

DOCUMENTATIE

TEMA 1

NUME STUDENT: Rujac Roxana
GRUPA: 30224

Cuprins:

1. Obiectivul temei	3
1.1 Obiectivul principal	3
1.2 Obiective secundare	3
2. Analiza problemei, modelare, scenarii, cazuri de utilizare	3
2.1 Cerinte functionale	3
2.2 Cerinte non-functionale	4
3. Proiectare	4
3.1 Proiectarea OOP	4
3.2 Diagrame UML	5
3.3 Structuri de date folosite	7
3.4 Algoritmi folositi	7
4. Implementare	7
4.1 Implementarea claselor	7
4.2 Implementarea Interfetei-Utilizator (GUI)	9
5. Rezultate	13
6. Concluzii	14
7. Bibliografie	14

1. Obiectivul temei

1.1 Obiectivul principal

Obiectivul principal al acestei teme este implementarea unui calculator polinomial in Java ce poate realiza diverse operatii: adunare, scadere, inmultire, impartire, derivare si integrare.

1.2 Obiective secundare

- proiectarea interfetei utilizator (UI)
- validarea si parsarea cu pattern recognition a unor intrari
- implementarea unor operatii de baza cu polinoame
- testarea calculatorului

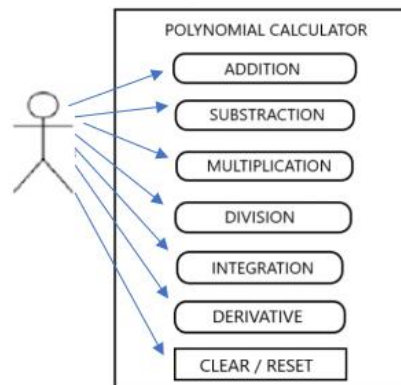
2. Analiza problemei, modelare, scenarii, cazuri de utilizare

2.1 Cerinte functionale

2.1.1 Clasificarea MoSCoW

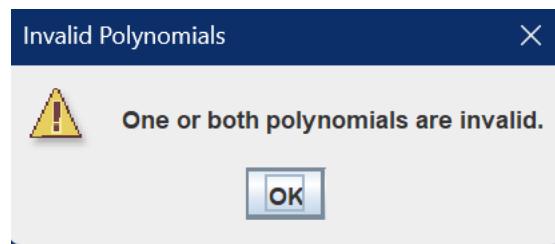
- Must have – operatiile de adunare, scadere, inmultire, impartire, derivare, integrare
- Should have – verificari in caz de eroare (ex:introducerea unui string ce nu reprezinta un polinom valid)
- Could have - calcularea radacinilor uni polinom
- Won't have - memorarea operatiilor si rezultatelor anterioare

2.1.2 Diagrama use-case



2.1.3 Scenarii de use-case

- Dacă utilizatorul introduce de la tastatură polinoame valide va putea alege să efectueze una din operațiile:
 - Adunare
 - Scadere
 - Înmulțire
 - Împartire
 - Derivare (se va deriva primul polinom introdus)
 - Integrare (se va integra primul polinom introdus)
- Utilizatorul poate șterge atât conținutul introdus de el în text-fields (cele două polinoame) cât și rezultatul obținut apăsând butonul de clear -“c”- din tastatură aflată în partea de jos a ferestrei
- În cazul în care utilizatorul introduce unul sau două polinoame invalide se va deschide o fereastră de avertizare cu mesajul “One or both polynomials are invalid”



2.2 Cerințe non-funcționale

- Programul este ușor de utilizat și intuitiv
- Design simplu și compact
- Oferă un timp de răspuns rapid pentru operațiile matematice
- Funcționează fără a genera erori sau situații excepționale

3. Proiectare

3.1 Proiectarea OOP

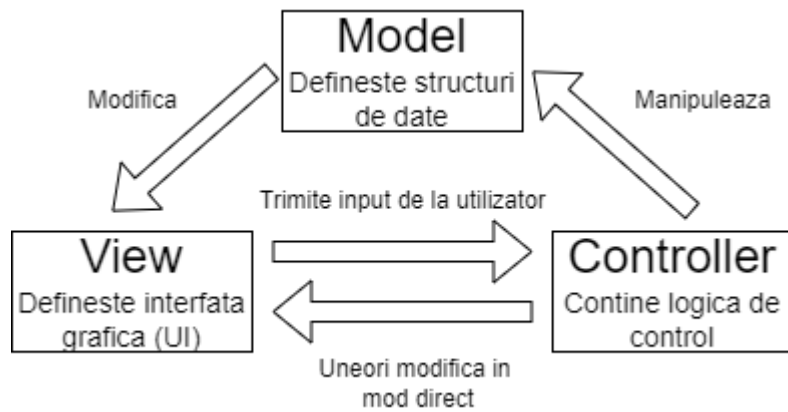
Codul respectă principiile OOP:

