

**Procesarea Polinoamelor**

**Documentatie**

**Student:**

**Manoilescu Victor**

Cuprins:

1. Obiectivul temei
2. Analiza problemei; Modelare;Scenarii

Cazuri de uitlizare

1. Proiectare
2. Implementare
3. Rezultate
4. Concluzii
5. Bibliografie

***1.Obiectivul temei:***

* Obiectivul principal al temei “Procesarea Polinoamelor” consta in proiectarea si implementarea unui calculator de polinoame cu o interfata grafica prin care utilizatorul poate insera polinoame asupra carora se pot aplica diferite operatii matematice si care va afisa rezultatul operatiei selectate
* Indeplinirea obiectivului principal necesita indeplinirea unor obiective secundare:

1. Analizarea problemei si identificarea cerintelor:

* Pentru a putea implementa aplicatia trebuie intai sa avem o idee concreta a ce ne dorim de la aceasta. Astfel, intelegerea conceptelor matematice din spatele operatiilor este cruciala.
* Dezvoltarea acestui subiect se afla in capitolul 2

1. Identificarea si solutionarea cazurilor de utilizare ale aplicatiei din ceea ce priveste interactiunea utilizatorului cu aceasta:

* Este necesar sa acoperim toate cazurile de utilizare astfel incat aplicatia sa functioneze cum trebuie in orice circumstanta
* Dezvoltarea acestui subiect se afla in capitolul 2

1. Dezvoltarea si proiectarea aplicatiei conform cerintelor din barem respectand paradigmele programarii orientate pe obiecte:

* Se descriu structurile de date si algoritmii utilizati in implementare (capitolul 3 si 4)
* Se prezinta structurarea aplicatiei in pachete, clase si relatiile dintre acestea ( capitolul 3 si 4)
* Se vor utiliza diagrame UML pentru exemplificarea legaturilor dintre componente

1. Testarea si evaluarea rezultatelor obtinute:

* Se vor realiza teste pe operatii folosind JUnit.
* Dezvoltarea acestui subiect se afla in capitolul 5

***2.Analiza problemei; Modelare; Scenarii; Cazuri de utilizare:***

Cerinta problemei este: “Proiectati si implementati un calculator de polinoame”.

Pentru rezolvarea acestei problemei intai ne trebuie o suita de cunostinte legate de operatiile asupra polinoamelor:

1. Adunarea polinoamelor: Se insumeaza coeficientii monoamelor de acelasi grad.
2. Scaderea polinoamelor: Se scad coeficientii monoamelor din al doilea polinom cu acelasi grad din primul polinom
3. Inmultirea polinoamelor: Se inmulteste fiecare monom din primul polinom cu fiecare monom din al doilea polinom dupa regula
4. Impartirea polinomului:

* Se imparte termenul cu gradul cel mai mare din polinomul 1

la cel mai mare din polinomul 2, catul fiind primul termen al catului final

* Se ia catul si se inmulteste cu polinomul 2, apoi se scade din polinomul 1
* Se repeta procesul pana cand restul are un grad mai mic decat polinomul 2.

1. Derivarea unui polinom: Se face dupa formula
2. Integrarea unui polinom: Se face dupa formula

***Cazuri de utilizare:***

* Cazurile de uitlizare vor fi evidentiate prin daigrama use-case,

unde Actorul este reprezentat de orice utilizator al aplicatiei

Diagram

Description automatically generated

1. Adunarea a doua polinoame

Use-case: Adunare

Primary actor: Utilizator

Main success scenario:

* Utilizatorul introduce primul polinom in campul corespunzator
* Utilizatorul introduce al doilea polinom in campul corespunzator
* Utilizatorul apasa pe butonul de adunare
* Aplicatia efectueaza operatia
* Rezultatul sumei este afisat in zona de rezultate

Alternative sequences:

1. Datele introduse de utilizator nu corespund unui polinom definit in mod corect matematic:

* Aplicatia afiseaza un mesaj care sa alerteze utilizatorul cu privire cu invaliditatea datelor
* Utilizatorul trebuie sa reitroduca un nou set de date

1. Unul dintre polinoame nu a fost introdus:

* Rezultatul este polinomul introdus

1. Scaderea a doua polinoame

Use-case: Scadere

Primary actor: Utilizator

Main success scenario:

* Utilizatorul introduce primul polinom in campul corespunzator
* Utilizatorul introduce al doilea polinom in campul corespunzator
* Utilizatorul apasa pe butonul de scadere
* Aplicatia efectueaza operatia
* Rezultatul diferentei este afisat in zona de rezultate

Alternative sequences:

1. Datele introduse de utilizator nu corespund unui polinom definit in mod corect matematic

* Aplicatia afiseaza un mesaj care sa alerteze utilizatorul cu privire cu invaliditatea datelor
* Utilizatorul trebuie sa reitroduca un nou set de date

