DOCUMENTATIE

TEMA 1

Calculator de polinoame

NUME STUDENT: Filip-Dud Cristian Călin

GRUPA: 30228

# CUPRINS

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

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 4](file:///C:\Users\Filip-Dud%20Cristian\Downloads\PT2023_Template_Documentatie_RO%20(3).doc#_Toc95297886)

[3. Proiectare 9](file:///C:\Users\Filip-Dud%20Cristian\Downloads\PT2023_Template_Documentatie_RO%20(3).doc#_Toc95297887)

[4. Implementare 11](file:///C:\Users\Filip-Dud%20Cristian\Downloads\PT2023_Template_Documentatie_RO%20(3).doc#_Toc95297888)

[5. Rezultate 15](file:///C:\Users\Filip-Dud%20Cristian\Downloads\PT2023_Template_Documentatie_RO%20(3).doc#_Toc95297889)

[6. Concluzii 17](file:///C:\Users\Filip-Dud%20Cristian\Downloads\PT2023_Template_Documentatie_RO%20(3).doc#_Toc95297890)

[7. Bibliografie 17](file:///C:\Users\Filip-Dud%20Cristian\Downloads\PT2023_Template_Documentatie_RO%20(3).doc#_Toc95297891)

1. Obiectivul temei

Obiectivul principal al temei este de a proiecta si implementa un calculator polinomial cu o interfata grafica dedicata prin care utilizatorul poate insera polinoame, poate selecta operatia matematica care trebuie efectuata si poate vizualiza rezultatul.

Pentru indeplinirea acestuia, trebuie indeplinite obiective secundare precum:

● Analiza problemei si identificarea cerintelor – capitolul 2

● Proiectarea calculatorului pentru polinoame – capitolul 3

● Implementarea calculatorului de polinoame – capitolul 4

● Testarea calculatorului de polinoame – capitolul 5

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

Cerinta problemei este cea de a proiecta si implementa un sistem de procesare a polinoamelor. Pentru a realiza acest lucru este nevoie sa avem cunostiinte teoretice din matematica care stau la baza polinoamelor, precum forma lor si operatiile care pot fi efectuate cu acestea.

Un polinom este o sumă de unul sau mai multe monoame. Un monom este alcatuit dintr-o constanta numita coeficient, inmultita cu una sau mai multe variabile.

Operatiile polinoamelor:

* **Adunarea:** se aduna coeficientii monoamelor de grade egale
* **Scaderea:** se scad coeficientii monoamelor de grade egale
* **Inmultirea:** se inmulteste fiecare monom pe rand cu fiecare monom al celui de al doilea polinom si se insumeaza monoamele rezultate
* **Impartirea:** se urmareste urmatorul algoritm de impartire:

Pasul 1 - Se ordoneaza descrescator monoamele celor doua polinoame P si Q in functie de gradul lor.

Pasul 2 - Se imparte polinomul cu gradul cel mai inalt la celalalt polinom cu gradul mai mic (se ia in considerare faptul ca P are cel mai inalt grad).

Pasul 3 - Se imparte primul monom al lui P la primul monom al lui Q si se obține primul termen al coeficientului.

Pasul 4 - Se inmulteste catul cu Q si se scade rezultatul inmultirii din P, obtinand restul.

Pasul 5 - Se repeta procedura de la pasul 2, considerand restul drept noul divizor, pana cand gradul restului este mai mic decat cel al lui Q.

* **Derivarea:** se inmulteste coeficientul fiecarui monom cu puterea, aceasta fiind scazuta cu 1
* **Integrarea:** se aduna gradul fiecarui monom cu 1, apoi coeficientul este impartit cu gradul

Cazurile de utilizare sunt prezentate cu ajutorul unei diagrame use-case. Actorul este reprezentat de orice utilizator al aplicatiei care doreste sa efectueze operatii cu polinoame

Diagram

Description automatically generated

1. Adunarea a doua polinoame

Use-case: Adunare

Primary actor: Utilizator

Main success scenario:

1. Utilizatorul introduce 2 polinoame in interfata grafica
2. Se apasa pe butonul „Adunare”
3. Aplicatia efectueaza operatia
4. Rezultatul operatiei este afisat

Alternative sequences:

-Utilizatorul introduce date care nu corespund formei de polinom, afisandu-se un mesaj de avertizare

1. Scaderea a doua polinoame

Use-case: Scadere

Primary actor: Utilizator

Main success scenario:

