

**FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE**

**CATEDRA CALCULATOARE**

Tehnici de programare

Documentație

**Calculator de polinoame**

Bartalus Izabella

Grupa 30221

**Cuprins**

1. Obiectivul temei

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

3. Proiectare (decizii de proiectare, diagrame UML, structuri de date, proiectare clase, interfete , relatii, packages, algoritmi, interfata utilizator)

4. Implementare

5. Rezultate

6. Concluzii

7. Bibliografie

1. Obiectivul temei

Obiectivul acestei teme este reealizarea unui calculator de polinoame . Calculatorul de polinoame poate realiza urmatoarele operatii de baza : adunarea a doua polinoame , scaderea a doua polinoame , inmultirea a doua polinoame , impartirea a doua polinoame , derivarea unui polinom si integrarea unui polinom. Fiecare dintre aceste polinoame vor fi introduse de catre utilizator din interfata grafica , de unde acesta poate alege operatia pe care doreste sa o efectueze pentru cele doua polinoame introduse anterior . De asemenea , rezulatul operatiei alese de utilizator va fi afisat in interfata grafica .

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

* Analiza:

Un polinom reprezinta o expresie formata din mai multe monoame care , la randul lor , sunt construite cu ajutorul unor coeficienti si exponenti. Cel mai important lucru in acest proiect este implementarea operatiilor polinomului si pentru acest lucru trebuie sa folosim cativa algoritmi ( adunare , scadere , inmultire , impartire , derivare , integrare).

* Modelare:

Pentru a putea realiza cele 6 operatii care au fost propuse, este necesar in primul rand sa gasim o metoda de a lucra mai concret cu un polinom care, atunci cand este introdus de la tastatura este mai intai un string, dar pentru a-l adapta problemei noastre il vom transforma intr-o lista de monoame, fiecare monom fiind alcatuit dintr-un coeficient si un exponent.

* Scenarii:

Pentru ca programul sa ruleze corect, exista un set de reguli pe care utilizatorul trebuie sa il indeplineasca atunci cand introduce un polinom:

* Necunoscuta tuturor polinoamelor va fi ‘x’ ;
* Fiecare termen ( monom ) va fi de tipul : ‘coeficientx^exponent’ ;
* Termenii vor fi delimitati de semnele ‘+’ sau ‘-‘ ;
* Primul termen al polinomului va fi intodeauna pozitiv ;
* Nu se va pune spatiu intre termenii polinomului , sau intre semnele ‘ + ‘ sau ‘ – ‘ .
* Cazuri de utilizare:

Atunci cand utilizatorul apasa butonul de ‘Run’ se deschide interfata grafica. Aceasta interfata este una simpla si usor de utilizat. Este alcatuita din 6 butoane aferente fiecare operatii ( ‘Addition’ , ‘Substraction’ , ‘Multiplication’, ‘Division’ , ‘ Derivation’ , ‘Integration’ ) si 3 casete de text. Primele 2 casete de text deasupra carora scrie ‘First polynomial:’ si ‘Second Polynomial:’ sunt destinate introducerii celor 2 polinoame asupra carora se vor realiza operatiile iar cea de a treia caseta de text deasupra careia scrie ‘ Result:’ este destinata afisarii rezultatului unei anumite operatii. Un avantaj al acestui calculator de polinoame este faptul ca utilizatorul poate efectua mai multe tipuri de operatii asupra acelorasi polinoame fara a fi nevoit sa le rescrie .

1. Proiectarea ( decizii de proiectare , diagrama UML , structuri de date , proiectare clase , interfete , relatii , packages , algoritmi , interfata utilizator )

Pentru a respecta paradigmele POO, programul a fost impartit in mai multe pachete cu nume sugestive care contin diferite clase :

* Pachetul execution contine clasa App ;
* Pachetul monomial contine clasele Monimial si MyComp ;
* Pachetul mvc contine clasa CalculatorView ;
* Pachetul polynomial contine clasele Polynomial si PatternMatching ;
* Pachetul tests contine clasa Test .

Graphical user interface, text, application, chat or text message

Description automatically generated

Diagrama UML a proiectului :

Diagram, schematic

Description automatically generated

Interfata grafica cuprinde urmatoarele elemente:

