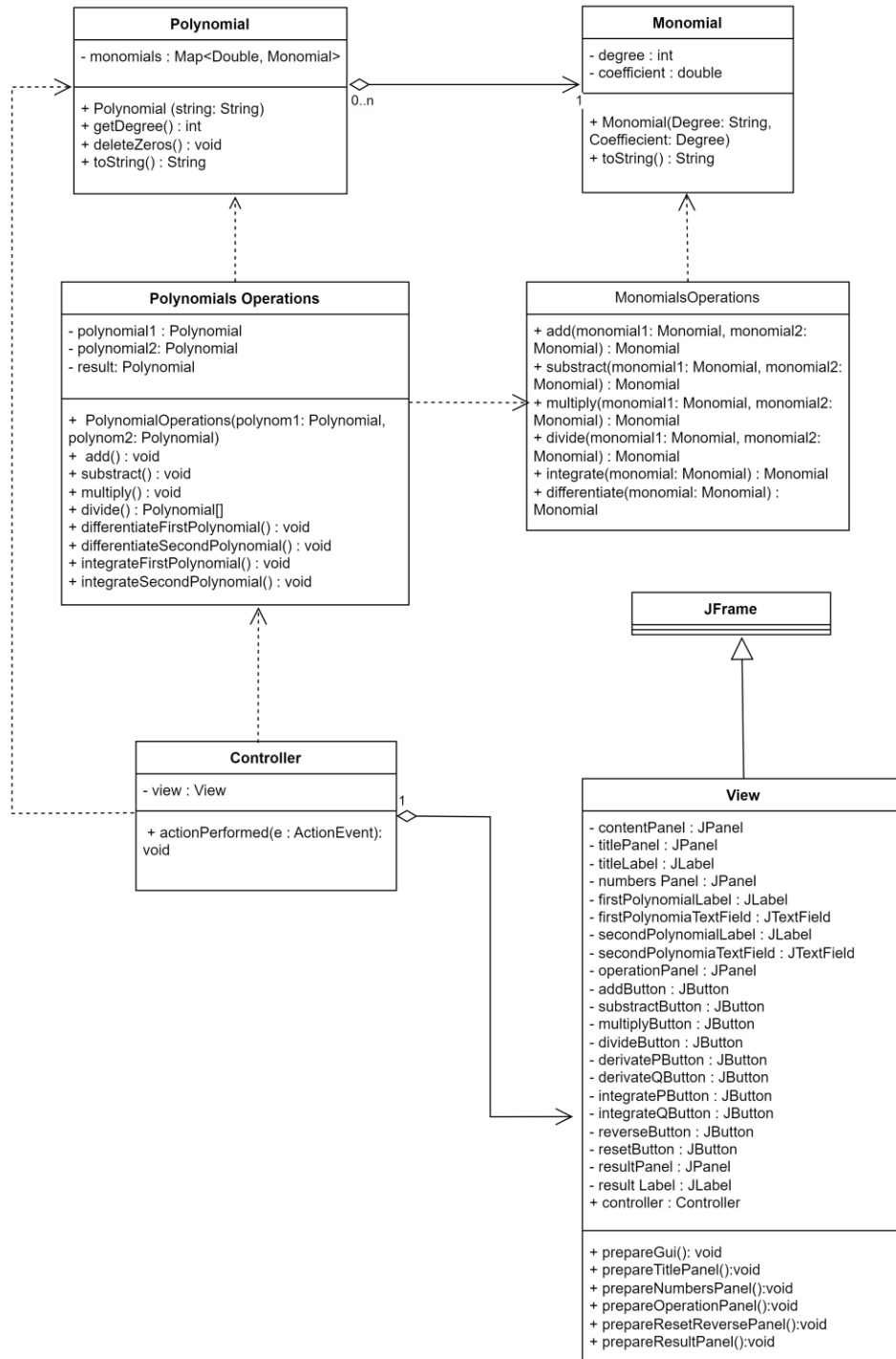
3. Proiectare



Graphical User Interface : View, Controller, Main

Data Models : Polynomial, Monomial

Bussiness Logic : PolynomialsOperation, MonomialsOperation



4. Implementare

Clasa Monomial reprezintă un monom ($a \cdot X^b$), conține gradul (b) și coeficientul (a) acestuia. Conține constructori și metoda toString.