1. Utilizatorul introduce 2 polinoame in interfata grafica
2. Se apasa pe butonul „Scadere”
3. Aplicatia efectueaza operatia
4. Rezultatul operatiei este afisat

Alternative sequences:

-Utilizatorul introduce date care nu corespund formei de polinom, afisandu-se un mesaj de avertizare

1. Inmultirea a doua polinoame

Use-case: Inmultire

Primary actor: Utilizator

Main success scenario:

1. Utilizatorul introduce 2 polinoame in interfata grafica
2. Se apasa pe butonul „Inmultire”
3. Aplicatia efectueaza operatia
4. Rezultatul operatiei este afisat

Alternative sequences:

-Utilizatorul introduce date care nu corespund formei de polinom, afisandu-se un mesaj de avertizare

1. Impartire a doua polinoame

Use-case: Impartire

Primary actor: Utilizator

Main success scenario:

1. Utilizatorul introduce 2 polinoame in interfata grafica
2. Se apasa pe butonul „Impartire”
3. Aplicatia efectueaza operatia
4. Catul si restul operatiei sunt afisate

Alternative sequences:

-Utilizatorul introduce date care nu corespund formei de polinom, afisandu-se un mesaj de avertizare

1. Derivare a doua polinoame

Use-case: Derivare

Primary actor: Utilizator

Main success scenario:

1. Utilizatorul introduce polinomul in campul primului polinom din interfata grafica
2. Se apasa pe butonul „Derivare”
3. Aplicatia efectueaza operatia
4. Rezultatul operatiei este afisat

Alternative sequences:

-Utilizatorul introduce date care nu corespund formei de polinom, afisandu-se un mesaj de avertizare

1. Integrare a doua polinoame

Use-case: Integrare

Primary actor: Utilizator

Main success scenario:

1. Utilizatorul introduce polinomul in campul primului polinom din interfata grafica
2. Se apasa pe butonul „Integrare”
3. Aplicatia efectueaza operatia
4. Rezultatul operatiei este afisat

Alternative sequences:

-Utilizatorul introduce date care nu corespund formei de polinom, afisandu-se un mesaj de avertizare

3. Proiectare

A screenshot of a computer program

Description automatically generated with low confidenceDiagrama UML:

Aplicatia contine 4 clase:

* Monom: aceasta clasa contine atributele unui monom
* Polinom: aceasta clasa contine operatiile specifice unui polinom
* Parsing: aceasta clasa continue metode care ajuta la extragerea polinoamelor introduse de utilizator pentru a aplica operatii pe acestea
* Interfata: aceasta clasa contine functia main si declararea si instantierea elementelor de interfata grafica

4. Implementare

* Monom

Clasa reprezintă un monom matematic, care este o expresie algebraică constând dintr-un singur termen.

* Clasa are două câmpuri private: grad, care reprezintă gradul monomului de tipul Integer, și coeficient, care reprezintă coeficientul monomului de tipul Double.
* Clasa oferă metode de accesare și modificare a ambelor câmpuri.
* Clasa suprascrie metoda toString(), care returnează o reprezentare sub formă de șir a monomului în formatul "coeficientx^grad".
* Clasa suprascrie și metoda equals(), care compară câmpurile grad și coeficient ale două obiecte Monom pentru a determina dacă sunt egale.
* Polinom

Clasa definește un polinom utilizând un HashMap unde cheile sunt exponenții și valorile sunt obiecte ale clasei Monom. Clasa Polinom are mai multe metode pentru manipularea polinoamelor.

* Constructor care inițializează HashMap-ul care stochează monoamele polinomului, gradul fiind folosit ca si cheie.
* Metoda addMonom() adaugă un monom la polinom prin inserarea acestuia în map.
* Metoda cleanPolinom() elimină orice monom cu un coeficient de 0 dintr-un polinom.
* Metoda getMaxGradeMonom() returnează monomul cu cel mai mare grad dintr-un polinom.
* Metoda sumPolinom() adaugă două polinoame împreună prin adunarea coeficienților monoamelor cu același grad și crearea unui nou monom cu suma.
* Metoda substractPolinom() scade un polinom din altul prin scăderea coeficienților monoamelor cu același grad și crearea unui nou monom cu diferența.
* Metoda multiplyPolinom() înmulțește două polinoame prin înmulțirea fiecărui monom dintr-un polinom cu fiecare monom din celălalt polinom, adunând coeficienții monoamelor rezultate cu același grad și creând un nou monom cu suma.
* Metoda dividePolinom() împarte un polinom la un alt polinom folosind divizia lungă, returnând o listă de două polinoame: catul și restul. Aruncă o excepție dacă divizorul este un polinom nul.
* Parsing

Clasa se ocupa de analizarea și convertirea unui șir de caractere într-un obiect de tip Polinom, folosind un set de reguli Pattern care definesc o sintaxă corectă pentru un Polinom.