* Frame *=* “rama” in care se adauga toate elementele de care avem nevoie pentru buna functionare a programului . E practice o fereastra care , dupa utilizare , apasand butonul ‘X’ din dreapta sus se poate inchide ( Exit\_on\_close ) ;
* Butoane – avem 6 butoane fiecare cu alta functionalitate :
* Addition – pentru a face suma a doua polinoame ;
* Substraction – pentru a face scaderea a doua polinoame ;
* Multiplication – pentru a face inmultirea a doua polinoame ;
* Division – pentru a face impartirea a doua polinome ; --NU FUNCTIONEAZA!!
* Derivation – pentru a face derivarea asupra primului polinom ;
* Integration – pentru a face integrarea asupra primului polinom ;
* TextField = spatii dreptunghiulare in care se pot introduce date de la tastatura , dar pe langa asta pot fi folosite si pentru a afisa rezultatul fara a se putea introduce date de la tastatura . In proiectul prezentat exista urmatoarele textField – uri :
* ‘ First polynomial: ‘ – pentru introducerea primului polinom ;
* ‘ Second polynomial: ’ – pentru intoducerea celui de al doilea polinom;
* ‘ Result ‘ – caseta de text unde se afiseaza rezultatele aferente ;
* Label = o etichetă , care poate fi titlul , informații , indicații etc . , care pot ajuta utilizatorul să folosească progamul. In cazul nostru e folosit doar pentru afisarea informatiilor aflate deasupra casetelor de text : ‘ First polynomial: ‘ , ‘ Second polynomial: ‘ , ‘ Result: ‘ .

1. Implementare

In cadrul proiectului au fost implementate 5 din cele 6 operatii cerute si anume: adunarea a doua polinoame , scaderea a doua polinoame , inmultirea a doua polinoame , derivarea unui polinom si integrarea unui polinom .

Clasele existente sunt:

* Clasa App unde este instantiata interfata grafica si de unde se ruleaza aplicatia ;
* Clasa MyComp - care implementeaza interfata Comparator pt a realiza comparatii intre monoame ;
* Clasa Monom – contine atributele unui monom: coefficient ( pt coeficientul unui monom ) , degree ( pt putere ) si verifyOp ( folosit la operatia de adunare pt o verificare mai detaliata ). Tot aici se afla si operatiile care pot fi realizate asupra unui monom :
* Adunarea : am declarat variabila result pentru a stoca rezultatul ( aceasta fiind un monom ) , iar pentru a obtine coeficientul , am adunat variabila de intrare m cu coeficientul referinta la this , puterea ramanand neschimbata, urmand sa returnam variabila result ;

Text

Description automatically generated

* Scaderea : am declarat variabila result pentru a stoca rezultatul ( aceasta fiind un monom ) , iar pentru a obtine coeficientul , am scazut din variabila de intrare m , coeficientul referinta la this , puterea ramanand neschimbata , urmand sa returnam variabila result;

Text

Description automatically generated

* Inmultirea : am declarat variabila result pt a stoca rezultatul ( aceasta fiind un monom ) , iar pentru a obtine coeficientul, am inmultit variabila de intrare m cu coeficientul referinta la this, puterile acestora le-am adunat , urmand sa returnam variabila result;

Text

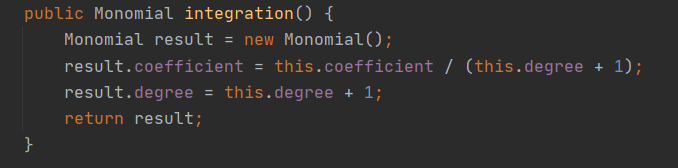
Description automatically generated

* Impartirea : am declarat variabila result pt a stoca rezultatul ( aceasta fiind un monom ) , iar pentru a obtine coeficientul, am impartit la coeficientul referinta la this , coeficientul monomului m de la intare, iar puterile acestora le-am scazut , urmand sa returnam variabila result ;

Text

Description automatically generated

* Integrarea: am aplicat formula matematica : (x^n) = ( 1/(n+1 ) \* x^(n+1)



* Derivarea : am tratat cazurile speciale iar pentru cazul normal am aplicat formula matematica : ( x^n ) ’ = n \* x^(n-1)

Text

Description automatically generated

Tot in interiorul clasei Monom sunt si metodele folosite la afisarea monoamelor , metoda format() fiind folosita la afisarea oricarui monom iar metoda format2() folosita la afisarea rezultatului unui monom integrat .

Text

Description automatically generated

* Clasa CalculatorView unde am scris codul pentru realizarea interfetei grafice, cod care cuprinde : modul in care am aranjat frame-ul , etichetele , butoanele , casetele text , atribuindu-le tututor niste coordonate , pentru o vizibilitate mai buna;
* Clasa PatternMatching – aici am realizat impartirea polinomului ( care este initial un string ) in monoame , fiecare polinom fiind astfel privit ca o lista de monoame. Metoda getExp ( String a ) afla exponentul din stringul de la intrare si metoda getCoeff ( string m ) aflta coeficientul stringului de la intrare.

Text

Description automatically generated

Folosind aceste doua metode, am creat metoda getMonoms ( String polynom ) care primeste la intrare un polinom sub forma de string si care este transformat intr-o lista de monoame.