1. Unul dintre polinoame nu a fost introdus

* Polinomul neintrodus e considerat ca fiind 0

1. Inmultirea a doua polinoame

Use-case: Inmultire

Primary actor: Utilizator

Main success scenario:

* Utilizatorul introduce primul polinom in campul corespunzator
* Utilizatorul introduce al doilea polinom in campul corespunzator
* Utilizatorul apasa pe butonul de inmultire
* Aplicatia efectueaza operatia
* Rezultatul inmultirii este afisat in zona de rezultate

Alternative sequences:

1. Datele introduse de utilizator nu corespund unui polinom definit in mod corect matematic

* Aplicatia afiseaza un mesaj care sa alerteze utilizatorul cu privire cu invaliditatea datelor
* Utilizatorul trebuie sa reitroduca un nou set de date

1. Unul dintre polinoame nu a fost introdus

* Polinomul neintrodus e considerat ca fiind 0

1. Impartirea a doua polinoame

Use-case: Impartire

Primary actor: Utilizator

Main success scenario:

* Utilizatorul introduce primul polinom in campul corespunzator
* Utilizatorul introduce al doilea polinom in campul corespunzator
* Utilizatorul apasa pe butonul de impartire
* Aplicatia efectueaza operatia
* Catul si restul impartirii sunt afisate in zona de rezultate

Alternative sequences:

1. Datele introduse de utilizator nu corespund unui polinom definit in mod corect matematic

* Aplicatia afiseaza un mesaj care sa alerteze utilizatorul cu privire cu invaliditatea datelor
* Utilizatorul trebuie sa reitroduca un nou set de date

1. Unul dintre polinoame nu a fost introdus

* Polinomul este luat ca fiind 0

1. Impartitorul este defapt un polinom nul:

* Rezultatul este 0

1. Derivarea unui polinom

Use-case: Derivare

Primary actor: Utilizator

Main success scenario:

* Utilizatorul introduce primul polinom in campul corespunzator primului polinom
* Utilizatorul apasa pe butonul de derivare
* Aplicatia efectueaza operatia
* Rezultatul derivarii este afisat in zona de rezultate

Alternative sequences:

1. Datele introduse de utilizator nu corespund unui polinom definit in mod corect matematic

* Aplicatia afiseaza un mesaj care sa alerteze utilizatorul cu privire cu invaliditatea datelor
* Utilizatorul trebuie sa reitroduca un nou set de date

1. Polinomul nu a fost introdus

* Rezultatul este 0

1. Integrarea unui polinom

Use-case: Derivare

Primary actor: Utilizator

Main success scenario:

* Utilizatorul introduce primul polinom in campul corespunzator primului polinom
* Utilizatorul apasa pe butonul de derivare
* Aplicatia efectueaza operatia
* Rezultatul integrarii este afisat in zona de rezultate

Alternative sequences:

1. Datele introduse de utilizator nu corespund unui polinom definit in mod corect matematic

* Aplicatia afiseaza un mesaj care sa alerteze utilizatorul cu privire cu invaliditatea datelor
* Utilizatorul trebuie sa reitroduca un nou set de date

1. Polinomul nu a fost introdus

* Rezultatul va fi 0 + C

***3.Proiectare***

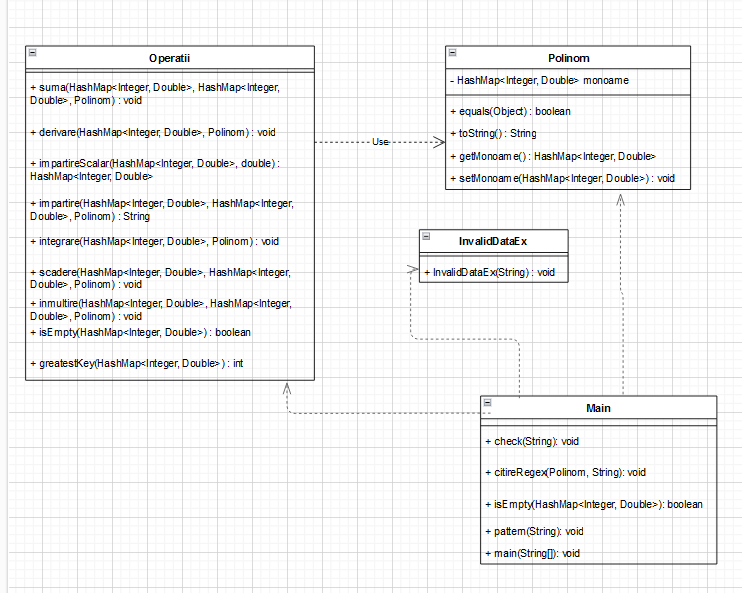
Acest capitol va prezenta diagramele UML pentru pachete si pentru clase.

