DOCUMENTATIE

TEMA *1*

NUME STUDENT: …Manea Teodora………………

GRUPA: …1………………

# 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

*Se vor prezenta: (i) obiectivul principal al temei printr-o fraza, si (ii) obiectivele secundare sub forma tabelara sau ca lista. Obiectivele secundare reprezinta pasii care trebuie urmati pentru indeplinirea obiectivului principal. Fiecare obiectiv secundar va fi descris si se va indica in care capitol al documentatiei va fi detaliat.*

Obiectivul principal al proiectului este de a crea un calculator polynomial, utilizand Java Swing pentru interfata, care sa permita utilizatorului sa efectueze operatii matematice cu polinoame.

Obiectivele secundare

* Utilizarea designului de programare orientat pe obiect (OOP), care sa includa incapsularea si definirea claselor adecvate, cum ar fi clasa Polynomial si clasa Monomial.
* Implementarea interfetei grafice utilizand Java Swing
* Implementarea operatiilor de adunare, scadere, inmultire, impartire, integrare si derivare a doua polinoame
* Folosirea interfetei Map in scopul gestionarii polinoamelor
* Implementarea functiei de afisare (toString) a polinoamelor in interfata si de salvare a polinoamelor din interfata in obiecte de tip Polinom

# Analiza problemei, modelare, scenarii, cazuri de utilizare

*Se va prezenta cadrul de cerinte functionale/ non functionale si cazurile de utilizare ca si diagrame si descrieri de use-case. Descrierile use-case-urilor se vor face sub forma unui flow-chart ori sub forma unei liste continand pasii executiei specifici fiecarui use-case.*

Cerinte Functionale

* Să permită utilizatorului să introducă (de la tastatura) polinoame cu coeficienți întregi
* Să ofere operații aritmetice cu polinoame (adunare, scădere, înmulțire, împărțire cu rest, derivare, integrare).
* Să afișeze rezultatul fiecărei operații într-un format ușor de citit.

Cerinte non-functionale

* Interfața grafică trebuie să fie intuitivă și ușor de utilizat.
* Calculatorul polinomial trebuie să funcționeze rapid și să poată procesa polinoamele cu coeficienți mari și cu grade mari.
* Interfața grafică trebuie să fie estetică și să ofere utilizatorului o experiență plăcută de utilizare.

Cazurile de utilizare

* Adunarea a doua polinoame
* Scaderea a doua polinoame
* Inmultirea a doua polinoame
* Impartirea a doua polinoame
* Derivarea unui polinom
* Integrarea unui polinom