Text

Description automatically generated

* Polynomial – care contine atributele unui polinom : monomialsList ( o lista de monoame ) , operatiile care se pot realiza asupra unui polinom , metodele folosite pentru afisarea unui polinom si o metoda de sortare. Operatiile implementate care pot fi realizare asupra unui polinom sunt :
* Adunarea : am declarant o variabila result ( aceasta fiind tot un polinom ) , primesc ca si parametrii de intrare doua polinoame, se sorteaza si se parcurg . Atunci cand intalnesc doua monoame cu acelasi grad ( monoamele fac parte din lista de polinoame ) verific daca acestea nu au fost adaugate in polinomul result, iar caz pozitiv ( verifyOp = ) fac suma acelor monoame folosind metoda sum din clasa Monomial , iar rezultatul adunarii il adaug in result ; la sfarsit mai fac o trecere prin ambele polinoame si adaug monoamele care nu au fost deja adaugate ; la final returnez variabila result;
* Scaderea : am declarant o variabila result (aceasta fiind tot un polinom ) , primesc ca si parametrii de intrare doua polinoame, inversez semnele de la al doilea polinom ( inmultesc coeficientii cu -1 ) si apoi adun cele doua polinoame; la final returnez variabila result ;
* Inmultirea : am declarant o variabila result (aceasta fiind tot un polinom ) , primesc ca si parametrii de intrare doua polinoame , le parcurg si inmultesc ( folosind metoda de inmultire a doua monoame multiply din clasa Monomial ) fiecare termen din primul polinom cu fiecare termen din al doilea polinom ( un termen este un monom ) si adaug rezultatul fiecarei inmultiri in result; la final returnez variabila result ;
* Derivarea : am declarant o variabila result (aceasta fiind tot un polinom) si parcurg polinomul referinta this care este de fapt o lista de monoame, iar asupra fiecarui monom aplic operatia de derivarea asupra unui monom (din clasa Monomial) , rezultatul il adaug in polinomul result iar la final returnez variabila result ;
* Integrarea : am declarant o variabila result ( aceasta fiind tot un polinom ) si parcurg polinomul referinta this care este de fapt o lista de monoame, iar asupra fiecarui monom aplic operatia de integrare a unui monom ( din clasa Monomial ) , rezultatul il adaug in polinomul result iar la final returnez variabila result ;
* Clasa Test unde am facut teste pentru :
* Adunarea a doua monoame ( am initializat doua monoame si am calculat suma lor , dar am iniliatizat si un monom care este de fapt suma primelor doua iar cu assertEquals am verificat daca suma declarata de mine este aceeasi cu suma obtinuta in urma aplicarii metodei de insumare a doua monoame ) :

Text

Description automatically generated

* adunarea a doua polinoame ;
* scaderea a doua polinoame ;
* inmultirea a doua polinoame ;
* derivarea unui polinom si integrarea unui polinom .

1. Rezultate

Pentru a exemplifica rezulatele obtinute am introdus in interfata , in cele doua casete de text destinate introducerii polinoamelor , urmatoarele doua polinoame:

* First polynomial : 4 \*x^3 – 2 \* x^1 -1 \* x^0
* Second polynomial : 12 \* x^3 – 10 \* x^2 -10 \* x^1 + 1 \* x^0

Rezultatul obtinut in urma adunarii celor doua polinoame :

Graphical user interface, text, application

Description automatically generated

Rezultatul obtinut in urma scaderii celor doua polinoame :

Graphical user interface, text, application

Description automatically generated

Rezultatul obtinut in urma inmultirii celor doua polinoame :

A screenshot of a computer

Description automatically generated with medium confidence

Rezultatul obtinut in urma derivarii primului polinom :

A screenshot of a computer

Description automatically generated with medium confidence

Rezultatul obtinut in urma integrarii primului polinom :

A screenshot of a computer

Description automatically generated with medium confidence

Dupa cum se poate observa , toate rezultatele obtinute sunt corecte din punct de vedere matematic . De asemenea , toate rezultatele furnizate de aceasta aplicatie , in urma metodelor implementate au fost verificate si valiadate prin teste Junit ( pachetul Test ) .

1. Concluzii

In concluzie , sunt de parere ca acest proiect mi-a aprofundat cunostiintele in tot ce inseamna limbajul Java , implementarea paradigmelor POO , creearea unui program cu o interfata grafica . De asemenea , m-a ajutat sa imi reamintesc tehnicile de programare invatate in semestrul trecut .

Aplicatia realizata este o aplicatie simplista pentru realizarea operatiilor pe polinoame. Ca si imbunatatiri ulterioare putem adauga operatia de impartire a doua polinoame, o interfata mult mai complexa si mai elaborata si putem adauga chiar si alte tipuri de operatii asupra polinoamelor sau elaborarea acestora , fiind mai complexe.

1. Bibliografie
2. Wikipedia
3. StackOverflow