Clasa Polynomial reprezintă un polinom și conține o colecție (Map) de monoame, acestea fiind mapate după grad. Conține constructori (inclusiv un constructor în care se face extragerea polinomului dintr-un String), o metodă pentru determinarea gradului unui polinom, o metodă pentru eliminarea termenilor cu coeficientul 0 și metoda toString.

Clasa MonomialsOperations conține doar metode statice pentru fiecare dintre cele 6 operații implementate.

Clasa PolynomialsOperations are trei polinoame : două input-uri și rezultatul. Conține implementări pentru cele 6 operații.

Interfața utilizator a fost implementată folosind Java Swing. Este compusă dintr-o fereastră principală ce conține mai multe panel-uri organizate cu FlowLayout. Conține două câmpuri de text pentru introducerea polinoamelor și butoane corespunzătoare fiecărei operații implementate. De asemenea, conține și un buton „Reset” pentru resetarea câmpurilor și a rezultatului, dar și un buton pentru interschimbarea input-urilor. Panel-urile folosesc în cea mai mare parte un layout de tip GridLayout pentru organizarea butoanelor, câmpurilor de text sau a etichetelor.

Clasa Controller este cea care gestionează interacțiunea utilizatorului cu aplicația. Primește evenimente din partea butoanelor și în funcție de butonul apăsat se vor prelua input-urile și se va realiza operația. Verifică, de asemenea, și datele introduse, afișând mesaje de eroare dacă este cazul.

5. Rezultate

Au fost create teste pentru verificarea metodelor din clasa PolynomialsOperations, dar și a extragerii unui polinom dintr-un String. Scenariile de testare sunt similare pentru toate operațiile : două polinoame valide, un polinom zero, un polinom sau ambele nule. Se testează și cazul împărțirii cu 0. Pentru fiecare scenariu se așteaptă ca rezultatul obținut să coincidă cu cel așteptat, iar în cazul unei erori aruncate, testul va detecta această eroare.

testDeleteZeros()	passed	34 ms
testNullPolynomial()	passed	2 ms
testInvalidFormat()	passed	1 ms
testValidFormat()	passed	4 ms

TestNullPolynomialAddition()	passed	23 ms
TestZeroMultiplication()	passed	4 ms
TestZeroDivision()	passed	1 ms
TestValidDifferentiate()	passed	10 ms
TestNullSubstraction()	passed	1 ms
TestZeroSubstraction()	passed	10 ms
TestDivisionbyZero()	passed	1 ms
TestNullIntegrate()	passed	1 ms
TestZeroDifferentiate()	passed	
TestZeroIntegrate()	passed	1 ms
TestValidAddition()	passed	
TestAdditionZero()	passed	1 ms
TestValidIntegrate()	passed	2 ms
TestZeroSubstraction2()	passed	1 ms
TestNullMultiplication()	passed	1 ms
TestNullDivision()	passed	2 ms
TestValidMultiplication()	passed	1 ms
TestValidSubstraction()	passed	1 ms
TestValidDivision()	passed	2 ms
TestNullDifferentiate()	passed	

6. Concluzii

În urma acestei teme, s-au consolidat cunoștințe semnificative în înțelegerea și aplicarea conceptelor de Programare Orientate pe Obiecte.

Aplicația poate fi îmbunătățită prin optimizare și adăugare de noi funcționalități (de exemplu calcularea rădăcinilor unui polinom).

7. Bibliografie

1. https://dsrl.eu/courses/pt/materials/PT_2024_A1_S1.pdf
2. https://dsrl.eu/courses/pt/materials/PT_2024_A1_S2.pdf
3. <https://gitlab.com/dsrl-pt/pt-gui-demo-swing>
4. https://dsrl.eu/courses/pt/materials/PT_2024_A1_S3.pdf
5. <https://regexr.com/>