- încapsularea
- abstractizare
- moștenire
- polimorfism

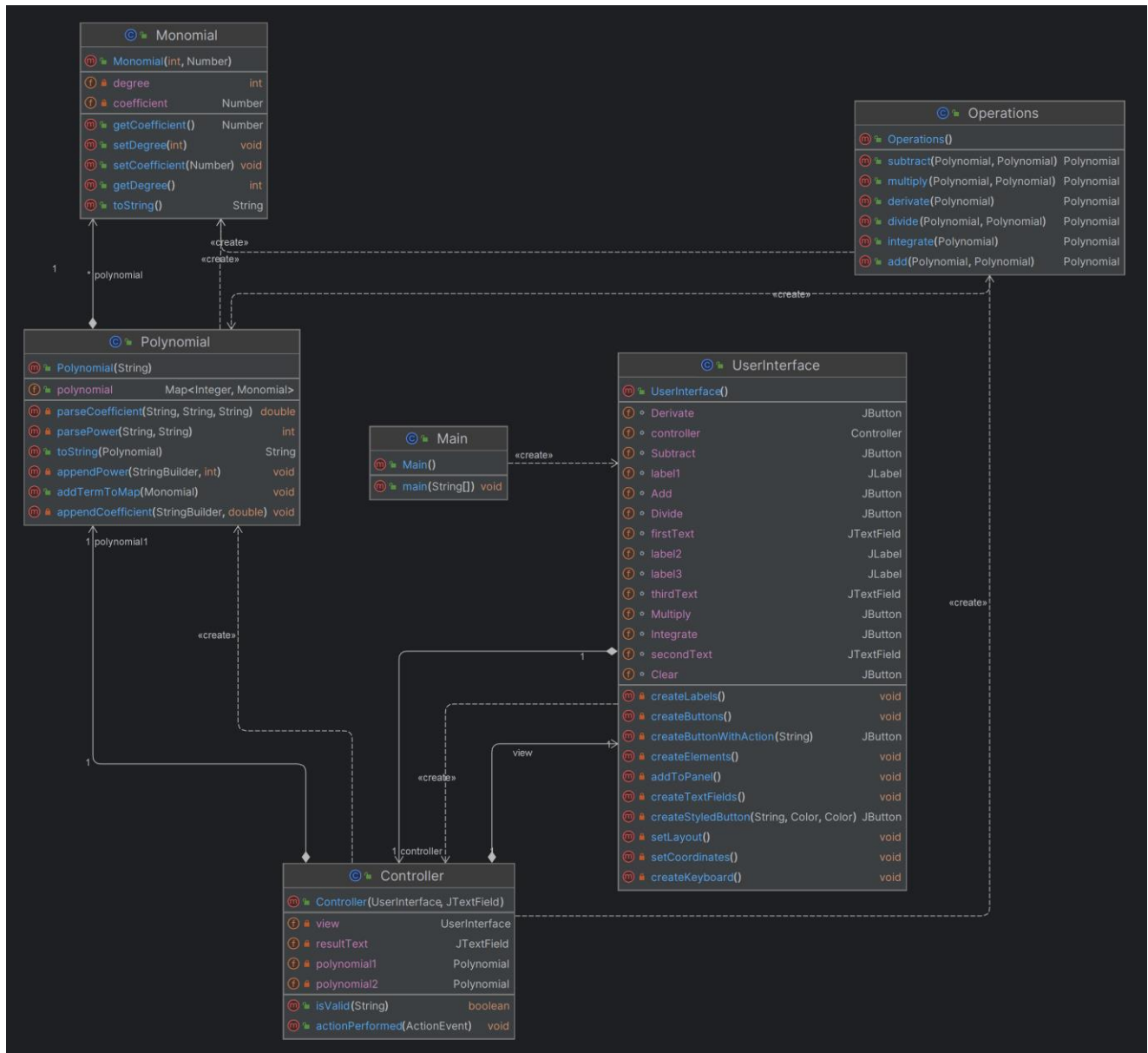
3.2 Diagrame UML

3.2.1 Diagrama UML a pachetelor

Am folosit o arhitectura de tip MVC (Model-View-Controller) pentru pachetele programului.



