DOCUMENTATIE

TEMA POLYNOMIAL CALCULATOR

NUME STUDENT: Rafa Ioana-Sorina

GRUPA: 30223

# 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

1. Obiectivul principal al temei este de a dezvolta o aplicatie Java pentru operatii polinomiale, care sa ofere utilizatorilor posibilitatea de a efectua adunarea, scaderea, inmultirea, derivarea si integrarea a doua polinoame.
2. Obiective secundare:

- Implementarea clasei Polynomial pentru reprezentarea si manipularea polinoamelor. [Detaliat in capitolul "Implementarea"]

- Dezvoltarea clasei Logic care sa ofere operatiile necesare pe polinoame.

[Detaliat in capitolul "Implementarea"]

- Crearea unei interfete grafice (GUI) pentru utilizator, folosind Java Swing, pentru a interactiona cu operatiile polinomiale. [Detaliat in capitolul "Implementarea"]

- Testarea unitara a functionalitatilor implementate pentru a asigura corectitudinea si robustetea aplicatiei. [Detaliat in capitolul "Testarea"]

- Documentarea codului si a functionalitatilor aplicatiei pentru a facilita intelegerea si intretinerea acestuia. [Detaliat in capitolul "Documentarea"]

# Analiza problemei, modelare, scenarii, cazuri de utilizare

*Cerinte functionale:*

1. *Adunarea polinomiala: Utilizatorul poate aduna doua polinoame.*
2. *Scaderea polinomiala: Utilizatorul poate scadea doua polinoame.*
3. *Inmulirea polinomiala: Utilizatorul poate inmulti doua polinoame.*
4. *Derivarea polinomiala: Utilizatorul poate calcula derivarea unui polinom.*
5. *Integrarea polinomiala: Utilizatorul poate calcula integrarea unui polinom.*

*Cerinte non-functionale:*

1. *Eficienta: Operatiile polinomiale trebuie sa fie efectuate rapid, fara interzieri semnificative.*
2. *Fiabilitatea: Aplicatia trebuie sa fie robusta si sa gestioneze correct toate cazurile utilizate.*
3. *Usurinta de utilizare: Interfata grafica trebuie sa fie intuitive si usor de folosi pentru unitilizatorii finali.*

*Cazuri de utilizare:*

In acest calculator se vor introduce doua polinoame preluate prin intremediul text field-urilor din interfata grafica. Dupa ce doua polinoame au fost introduse in intervalele de text ale interfetei grafice, acest calculator le va procesa si interpreta ca stringuri folosind tehnica de corelare a pattern-urilor pentru a modela matematic.

Un scenariu de utilizare ar fi introducerea polinoamelor de catre utilizator (exact sub aceasta forma! detectabila prin corelare de pattern):

Primul polinom: 4x^2+1x^2+3x^4

Al 2-lea polinom: 4x^5-1x^2+5x^4

In cazul in care utilizatorul apasa pe butoanele care au listeneri, se vor afisa rezultatele operatiilor asociate cu textul de pe aceste butoane, deci daca userul apasa:

Addition–> Rezultatul afisat: 4x^5+8x^4+4x^2

Subtraction–> Rezultatul afisat: -4x^5-2x^4+6x^2

Multiplication–> Rezultatul afisat: 12x^9+15x^8+20x^7+22x^6-5x^4

Derivation–> Rezultatul afisat: +12x^3+10x^1 (derivarea are loc doar pentru primul polinom)

Integration–> Rezultatul afisat: 0.6x^5 +1.6666666x^3 (integrarea are loc doar pentru primul polinom)

# Proiectare

Am folosit JFrame pentru a crea o interfata de utilizator GUI simpla si usor de folosit. Rezultatul dorit va fi afisat instantaneu daca datele au fost introduse corect. Formatul acestui frame este vazut in figura care urmeaza, care este asemanator cu cel al unui calculator clasic.

A screenshot of a computer

Description automatically generated

Am create clasele:

* Polynomial
* Calculator
* Logic
* Main
* TestOp

A diagram of a computer

Description automatically generated

# Implementare

*Clase:*

1. *Clasa Polynomial*

* Atribute:
  + polyMap : TreeMap utilizat pentru a stoca coeficientii si gradele polinomului.
  + Text : Sir de caractere utilizat pentru a reprezenta polinomul in format text.
* Metode importante:
* addTerm(float coefficient, int grad) : Adauga un nou termen (coeficient si grad) in polinom.
* toString() : Returneaza o reprezentare text a polinomului.
* freeP() : Curata polinomul, eliminand toti termenii.
* removeZeroCoefficients() : Elimina termenii cu coeficienti zero din polinom.