Diagrama UML pentru package-uri:

Diagram

Description automatically generated

Diagrama UML pentru clase:



***4.Implementare***

Aplicatia contine 4 clase :

1. Clasa Polinom:

* Are un singur atribut: un HashMap ce contine chei de tip Integer, ce reprezinta puterile monoamelor si valori de tip Double, ce reprezinta coeficientii monoamelor.
* Metode:
* *Equals:* Este folosit pentru a verifica daca 2 polinoame sunt egale
* *Getters and setters:* Folositi pentru accesarea datelor
* *ToString:* Translateaza HashMap-ul intr-o forma citibila

Text

Description automatically generated

1. Clasa Operatii:

* Utilizata doar pentru efectuarea operatiilor asupra polinoamelor
* Nu are atribute
* Metode:
* Suma:
* Primeste ca parametru doua HashMap-uri si un polinom
* Face suma polinoamelor reprezentate de HashMap-uri si pune rezultatul in parametrul polinom;

Text

Description automatically generated

* Scadere:
* Primeste ca parametru doua HashMap-uri si un polinom
* Face diferenta polinoamelor reprezentate de HashMap-uri si pune rezultatul in parametrul polinom;

Text

Description automatically generated

* Inmultire:
* Primeste ca parametru doua HashMap-uri si un polinom
* Face inmultirea polinoamelor reprezentate de HashMap-uri si pune rezultatul in parametrul polinom;

Text

Description automatically generated

* Impartire:
* Primeste ca parametru doua HashMap-uri si un polinom
* Face impartirea polinoamelor reprezentate de HashMap-uri si pune rezultatul in parametrul polinom;
* Returneaza restul sub forma de String;

Text

Description automatically generated

* Derivare:
* Primeste ca parametru un HashMap si un polinom
* Deriveaza polinomul reprezentat de HashMap si pune rezultatul in parametrul polinom;
* Integrare:
* Primeste ca parametru un HashMap si un polinom
* Integreaza polinomul reprezentat de HashMap si pune rezultatul in parametrul polinom;

Text

Description automatically generated

Metode auxiliare:

* *isEmpty:* Verifica daca HashMap-ul este gol sau nu
* *greatestKey:* Returneaza cea mai mare cheie a unui HashMap
* *impartireScalar:* Caz particular al impartirii in care impartirea se face cu un scalar.

1. *Clasa InvalidDataEx:*

* Utilizata doar pentru eroarea custom;

1. Clasa Main

* Contine interfata grafica
* Contine metoda de parsare ce utilizeaza Regex:
  + Sunt extrase monoamele
  + Sunt verificate cazurile particulare
  + Se insereaza datele in HashMap daca sunt valide

***5.Rezultate***

1. Adunare: Testul Junit s-a facut pe polinoamele p1 = x^2 + x + 2

si p2 = 2x^2 + 3x + 1 si se asteapta raspunsul rez = 3x^2 + 4x + 3

1. Scadere: Testul Junit s-a facut pe polinoamele p1 = x^2 + x + 2

si p2 = 2x^2 + 3x + 1 si se asteapta raspunsul rez = -x^2 + -2x + 1

1. Inmultire: Testul Junit s-a facut pe polinoamele p1 = x + 2

si p2 = x^2 + 3x + 1 si se asteapta raspunsul rez = x^3 + 5x^2 + 7x + 2

1. Impartire: Testul Junit s-a facut pe polinoamele p1 = x^2 + 2x +1

si p2 = x + 4 si se asteapta raspunsul rez = x – 2 cu restul 9

1. Derivare: Testul Junit s-a facut pe polinoamele p1 = x^2 + x + 2

si se asteapta raspunsul rez = 2x + 1

1. Derivare: Testul Junit s-a facut pe polinoamele p1 = x^2 + x + 2

si se asteapta raspunsul rez = 1/3x^3 + 1/2x\*2 + 2x

***6.Concluzie***

Concluzionand, aceasta tema a avut un rol important in solidificarea cunostintelor mele de java, dupa aceasta simtindu-ma mai stapan pe sine in lucrul cu clase si pachete de date. De asemenea subiectul abordat mi-a reimprospatat cunostinte esentiale unui student la calculatoare din domeniul matematicii, recapituland operatiile de baza pe polinoame

Aplicatia inca ar avea loc de imbunatatiri, putand adauga operatii de:

* Aflare a radacinilor unui polinom cand el este egal cu 0
* Rescrierea unui poliniom ca inmultire de alte polinoame

***7.Bibliografie***

- <https://www.w3schools.com/java/java_regex.asp>