* Metoda numită monomMatcher() care primește un șir de caractere și returnează un obiect de tip Monom. Această metodă verifică șirul pentru a determina coeficientul și gradul monomului, iar apoi creează un obiect de tip Monom corespunzător.
* Metoda parse() primește un șir de caractere care reprezintă un Polinom și îl convertește într-un obiect de tip Polinom. Înainte de a face acest lucru, aceasta verifică șirul pentru a se asigură că urmează regulile de scriere și inserare a polinoamelor folosind clasele Pattern și Matcher, respectiv metode ale acestora. Metoda descompune șirul în monomi utilizând expresia regulată definită anterior și construiește un obiect de tip Polinom folosind metoda monomMatcher.
* Interfata

Clasa permite utilizatorului să introducă două polinoame în două câmpuri text, să efectueze operații asupra lor și să afișeze rezultatele într-un al treilea câmp text folosind o interfață grafică de utilizator.

* GUI-ul este construit folosind componente Java Swing, cum ar fi JFrame, JLabel, JTextField și JButton.
* Fiecarui buton îi este implementat un ActionListener care face operația respectivă pe polinoamele extrase folosind metoda de parsare din câmpurile text.

Timeline

Description automatically generated

5. Rezultate

Pentru a verifica si valida veridicitatea operațiilor implementate, am definit o clasă PolinomTest pentru teste implementate cu framework-ul JUnit.

Adunare:

Pentru a testa operația de adunare am definit următoarele polinoame:

P1= x^3 +2x^2 +3x

P2= 10x^5 +x^3 + 2.5x^2 +3x

Iar rezultatul așteptat este următorul: 10x^5 +2x^3 +4.5x^2 +6x

Scădere:

Pentru a testa operația de scădere am definit următoarele polinoame:

P1= 2x^3 +4x^2 +5x

P2= 10x^5 +x^3 + 2.5x^2 +3x

Iar rezultatul așteptat este următorul: 10x^5 -1x^3 -1.5x^2 -2x

Înmulțire:

Pentru a testa operația de înmulțire am definit următoarele polinoame:

P1= x^3 +3x^2 -2x

P2= x^2 +5

Iar rezultatul așteptat este următorul: x^5 +3x^4 +3x^3 +15x^2 -10x

Împărțire:

Pentru a testa operația de împărțire am definit următoarele polinoame:

P1= 32x^6 +5x^4 -4x^3 -12x^2 -2

P2= x^3 -2x^2 -x +4

Iar rezultatul așteptat este următorul:

Cât= 32x^3 +64x^2 +165x +262

Rest= 421x^2 -398x -1050

Derivare:

Pentru a testa operația de derivare am definit următorul polinom:

P1= 7x^6 -10x^5 +5x^3 +1.3x^2 +2x -5

Iar rezultatul așteptat este următorul: 42x^5 -50x^4 +15x^2 +2.6x +2

Integrare:

Pentru a testa operația de integrare am definit următorul polinom:

P1= x^3 +6x^2 +5

Text

Description automatically generatedIar rezultatul așteptat este următorul: 0.25x^4 +2x^3 +5x

6. Concluzii

Dezvoltarea acestei teme m-a ajutat în a aprofunda cunoștiințele legate de modul de lucru și pașii care trebuie urmați în proiectarea, dezvoltarea și implementarea de aplicații Java complete.

Idei pentru dezvoltări ulterioare:

* Dezvoltarea funcției de parsare pentru a acoperi mai multe cazuri d date de utilizatori
* Implementarea unui istoric al operațiilor
* Aflarea rădăcinilor polinoamelor
* Eficentizarea operațiilor implementate

7. Bibliografie

* <https://www.cuemath.com/algebra/long-division-of-polynomials/>
* <https://stackoverflow.com/questions/36490757/regex-for-polynomial-expression>