1. Clasa Logic:

* Metode importante:
* additionPolynomial(Polynomial polinom1, Polynomial polinom2): Realizeaza adunarea a doua polinoame.
* subtractionPolynomial(Polynomial polinom1, Polynomial polinom2): Realizeaza scaderea a doua polinoame.
* multiplicationPolynomial(Polynomial polinom1, Polynomial polinom2): Realizeaza inmultirea a doua polinoame.
* divisionPolynomial(Polynomial polinom1, Polynomial polinom2): Realizeaza impartirea a doua polinoame (nu implementata in prezent).
* derivativePolynomial(Polynomial polinom1): Calculeaza derivata unui polinom.
* integrationPolynomial(Polynomial polinom1): Calculeaza integrala unui polinom.

1. Clasa Calculator:

* Atribute:
* polinom1Field, polinom2Field, rezultatField: JTextField-uri pentru introducerea polinoamelor si afisarea rezultatului.
* calculatorLogic: O instanta a clasei Logic pentru a efectua operatiile polinomiale.
* Metode importante:
* initComponents(): Initializeaza toate componentele interfetei grafice.
* Metodele ListenAdd(), ListenSub(), ListenMul(), ListenDiv(), ListenDer(), ListenInte(): Adauga ascultatori pentru butoanele de operatii polinomiale.

Implementarea interfetei utilizatorului:

Interfata utilizatorului este implementata folosind Java Swing si include componente precum JTextField-uri pentru introducerea polinoamelor, butoane pentru diferite operatii polinomiale si un JTextField pentru afisarea rezultatului.

Utilizatorul poate introduce polinoamele in campurile corespunzatoare, apoi poate efectua operatii folosind butoanele aferente. Rezultatul operatiei este afisat in JTextField-ul rezultatului.

Pentru fiecare operatie, sunt definite metode care interactioneaza cu clasa Logic pentru a efectua operatia corespunzatoare si pentru a afisa rezultatul in interfata utilizatorului.

# Rezultate

Clasa TestOp este inclusa in folderul proiectului numit „test”, care este special dedicat testarii functionalitatilor programelor si este responsabila de verificarea corectitudinii operatiilor. Au fost create clasa TestOp si cateva teste operationale legate pentru a verifica corectitudinea functionalitatii fiecareia prin utilizarea pachetelor JUnit, care au fost descrise si sugerate pentru utilizare in resuresele de suport de laborator.

Folosind-o, putem vedea succesul testarii fiecareia dintre cele cinci operatii ale calculatorului de polinoame care indica functionarea optima a acestora.

Aceste teste compara rezultatele returnate de functii pentru niste polinoame cunoscute cu rezultatul corect.

A screenshot of a computer

Description automatically generated

# Concluzii

* Proiectul a dus la dezvoltarea unei aplicatii Java care permite utilizatorilor sa efectueze operatii polinomiale de adunare, scadere, inmultire, derivare si integrare. Implementarea a inclus clasele necesare pentru reprezentarea si manipularea polinoamelor, precum si o interfata grafica (GUI) intuitiva pentru interactiunea cu utilizatorul
* Dezvoltarea acestui proiect a oferit oportunitatea de a lucra cu concepte OOP (Programare Orientata pe Obiecte) in Java si de a aplica diverse principii si tehnici de proiectare software. In plus, gestionarea datelor si a interfetei utilizatorului in cadrul aplicatiei a implicat explorarea si invatarea unor concepte precum evenimente GUI si gestionarea interactiunii cu utilizatorul.
* Exista mai multe modalitati prin care proiectul poate fi extins si imbunatatit in viitor. Acestea includ adaugarea de noi functionalitati, cum ar fi suportul pentru divizarea polinoamelor, optimizarea performantei algoritmilor existenti, imbunatatirea aspectului si simplitatii interfetei utilizatorului si extinderea aplicatiei pentru a permite integrarea cu alte sisteme sau platforme.

# Bibliografie

1. [*https://uncoded.ro/dezvoltarea-aplicatiilor-cu-interfata-grafica-in-java/*](https://uncoded.ro/dezvoltarea-aplicatiilor-cu-interfata-grafica-in-java/)
2. https://uncoded.ro/componente-grafice-in-aplicatiile-java-cu-interfata-grafica/
3. Proiecte precendente la POO
4. Class Materials > Laborator