DOCUMENTATIE

TEMA 1

**CALCULATOR POLINOAME**

NUME STUDENT: Maxim Francesco

GRUPA: 30224

# CUPRINS

[1. Obiectivul temei 3](#_Toc95297885)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](#_Toc95297886)

[3. Proiectare 3](#_Toc95297887)

[4. Implementare 3](#_Toc95297888)

[5. Rezultate 3](#_Toc95297889)

[6. Concluzii 3](#_Toc95297890)

[7. Bibliografie 3](#_Toc95297891)

# Obiectivul temei

*(i) Obiectiv principal: Dezvoltarea unui calculator de polinoame care să permită efectuarea diverselor operații între polinoame, precum adunare, scădere, înmulțire, împărțire, integrare, derivare și calculul restului în divizarea polinoamelor.*

*(ii) Obiective secundare:*

*Implementarea claselor Monom și Polynom care să permită reprezentarea și manipularea polinoamelor și monomilor (Capitolul 3, 4).*

*Dezvoltarea unei interfețe grafice pentru interacțiunea cu utilizatorul (Capitolul 4).*

*Asigurarea corectitudinii și eficienței operațiilor definite asupra polinoamelor prin testare adecvată (Capitolul 5).*

*Documentarea adecvată a întregului proces de dezvoltare a aplicației (Capitolul 6, 7).*

# Analiza problemei, modelare, scenarii, cazuri de utilizare

*Analiza problemei constă în definirea cerințelor funcționale și non-funcționale ale aplicației, precum și în identificarea cazurilor de utilizare.*

***Cerinte funcționale:***

* *Efectuarea operațiilor de adunare, scădere, înmulțire, împărțire, integrare și derivare asupra polinoamelor.*
* *Rezultatul fiecărei operații să fie afișat utilizatorului.*

***Cerinte non-funcționale:***

* *Interfața grafică să fie intuitivă și ușor de utilizat.*
* *Aplicația să fie eficientă din punct de vedere al performanței și să gestioneze corect toate cazurile de intrare.*

***Cazuri de utilizare:***

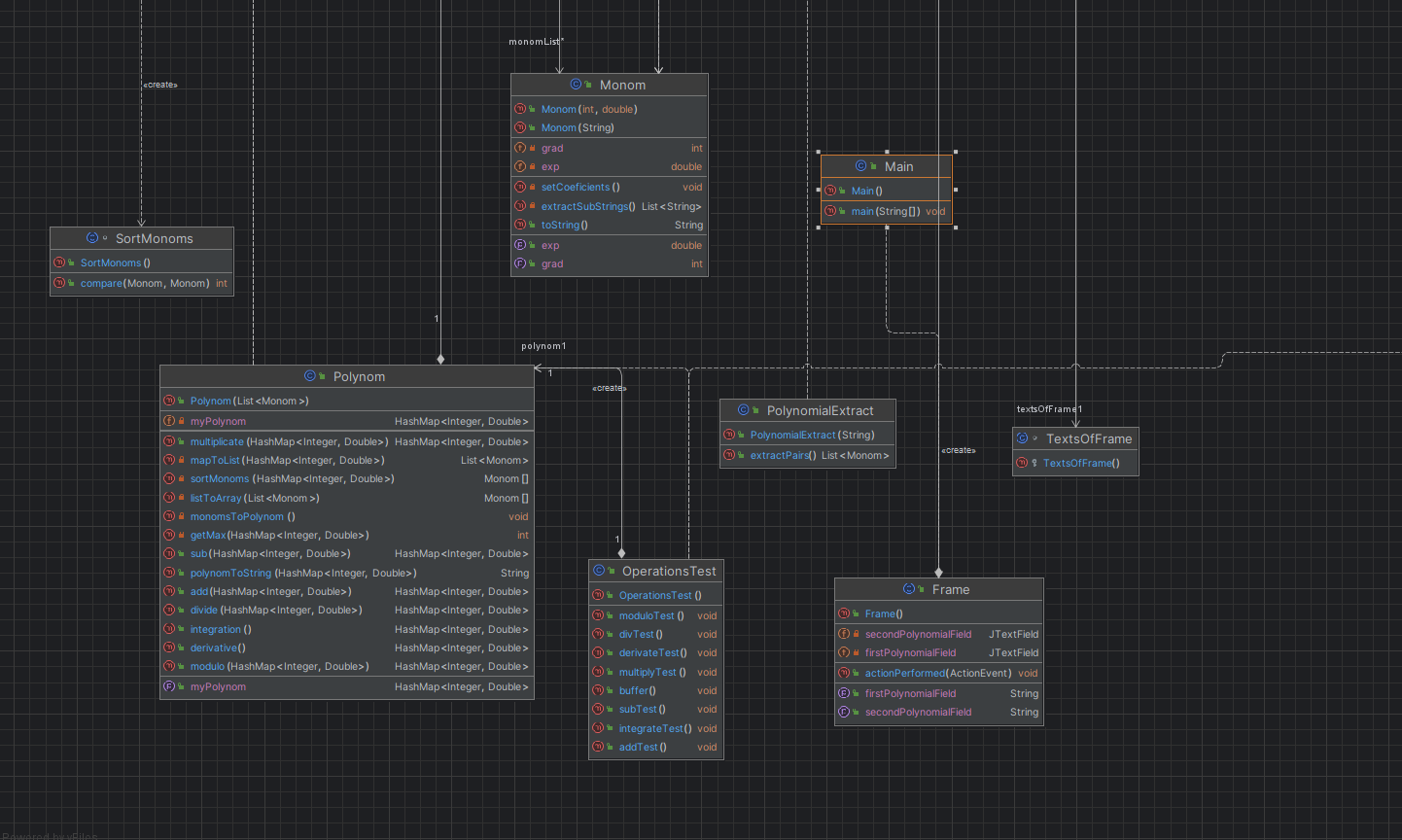
* *Utilizatorul introduce două polinoame în câmpurile corespunzătoare.*
* *Utilizatorul selectează operația dorită din lista de operații disponibile.*
* *Aplicația efectuează operația și afișează rezultatul.*