3.2.2 Diagrama UML a claselor



3.3 Structuri de date folosite

1. TreeMap

Această structură este utilizată pentru a reprezenta polinoamele în clasele Polynomial și Operations. Ea stochează monomiile polinomului, asociind fiecărui grad al monomului un obiect de tip Monomial. În cadrul acestei structuri de date, gradul monomului servește ca cheie, iar obiectul Monomial asociat este valoarea corespunzătoare.

2. Monomial

Această clasă reprezintă un monom dintr-un polinom. Atributele sale sunt gradul și coeficientul monomului. Utilizarea unei clase separate pentru a reprezenta monomul permite encapsularea datelor și definirea unor metode specifice pentru manipularea monomului.

3.4 Algoritmi folosiți

1. Pattern Recognition cu Regex
2. Impartirea polinoamelor – long division

4. Implementare

4.1 Implementarea claselor

Clasele sunt organizate în 3 pachete, conform arhitecturii MVC. Pachetul “Model” conține clasele “Monomial” și “Polynomial”. Pachetul “View” conține clasele “UserInterface” și “Controller”, iar pachetul “Controller” conține clasa “Operations”.

1. Clasa Main

Acesta este fisierul Main al aplicației, care se ocupă de pornirea și afișarea interfeței utilizator pentru calculatorul polinomial.

2. Clasa Monomial

Clasa Monomial reprezintă un termen monomial din algebra polinomială și este utilizată pentru a stoca și manipula informații despre un singur termen dintr-un polinom. Ca și atribute avem “degree” și “coefficient”, reprezentând gradul și respectiv coeficientul unui monom ($4x^3 \Rightarrow \text{degree} = 3, \text{coefficient} = 4$).

3. Clasa Polynomial

Clasa Polynomial reprezinta un polinom si contine un Map de monoame. Oferă funcționalități pentru crearea, manipularea și afișarea polinoamelor într-o formă corespunzătoare.

```
public Polynomial(String input) {
    //descending order of power
    this.polynomial= new TreeMap<>((o1, o2) -> Integer.compare(o2, o1));

    Pattern termPattern = Pattern.compile(regex: "[+\\-]?(\\d*)(x(?:\\^(\\d+))?)?");
    Matcher matcher = termPattern.matcher(input);

    //extracting components
    while (matcher.find()) {
        String signStr = matcher.group(1);           //for the sign - optional ([+\\-]?)
        String coefficientStr = matcher.group(2);    //the coefficient of the term - optional (\\d*)
        String hasX = matcher.group(3);             //contains x or not - optional (x(?:\\^(\\d+))?)
        //subgroup 4
        String powerStr = matcher.group(5);          //has an exponent (\\^(\\d+))
        //if x has an exponent or its 1 (\\d+)

        double coefficient = parseCoefficient(coefficientStr, hasX, signStr);
        int power = parsePower(hasX, powerStr);
        Monomial monomial = new Monomial(power, coefficient);
        addTermToMap(monomial);
    }
}
```

4. Clasa Operations

Clasa Operations contine metode pentru efectuarea operatiilor matematice pe polinoame: adunare, scadere, inmultire, impartire, derivare și integrare.

- Adunare : aduna doua polinoame prin adunarea coeficientilor monoamelor cu acelasi exponent; monoamele care nu au niciun exponent comun in cele doua polinoame se adauga asa cum sunt la rezultat
- Scadere : scade doua polinoame prin scaderea coeficientilor monoamelor cu acelasi exponent; monoamele care nu au niciun exponent comun in cele doua polinoame se adauga asa cum sunt la rezultat
- Inmultire : fiecare monom din primul polinom este înmulțit cu fiecare monom din al doilea polinom
- Impartire : se foloseste algoritmul de impartire a polinoamelor
- Derivare : toti coeficientii sunt inmultiti cu exponentul lor respectiv și se scade 1 din exponent
- Integrare : toti coeficientii sunt impartiti la (exponent + 1) si apoi se adauga 1 la exponent

Exemplu implementarea operatiei de inmultire:

```
3 usages
public Polynomial multiply(Polynomial p1, Polynomial p2){
    Polynomial result = new Polynomial( input: "");
    for (Map.Entry<Integer, Monomial> entry : p1.polynomial.entrySet()) {
        for (Map.Entry<Integer, Monomial> entry2 : p2.polynomial.entrySet()){
            Monomial monomial1 = entry.getValue();
            Monomial monomial2 = entry2.getValue();
            int degree = monomial1.getDegree() + monomial2.getDegree();
            double coefficient = (double) monomial1.getCoefficient() * (double) monomial2.getCoefficient();

            if(coefficient !=0)
                result.addTermToMap(new Monomial(degree, coefficient));
        }
    }

    return result;
}
```

4.2 Implementarea Interfetei-Utilizator (GUI)

4.2.1 Prezentare generala

Interfata grafice este una simpla si intuitiva. In cele doua casete text utilizatorul poate introduce unul sau doua polinoame (in cazul in care acestea nu reprezinta siruri de caractere valide, programul va genera o eroare) si sa selecteze operatia matematica pe care doreste sa o efectueze. De asemenea, exista si un buton de stergere, butonul "c" din tastatura din partea inferioara a ferestrei, care sterge atat polinoamele introduse, cat si rezultatul obtinut.



