Documentatie

**Tema 1**

**Nume Student: Hotupan Rares-Florin**

**Grupa: 30221**

**Cuprins**

[1.Obiectivul Temei 3](#_Toc97471449)

[2.Analiza problemei, modelare, cazuri de utilizare 3](#_Toc97471450)

[3.Proiectare 3](#_Toc97471451)

[4.Implementare 3](#_Toc97471452)

[4.1 Clasa Monomial 3](#_Toc97471453)

[4.2 Clasa Polinomial 4](#_Toc97471454)

[4.3 Clasa GUI 5](#_Toc97471455)

[4.4 Clasa Main 6](#_Toc97471456)

[5.Concluzii 6](#_Toc97471457)

# 1.Obiectivul Temei

Obiectivul principal al acestei teme este realizarea unui calculator polynomial cu operatiile de baza adunarea si scaderea si ca operatii suplimentare inmultirea, impartirea, integrarea si derivarea. Pentru realizarea acestui calculator sunt necesare urmatoarele

|  |  |
| --- | --- |
| Clasa de monom | cap |
| Clasa de polinom | cap |
| Clasa de GUI | cap |

Clasa de monom ne permite construirea unui monom cu grad si coefficient, clasa de polinom ne permite construirea unei liste de monoame iar clasa de GUI creeaza interfata grafica si o leaga de partea de calcul.

# 2.Analiza problemei, modelare, cazuri de utilizare

Calculatorul trebuie sa indeplineasca urmatoarele cerinte functionale:

Citirea polinoamelor din String-uri

Adunarea a doua polinoame

Scaderea a doua polinoame

Inmultirea a doua polinoame

Integrarea unui polinom

Derivarea unui polinom

# 3.Proiectare

Clasa de baza este clasa monom, pe aceasta este construita clasa polinom iar pe aceasta se construieste clasa GUI

# 4.Implementare

## 4.1 Clasa Monomial

Clasa monomial are campurile grad, care reprezinta gradul unui monom,si coefficient,care reprezinta coeficientul unui monom. De asemenea clasa detine un constructor Monom(int,double) care atribuie valoarea int-ului gradului iar valoarea double-ului coeficientului monomului. Aceasta clasa are 2 setteri si doi getteri, un setter pentru grad si un getter pentru grad si un setter pentru coefficient si un getter pentru coefficient.

## 4.2 Clasa Polinomial

Clasa polinomial are un singur camp, o lista de tip ArrayList(P), inclusa in extensia java.util.\*, de tip Monomial, practic datele sunt structurate ca un sir de monoame. Clasa face uz de doi constructori, unul simplu Polinomial() care creeaza o lista de Monoame nula, si unul complex Polinomial(ArrayList <Monomial>) care creeaza o noua lista cu elementele listei trimise ca parametru. Clasa detine o multitudine de metode care au diverse functionalitati, metoda SetArrayList( ArrayList<Monomial> ) care initializeaza lista P cu valorile trimise prin parametru. Metoda error(int) arunca o exceptie cand parametrul trimis este 0. Metoda check (char,char) careia I se trimit 2 caractere consecutive si verifica daca respecta modelul “necunoscuta^”. Metoda SecondCheck(int,char) continua verificarea pentru cazul particular in care avem un termen liber, deci acesta nu este urmat de modelul “necunoscuta^” si nici de o variabila de tip int care sa ne indice puterea aceasta fiind 0. Errorcheck( int, char, char ) face verificarile check si secondcheck si returneaza o valoare pentru ok in functie de caz: 1 pentru monom correct citit, 2 pentru termen liber si 0 pentru eroare la citire. Metoda SetArrayFromString face conversia dintr-un string la un polinom prin intermediul converisiei la multiple monoame. Aceasta metoda converteste din string mai intai un double sau semn ,dupa caz(primul element), aceasta transformare are si ea la randul ei doua cazuri: cand numarul este pozitiv si cand acesta este negativ. Dupa convertirea din string a coeficientului de tip double acesta este salvat intr-o variabila si se verifica ce urmeaza cu metoda Errorcheck daca metoda returneaza 2 atunci coeficientul gasit este termen liber si puterea acestuia este 0, ca urmare se adauga in sirul de monoame un monom cu coeficientul salvat si gradul 0. Daca metoda returneaza 1 atunci avem un monom si trebuie verificata corectudinea codului se verifica daca urmeaza o variablia de tip litera, este permisa orice litera dar va fii tratata tot ca un x si dupa aceasta trebuie sa urmeze un semn “^” care indica ridicarea la putere, daca si aceasta conditie a fost indeplinita putem incepe convertirea din string a numarului de tip int care reprezinta gradul polinomului. Din nou se verifica daca am primit un numar valid si daca da se adauga un monom cu coeficientul salvat anterior si gradul calculat recent. Mai exista un caz particular cand coeficientul lui x este 1 si acesta nu este specificat la input i.e. “x^2”, acest caz este tratat si coeficientul este setat la 1 si metoda continua normal.