Descrierea cazului de utilizare

Caz de utilizare: Calcularea unei operații polinomiale

Actor principal: Utilizatorul

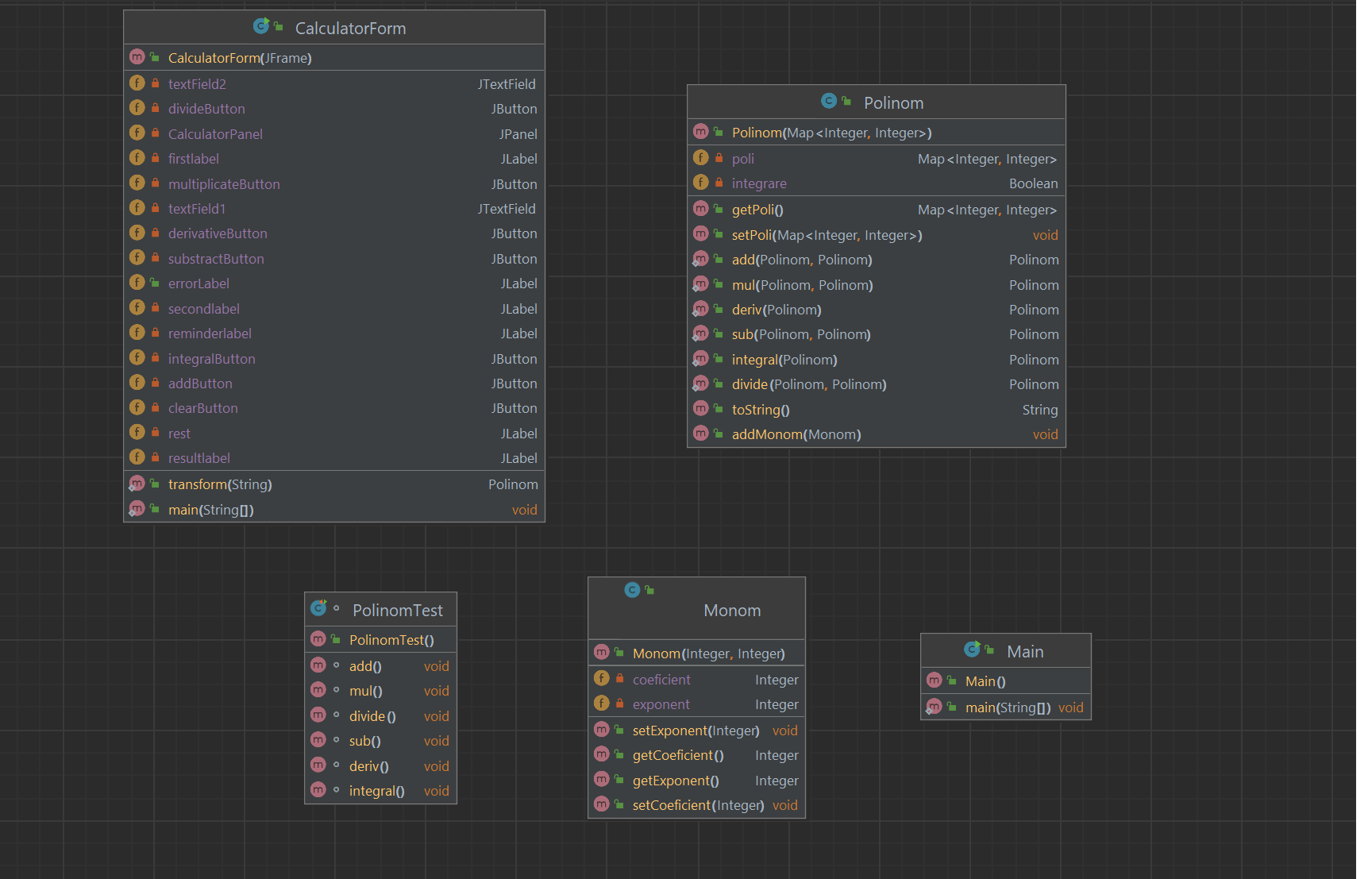
Pași de scenariu de succes:

* Utilizatorul introduce două (sau unul) polinoame în câmpurile de text specificate.
* Aplicația verifică dacă intrarea are formatul corect și o transpune în obiecte de tip Polinom.
* Utilizatorul selectează operația dorită.
* Aplicația calculează rezultatul.
* Rezultatul este afișat în câmpul de text specificat.
* Utilizatorul poate apasa pe butonul de clear ca sa reseteze campurile de text, de result si eventual de remainder (in cazul efectuarii unei impartiri) si poate introduce alte polinoame sau poate schimba operația.

# Proiectare

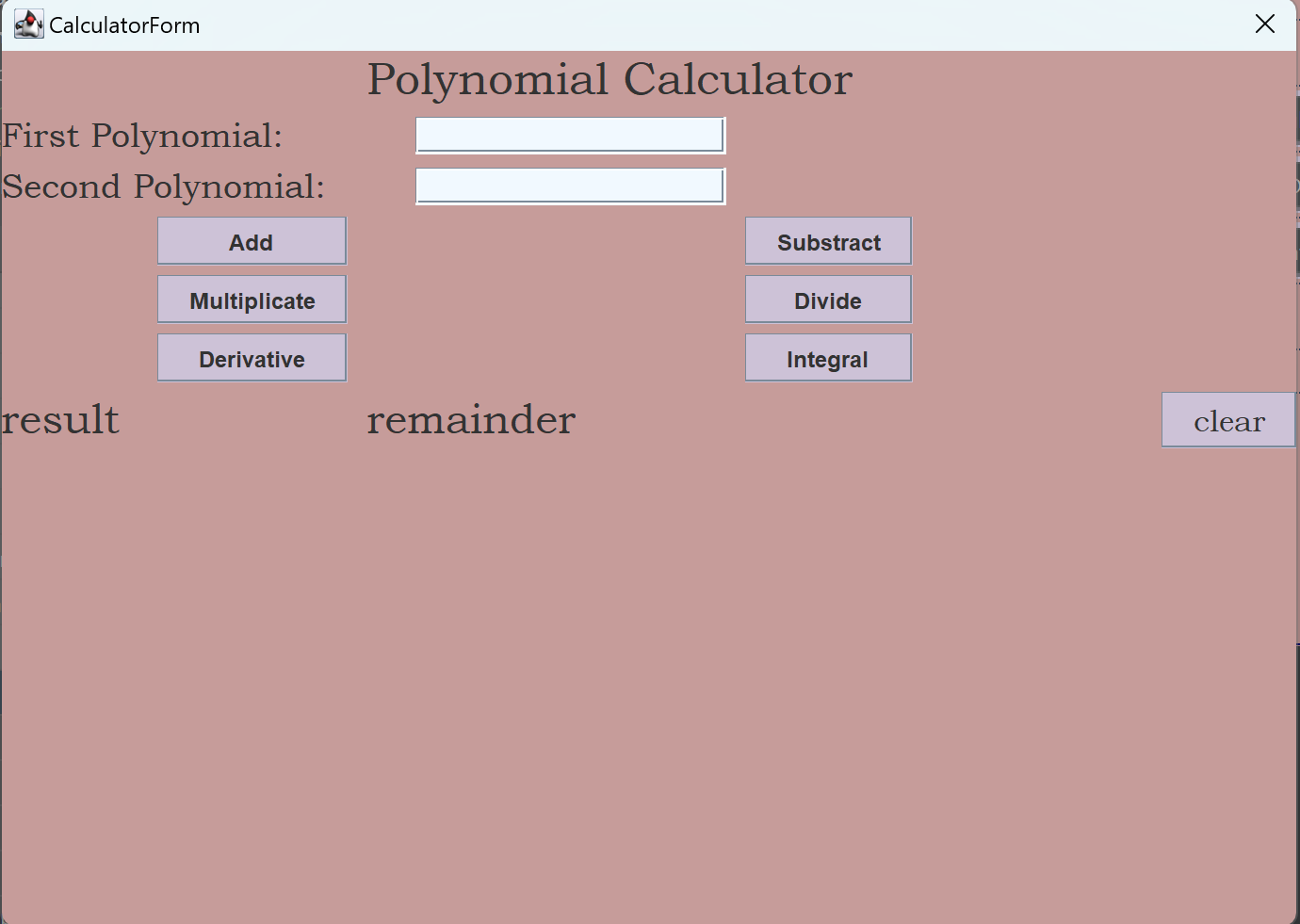
*Se va prezenta proiectarea OOP a aplicatiei, diagramele UML de clase si de pachete, structurile de date folosite, interfetele definite si algoritmii folositi (*daca e cazul)

Diagrama UML a claselor:



Această aplicație folosește atât tipuri de date primitive, cum ar fi întregi sau de tip dublu, șiruri, cât și tipuri de date mai complexe, cum ar fi HashMap. Este eficient să folosim HashMap în cazul construirii de polinoame deoarece acesta oferă o căutare foarte rapidă a perechilor de cheie-valoare. În cazul polinoamelor, cheia poate fi exponentul termenului, iar valoarea poate fi coeficientul termenului respectiv. De asemenea, utilizarea HashMap poate reduce complexitatea timpului și spațiului, deoarece permite accesarea directă la perechi de cheie-valoare, fără a fi nevoie să se parcurgă în mod repetat întregul polinom pentru a găsi un anumit termen.

Aspect interfata grafica:



# Implementare

*Se va descrie fiecare clasa cu campuri si metodele importante. Se va descrie implementarea interfetei utilizator.*

Clasa monom

Clasa monom are campurile coeficient si exponent si folosesc instantierea de obiecte de tip monom in cadrul clasei Polinom la definitia metodei addMonom() pentru adaugarea unui monom la un polinom.

Clasa Polinom

In clasa Polinom am implementat toate metodele cu operatii pe polinoame: add(), sub(), mul(), deriv(), integral(), divide() si metoda toString() care imi printeaza un polinom in format de string si pe care o folosesc si in clasa interfetei CalculatorForm pentru afisarea rezultatelor operatiilor pe resultlabel respective pe remainderlabel.

Clasa CalculatorForm

Acesta este clasa în care am implementat interfața grafică a utilizatorului (GUI). Am ales să lucrez cu Java Swing deoarece consider că este mult mai ușor și mai simplu de înțeles decât alte generatori de interfețe grafice.

Ca si metode am metoda transform() care primeste ca input un string si care foloseste un regex care imi separa stringul in monoame, iar mai apoi parcurge fiecare monom si sustrage coeficientul si exponentul adaugandu-le monomului si in final polinomului.

In cadrul constructorului avem 6 metode care interpreteaza actiunile efectuate pe interfata grafica (GUI). In fiecare dintre aceste metode se instantiaza obiecte de tip Polinom care mai apoi sunt initializate cu polinomul rezultat de functia transform() si apoi trimise ca parametru functiilor de adunare, scadere etc. Rezulatul este afisat pe resultlabel, remainderlabel.

Clasa main

In clasa main am instantiat 3 obiecte de tip polinom si mi am testat metodele cu operatii inainte de a implementa interfata.

# Rezultate

*Se vor prezenta scenariile pentru testare. In cazul in care in cerinta temei se specifica sa se faca testare unitara cu utilitarul JUnit, se vor integra in acesta sectiune rezultate testari cu JUnit.*

Pentru a testa această aplicație am folosit JUnit. Am creat o clasă nouă numită "PolinomTest" în care am simulat operațiile implementate pentru un polinom. Condiția este verificată folosind API-ul "assertEquals" din Junit.

* Adunare

Polinom1: 4x^3 - 2x + 5

Polinom2: 2x^2 + 2x

Result: + 5+ 2x^2+ 4x^3

Scadere

* Polinom1: 2x+56

Polinom2: 54x^2+190

Result: -134+2x-54x^2

* Inmultire

Polinom1: x+9

Polinom2: x

Result: +9x+x^2

* Impartire

Polinom1: x^3-2x^2+6x-5

Polinom2: x-2

Result: +1+x^2

* Derivare

Polinom1: 9x^2-14x-70

Result: -14+18x

* Integrare

Polinom1: 4x+3

Result: +3x+1/2\*4x^2

# Concluzii

*Se vor prezenta concluziile, ce s-a invatat din tema, posibile de dezvoltari ulterioare.*

În final, acest proiect m-a ajutat să-mi îmbunătățesc cunoștințele despre programarea orientată pe obiecte și abilitatea de a gestiona diferite structuri de date.Pentru a dezvolta un calculator polinomial funcțional, este necesară o cunoaștere temeinică a conceptelor matematice legate de polinoame, cum ar fi gradul și coeficienții acestora, precum și a operațiilor matematice care se pot efectua asupra acestora, precum adunarea, scăderea, înmulțirea și împărțirea. O bună înțelegere a acestor concepte și operații este esențială pentru a implementa și testa corespunzător funcționalitatea calculatorului polinomial. În plus, dezvoltarea ulterioară a calculatorului polinomial poate implica adăugarea de funcționalități suplimentare, cum ar fi integrarea și derivatele polinoamelor, ceea ce necesită o cunoaștere avansată a matematicii.

# 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)
3. <http://www.stackoverflow.com/>
4. <http://www.wikipedia.org>
5. <https://www.draw.io>