Polynomial Calculator

first polynomial =

second polynomial =

result =

add

multiply

derivate

subtract

divide

integrate

1

2

3

0

.

,

4

5

6

+

-

*

7

8

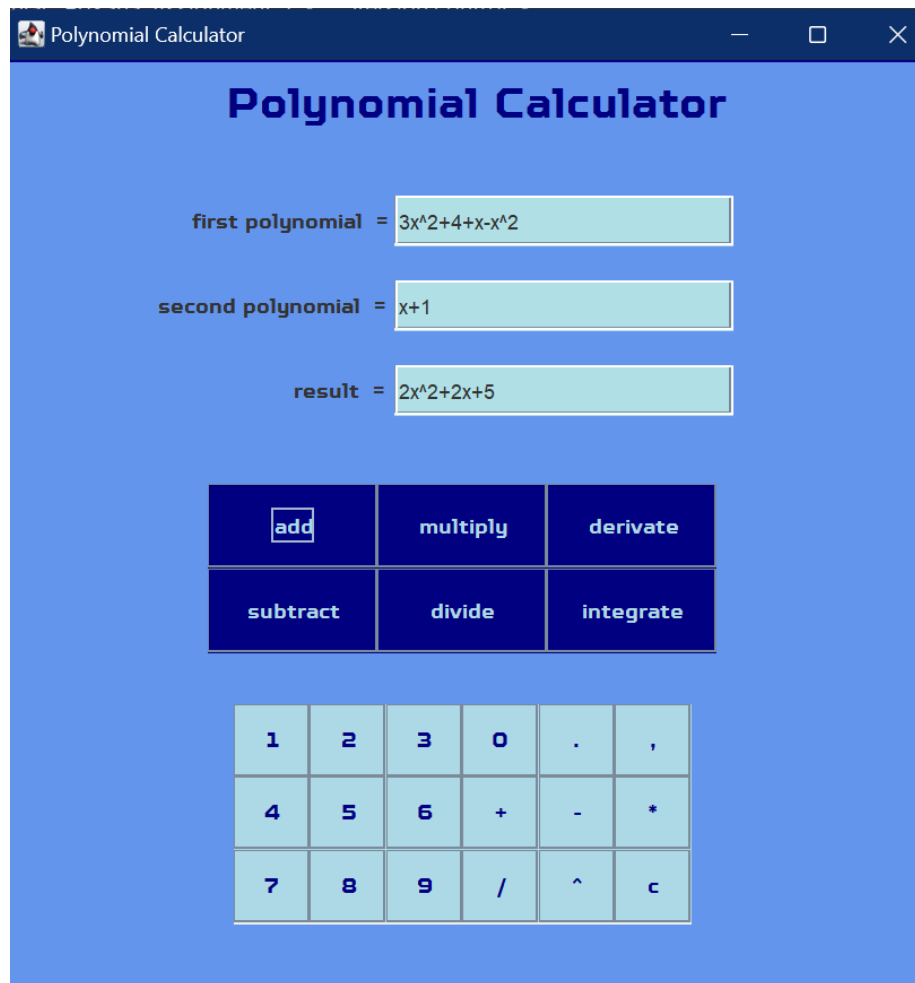
9

/

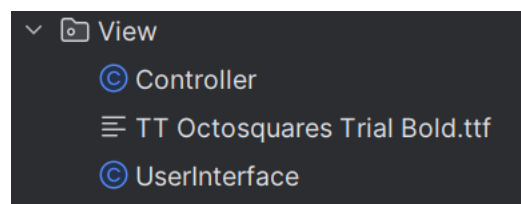
^

c

Exemplu pentru operatia de adunare:



4.2.2 Organizarea in pachetul "View" si implementarea propriu-zisa



Pachetul "View" contine clasele "UserInterface" si "Controller", si un fisier de tip ttf pentru font-ul folosit la design-ul interfetei grafice.

Clasa “UserInterface” este responsabila pentru definirea și gestionarea interfeței grafice a aplicației de calculator polinomial. Aceasta utilizează componente grafice din Java Swing pentru a crea o interfață utilizator intuitivă.