Metoda toString face override la metoda standard din java si transforma lista de polinoame intr-un string pentru a putea fii afisat. String-ul este initializat cu primul element care este plasat fara semn si dupa aceea se parcurge lista pentru a se plasa elementele ramase in aceasta.

Metoda de suma Sum face suma a doua polinoame si returneaza un al treilea polinom in care sa afla aceasta. Aceasta metoda parcurge in parallel cele doua polinoame si compara gradele elementelor in cazul gradurilor egale acestea sunt adaugate in polinomul rezultat cu coeficientii adunati si sunt sterse din polinomoamele intitiale pentru a modela polinoamele pentru pasul 2, la pasul 2 se iau ramasitele celor doua polinoame si se adauga la polinomul rezultat, ramasitele nu au grade comune conform verificarilor de la pasul 1 si ca urmare se adauga fara nici o verificare, acestea nu mai sunt sterse din polinoamele intiale generand niste date incomplete lasate in memorie, cu toate acestea timpul de executare este sporit de aceasta omitere iar acele date nu pot intervenii niciunde altundeva in program.

Metoda de diferenta Difference asemenea metodei de suma este o metoda static care returneaza un polinom rezultat construit din monoame care reprezinta scadrea primului polinom la cel de al doilea. Structura acestei metode este de asemenea pe 2 pasi primul pas in care identificam elementele cu grad comun din cele doua polinoame si adaugam in sirul rezulat un monom cu coeficient egal cu diferenta celor doua si gradul egal cu gradul comun si elementele sunt sterse din polinoamele corespunzatoare. La pasul doi se aplica o abordare similara cu metoda de adunare in care ramasitele din cele doua polinoame sunt transpuse pe polinomul rezultat. Elementele din primul polinom sunt adaugate asa cum sunt iar cele din cel de al doilea polinom sunt adaugate cu – in fata pentru a simula o scadere. Aceasta metoda similar cu punctual 1 lasa 2 variabile incomplete in memorie dar cu toate acestea este mult mai rapida decat o metoda care ar sterge datele complet.

Metoda de integrare Integrate este o metoda mult mai simplu de implementat din cauza matematicii aplicate in spate, formula ne spune ca integrala unei sume este egala cu suma integralei, ceea ce inseamna ca putem aplica integrala in fiecare membru, ceea ce si face metoda: parcurge lista cu un iterator si creste toate gradele cu 1 si imaprte coeficienti cu gradele crescute, precum formula integrarii pentru x^p=1/(p+1) \* x^(p+1).

Metoda de derivare Derivate este o metoda similara cu cea de integrare care din nou se simplifica foarte mult din cauza formulei din spate, derivate sumei este egala cu suma derivatelor, deci putem deriva element cu element, parcurgem lista cu un iterator , scadem gradul fiecarui element si inmmultim coeficientul cu gradul dinainte de modificare, formula provenita din formula de derivare (x^ p)’=p\* x^ (p-1)

Metoda de inmultire Multiply face inmultirea a doua polinoame, inmultind element cu element. Problema la aceasta metoda este surplusul de polinoame cu acelasi grad iar pentru rezolvarea acestei metode este nevoie de o parcurgere in plus a listei rezulat si adunarea coeficientilor polinoamelor cu acelasi grad la primul de acest tip si stregerea lor din lista, din pacata aceasta rezolvare exclude iteratori din pricina necesitatii unei bucle in bucla pe o lista. Folosirea iteratorilor produce o eroare de tip “Concurent Modification Exception”.

Mai exista o metoda de GetArray() care produce array-ul de monoame, acesta fiind privat.

## 4.3 Clasa GUI

Clasa GUI implementeaza interfata grafica cat si face legatura intre parea vizuala si operatiile executate in cod.Aceasta clasa extinde clasa standard din java ActionListener pentru a se putea verifica apasarea unor butoane. Interfata grafica este formata din sapte butoane, patru sunt pentru setarea operatiei la adunare, scadere, inmultire,sau impartire, unul, egal, este pentru a afisa rezultatul in urma operatiei, si celelalte doua pentru a integra sau deriva un polinom. Butoanele de derivare si integrare nu necesita apasarea butonului egal. De asemenea interfata are trei TextField-uri. In primul textfield se insereaza primul polinom, in cel de al doilea al doilea polinom, iar al treilea textField prezinta rezultatul dupa operatie. Pe interfata mai apar 4 labeluri, 3 dintre acestea indica uzul textboxurilor(“Polinom 1”, “Polinom 2”, “Polinom rezultat”) iar cel de al patru-lea indica operatia curenta. Totate butoanele au implementat cate un action listener pentru a implementa operatia corespunzatoare.

Interfata grafica este implementata cu ajutorul librariilor swing si awt, interfata principala este formata dintr-un Jframe, element de swing dar pentru folosirea actionListenerilor, cand verificam apasarea unui buton, avem nevoie sa importam bibleoteca awt.event.\* . In rest bibleoteca javax.swing.\* este mai mult ca suficienta pentru a genera interfata necesara acestui proiect.

Constructorul acestei clase intializeaza atat fereastra cat si locatiile butoanelor, label-ulilor,cat si a textfield-urilor, si dimensile acestora. Fiecare buton are doua cordonate , cele ale coltului stanga sus, doua dimensiune lungime si latime si un action listener, acestea sunt adaugate, dupa ce sunt initializate la un panel care este plasat pe frame. Label-urile la randul lor au nevoie de doua coordonate, ale coltului din stanga sus, si de doua valori care simbolizeaza o lungime si o latime, acestea trebuie la randul lor adaugate pe panelul care este situat pe frame. In final textfield-urile au la fel doua coordonate care semnifica o lungime si o latime si doua valori care reprezinta cati pixeli sa mearga in jos si cati pixeli sa mearga in dreapta textfield-ul. La randul lor text-fieldurile trebuie sa fie adaugate la panel care la randul lui a fost adaugat la frame. La finalul constructorului trebuie facut vizibil frame-ul, nu la inceput, pentru a deveni vizibile toate elementele adaugate la panel. Si de asemenea trebuie mentionat ca content panelul este panelul pe care sunt adaugate toate elementele mentionate mai sus.

Clasa are la dispozitie doua gettere de tip string care procura valorile sub forma de string a celor doua polinoame care sunt tratate ca intrari in program, si un setter care transpune un string in cel de al treilea textfield.

In continuare trebuie facut un override la functia standard java actionPerformed pentru a verifica daca un buton este apasat. Fiecarui buton ii este atribuit cate un cod specific: plus 1, minus 2, inmultit 3, impartit 4, derivare 5, integrare 6 , egal 7. Pe baza acestui cod se seteaza operatia curenta afisata pe label. Cand se seteaza codul pentru integrare si derivare se si apeleaza metoda pentru egal pentru a nu a mai fii necesara apasarea egalului.

Metoda doMinus (polinomial, polinomial) face diferenta a doua polinoame prin intermediul metodei statice incluse in clasa polinomial. Polinomul returnat este convertit la string tot prin intermediul unei metode descrise in clasa polinomial, toString si rezutatul este plasat in textfieldul rezultat.

Metoda doSum (polinomial, polinomial) face suma a doua polinoame prin intermediul metodei statice incluse in clasa polinomial. Polinomul returnat este convertit la string tot prin intermediul unei metode descrise in clasa polinomial, toString si rezutatul este plasat in textfieldul rezultat.

Metoda doMultiply (polinomial, polinomial) face inmultirea a doua polinoame prin intermediul metodei statice incluse in clasa polinomial. Polinomul returnat este convertit la string tot prin intermediul unei metode descrise in clasa polinomial, toString si rezutatul este plasat in textfieldul rezultat.

Metoda doIntegrate (polinomial) Integreaza un polinoam prin intermediul metodei statice incluse in clasa polinomial. Polinomul returnat este convertit la string tot prin intermediul unei metode descrise in clasa polinomial, toString si rezutatul este plasat in textfieldul rezultat.

Metoda doDerivate (polinomial) Integreaza un polinoam prin intermediul metodei statice incluse in clasa polinomial. Polinomul returnat este convertit la string tot prin intermediul unei metode descrise in clasa polinomial, toString si rezutatul este plasat in textfieldul rezultat.

## 4.4 Clasa Main

Clasa main este o clasa foarte simpla tocmai din cauza proiectului format modular, impartit pe clase, in aceasta clasa se declara o interfata grafica de tip GUI si se apeleaza constructorul acesteia pentru a creea interfata cu toate elementele necesare.

# 5.Concluzii

In Concluzie acest proiect vizeaza creearea unui calculator de polinoame, cu operatii de baza. Acest proiect este foarte bun pentru inceput deoarece are atat cerinte de baza in ceea ce priveste progamarea orientata pe obiecte cat si cerinte pentru cei avansati. Cheia proiectului sta in combinarea de parte aritmetica cu cea de design a interfeteii grafice.

**6.Bibliografie**

<https://www.youtube.com/watch?v=Yat8l37XGFA>

https://dsrl.eu/courses/pt/materials/PT2021-2022\_Assignment\_1.pdf