# Proiectare

Proiectarea aplicației constă în definirea structurii OOP a acesteia, a interfeței grafice și a algoritmilor necesari pentru operațiile cu polinoame.

Diagrama de clase UML: Se definește structura claselor Monom și Polynom, precum și relațiile între acestea.

Diagrama de pachete UML: Se organizează clasele în pachete corespunzătoare.



Structurile de date utilizate: Se utilizează liste și hash map-uri pentru stocarea și manipularea polinoamelor și monomilor.

Interfața definită: Se dezvoltă interfața grafică care să permită interacțiunea cu utilizatorul.

Algoritmii utilizați: Se descriu algoritmii pentru operațiile de adunare, scădere, înmulțire, împărțire, integrare și derivare.

# Implementare

***4.1 Clase și Structuri de Date***

***4.1.1 Clasa Monom***

* *Această clasă reprezintă un monom dintr-un polinom.Ea conține câmpuri pentru string-ul care reprezintă monomul (monomString), exponentul (exp) și gradul (grad).*
* *Metoda constructor Monom(String monomString):*
* *Inițializează un obiect Monom pe baza unui șir de caractere care reprezintă monomul. În funcție de acest șir, se stabilesc coeficienții monomului și sunt setate câmpurile corespunzătoare.*
* *Metoda constructor Monom(int grad, double exp): Inițializează un obiect Monom cu gradul și exponentul specificați.*
* *Metoda privată setCoeficients(): Extrage coeficienții monomului din șirul dat și îi atribuie câmpurilor exp și grad.*
* *Metoda privată extractSubStrings(): Extrage subșirurile care conțin coeficienții și gradul monomului utilizând expresii regulate.*
* *Metoda toString(): Returnează o reprezentare sub formă de șir a monomului, afișând exponentul și gradul acestuia.*
* *Metodele getExp() și getGrad(): Oferă acces la câmpurile private exp și grad.*

***4.1.2 Clasa Polynom***

* *Această clasă reprezintă un polinom și implementează operațiile aritmetice comune pe polinoame.*
* *Câmpul monomList: Lista de monomi care compun polinomul.*
* *Câmpul myPolynom: Un map care reprezintă polinomul ca o asociere între grad și coeficient.*
* *Metode principale:*
* *Metoda constructor Polynom(List<Monom> monomList): Inițializează un obiect Polynom pe baza unei liste de monomi și convertește monomii în reprezentarea internă a polinomului.*
* *Metoda polynomToString(HashMap<Integer, Double> polynom): Converteste polinomul reprezentat ca map într-un șir de caractere, reprezentând polinomul sub forma sa standard.*
* *Metoda add(HashMap<Integer, Double> number): Adaugă un alt polinom la polinomul curent.*
* *Metoda sub(HashMap<Integer, Double> number): Scade un alt polinom din polinomul curent.*
* *Metoda integration(): Efectuează operația de integrare a polinomului.*
* *Metoda derivative(): Efectuează operația de derivare a polinomului.*
* *Metoda multiplicate(HashMap<Integer, Double> number): Înmulțește polinomul curent cu un alt polinom.*
* *Metoda divide(HashMap<Integer, Double> number): Împarte polinomul curent la un alt polinom și returnează rezultatul ca un nou polinom.*
* *Metoda modulo(HashMap<Integer, Double> number): Calculează restul împărțirii polinomului curent la un alt polinom și returnează rezultatul ca un nou polinom.*
* *Metodele private monomsToPolynom(), mapToList(), listToArray(), sortMonoms(): Metode auxiliare pentru conversia și manipularea polinoamelor.*
* *4.2 Implementarea Interfeței Utilizator*
* *Interfața utilizator este implementată în clasa Frame, care extinde clasa JFrame. Aceasta este responsabilă pentru interacțiunea cu utilizatorul și apelul metodelor din clasa Polynom în funcție de acțiunile utilizatorului.*
* *Metoda actionPerformed(ActionEvent e): Gestionează evenimentele produse de utilizator, cum ar fi apăsarea butoanelor pentru efectuarea operațiilor aritmetice.*
* *Metodele getFirstPolynomialField() și getSecondPolynomialField(): Returnează conținutul introdus în câmpurile de text pentru polinoamele primite de la utilizator.*

***4.3 Testare***

* *Testarea funcționalității este asigurată prin interacțiunea cu interfața grafică. Utilizatorul poate introduce polinoame și efectua operații aritmetice asupra lor, iar rezultatele sunt afișate în interfața grafică.*

***4.4 Utilizarea JUnit (dacă este cazul)***

* *În cazul în care este necesară testarea unitară utilizând JUnit, aceasta poate fi integrată în această secțiune. Se vor furniza scenarii de testare și rezultatele acestora.*

***4.5 Exemplu de Cod***

*// Exemplu de utilizare a interfeței grafice*  
*public static void main(String[] args) {*  
 *Frame gui = new Frame();*  
*}*

***4.6 Capturi de Ecran***

* *Se pot adăuga capturi de ecran pentru a ilustra funcționalitățile aplicației și modul în care acestea sunt implementate în interfața grafică.*

***4.7 Observații***

* *Se pot adăuga observații, precum potențiale îmbunătățiri sau dificultăți întâmpinate în timpul implementării.*

# Rezultate

Pentru testarea funcționalității și corectitudinii aplicației, au fost definite mai multe scenarii de testare care acoperă operațiile de adunare, scădere, înmulțire, împărțire, calculul restului și a derivatelor și integralelor pentru polinoamele introduse de utilizator.

**5.1 Scenarii de Testare**

Adunare a două polinoame:

* Introducerea a două polinoame în interfața grafică.
* Apăsarea butonului "Adunare".
* Verificarea afișării rezultatului corect al adunării în interfața grafică.

Scădere a două polinoame:

* Introducerea a două polinoame în interfața grafică.
* Apăsarea butonului "Scădere".
* Verificarea afișării rezultatului corect al scăderii în interfața grafică.

Înmulțire a două polinoame:

* Introducerea a două polinoame în interfața grafică.
* Apăsarea butonului "Înmulțire".
* Verificarea afișării rezultatului corect al înmulțirii în interfața grafică.

Împărțire a două polinoame:

* Introducerea a două polinoame în interfața grafică.
* Apăsarea butonului "Împărțire".
* Verificarea afișării rezultatului corect al împărțirii în interfața grafică.

Calculul restului împărțirii a două polinoame:

* Introducerea a două polinoame în interfața grafică.
* Apăsarea butonului "Modulo".
* Verificarea afișării corecte a restului împărțirii în interfața grafică.

Calculul derivatei unui polinom:

* Introducerea unui polinom în interfața grafică.
* Apăsarea butonului "Derivare".
* Verificarea afișării corecte a derivatei în interfața grafică.

Calculul integralei unui polinom:

* Introducerea unui polinom în interfața grafică.
* Apăsarea butonului "Integrare".
* Verificarea afișării corecte a integralei în interfața grafică.

# Concluzii

*Implementarea unei aplicații de calcul polinomial reprezintă o modalitate eficientă de a înțelege și de a aplica conceptele de programare orientată obiect și de a dezvolta abilități practice în dezvoltarea de aplicații cu interfață grafică.*

*În timpul realizării acestui proiect, am învățat cum să gestionăm și să manipulăm polinoame în cadrul unei aplicații Java, precum și cum să proiectăm și să implementăm interfețe grafice utilizator intuitiv și funcțional.*

*Deși aplicația oferă funcționalități de bază pentru operațiile aritmetice pe polinoame, există întotdeauna posibilitatea de a îmbunătăți și de a extinde aplicația prin adăugarea de funcționalități suplimentare, cum ar fi suport pentru polinoame cu coeficienți complecși sau implementarea unui algoritm mai sofisticat pentru divizarea polinoamelor.*

# Bibliografie

*Se vor adauga referintele care au fost consultate de student pe parcursul implementarii temei .*

*Exemplu:*

1. *Bruce Eckel, Thinking in Java (4th Edition), Publisher: Prentice Hall PTRUpper Saddle River, NJUnited States, ISBN:978-0-13-187248-6 Published:01 December 2005.*
2. *What are Java classes? -* [*www.tutorialspoint.com*](http://www.tutorialspoint.com)