Clasa “Controller” este responsabila pentru gestionarea logică a aplicației, inclusiv validarea polinoamelor introduse de utilizator și realizarea operațiilor matematice pe acestea.

In clasa “UserInterface” - crearea butoanelor:

```
1 usage
private void createButtons() {
    Add = createStyledButton( text: "add", new Color( r: 0, g: 0, b: 128), new Color( r: 173, g: 216, b: 230));
    Subtract = createStyledButton( text: "subtract", new Color( r: 0, g: 0, b: 128), new Color( r: 173, g: 216, b: 230));
    Multiply = createStyledButton( text: "multiply", new Color( r: 0, g: 0, b: 128), new Color( r: 173, g: 216, b: 230));
    Divide = createStyledButton( text: "divide", new Color( r: 0, g: 0, b: 128), new Color( r: 173, g: 216, b: 230));
    Integrate = createStyledButton( text: "integrate", new Color( r: 0, g: 0, b: 128), new Color( r: 173, g: 216, b: 230));
    Derivate = createStyledButton( text: "derivate", new Color( r: 0, g: 0, b: 128), new Color( r: 173, g: 216, b: 230));
    createKeyboard();
}
```

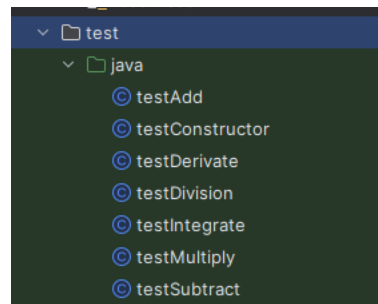
In clasa “Controller” - validarea de string-uri:

```
2 usages
public boolean isValid(String polynomial){
    for (char c : polynomial.toCharArray()) {
        if (Character.isLetter(c)) {
            //has any other letter besides x
            if (c != 'x') {
                return false;
            }
            //has any other signs besides ...
        } else if (!Character.isDigit(c) && c != '^' && c != '+' && c != '-' && c != ' ' && c != '.') {
            return false;
        }
    }
    return true;
}
```

5. Rezultate

Operatii supuse testarii:

- Adunare
- Scadere
- Inmultire
- Impartire
- Derivare
- Integrare
- Transformarea input-ului din string in polinom si din polinom in string



Pentru testarea fiecărei operatii am folosit JUnit, un framework pentru testarea unitara in limbajul de programare Java. Pentru fiecare operatie am facut cate o clasa de test separata, in care am facut doua teste folosind instructiunea “assert”, unul care trebuie trecut si unul care trebuie picat. In urma testarilor, toate operatiile au dat rezultatele asteptate.

Exemplu pentru clasa de test a operatiei de adunare:

```
import Model.Polynomial;
import Controller.Operations;
import org.junit.jupiter.api.Test;
import static org.junit.jupiter.api.Assertions.assertEquals;

public class testAdd {
    @Test
    public void testAdd1() {
        Polynomial p1 = new Polynomial( input: "x^2+3x^3+4");
        Polynomial p2 = new Polynomial( input: "x^3+x-1");

        Operations op = new Operations();
        Polynomial res = op.add(p1, p2);

        String s = res.toString(res);
        assertEquals("4x^3+x^2+x+3");
    }

    @Test
    public void testAdd2() {
        Polynomial p1 = new Polynomial( input: "x^2+3x^3+4");
        Polynomial p2 = new Polynomial( input: "x^3+x-1");

        Operations op = new Operations();
        Polynomial res = op.add(p1, p2);

        String s = res.toString(res);
        assertEquals("4x+5");
    }
}
```

6. Concluzii

Acest proiect m-a ajutat să descopăr noi funcționalități în Java, cum ar fi Regex pentru expresii regulate, Maven pentru gestionarea dependențelor și JUnit pentru testarea unitară în Java.

Pentru viitoare îmbunătățiri, s-ar putea adăuga opțiunea de a găsi rădăcinile unui polinom dat, validarea și gestionarea mai bună a erorilor și îmbunătățiri aduse interfeței (posibilitatea de a introduce polinoame în casetele text dintr-o tastatură incorporată în fereastra programului).

7. Bibliografie

1. <https://regex101.com/>
2. <https://www.geeksforgeeks.org/regular-expressions-in-java/>
3. <https://www.studysmarter.co.uk/explanations/math/pure-maths/operations-with-polynomials/>
4. <https://www.simplilearn.com/tutorials/java-tutorial/what-is-junit>
5. <https://stackoverflow.com/questions/13424302/java-tostring-method-for-polynomial-terms>
6. <https://www.javatpoint.com/java-swing>
7. https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm