Neagu Tudor

Grupa 30222

Documentație tema 1TP

1.Obiectivul temei

Tema constă în implementarea unui calculator de polinoame, cu o interfață grafică prin care utilizatorul poate introduce polinoame, poate selecta operația care va fi efectuată și care îi va afișa rezultatul. Operațiile care vor fi implementate sunt : adunare, scădere, înmulțire, împărțire, derivare și integrare.

Coeficienții polinoamelor sunt considerați întregi.(La introducerea polinoamelor de către utilizator, coeficienții introduși trebuie să fie întregi, dar rezultatul operației va fi afișat având coeficienții numere reale).

Obiective secundare:

- analiza problemei, modelare, cazuri de utilizare( cap.2)

- proiectare (decizii de proiectare, diagrame UML, structuri de date, proiectare clase, relații, pachete, algoritmi, interfața grafică) (cap.3)

- implementare ( cap.4)

- rezultate (cap.5)

- concluzii( cap.6)

- bibliografie (cap.7)

2.Analiza problemei, cazuri de utilizare

Cerințele funcționale, după introducerea polinoamelor de intrare de către utilizator, sunt:

- realizarea adunării celor două polinoame introduse, dacă se apasă butonul + ;

- realizarea scăderii celor două polinoame introduse , dacă se apasă butonul - ;

- realizarea înmulțirii celor două polinoame introduse, dacă se apasă butonul \* ;

- realizarea împărțirii celor două polinoame introduse, dacă se apasă butonul / ;

- realizarea derivării polinomului P1 introdus de utilizator, dacă se apasă butonul DVT ;

- realizarea integrării polinomului P1 introdus de utilizator, dacă se apasă butonul ITG ;

Cazurile de utilizare care vor da rezultatul așteptat:

- introducerea a două polinoame valide, care respectă formatul indicat de nota informativă afișată de interfața-grafică-utilizator și apăsarea unuia din butoanele: +, -, \*, / ;

- introducerea unui polinom valid în câmpul P1, care respectă formatul indicat de nota informativă afișată de interfața-grafică-utilizator și apăsarea unuia din butoanele DVT sau ITG;

Pentru introducerea unor date eronate de către utilizator, dacă acestea vor conține subșiruri care au structura unui termen dintr-un polinom, acest termen va fi găsit și adăugat la polinomul construit ; altfel, aplicația va atenționa utilizatorul în legătură cu incorectitudinea datelor introduse;

3.Proiectare

Structura proiectului, așa cum este specificată în descrierea temei, respectă șablonul arhitectural Model View Controller. Astfel, pachetele în care sunt implementate clasele , sunt gândite în conformitate cu acest model:

- com.assignment1 conține clasa care va rula aplicația;

- com.assignment1.model va conține clasele ”Polynomial” și ”Monomial”, care reprezintă obiectele ce vor fi prelucrate de aplicație;

- com.assignment1.controller va conține clasa ”PolynomialController”, care va efectua operațiile specificate în cerință asupra polinoamelor ;

- com.assignment1.view va conține clasa ”AppWindow”, care implementează interfața grafică;

În vederea optimizării spațiului de stocare, dar și a timpului de execuție al aplicației , polinomul a fost implementat cu ajutorul unui HashMap în care sunt stocate monoamele: cheia este reprezentată de gradul monomului, deoarece într-un polinom nu vrem să avem doi termeni de același grad, iar coeficientul monomului reprezintă valoarea asociată cheii.

Algoritmii folosiți au fost sintetizați pe baza algoritmilor matematici care se utilizează pentru rezolvarea operațiilor specificate în cerință.

Interfața grafică utilizator a fost construită folosind JavaSwing.

Diagrama UML:

O imagine care conține text

Descriere generată automat

4.Implementare

Clasa Monomial:

- are două câmpuri private, reprezentând numere reale ( coeficientul monomului și gradul monomului )

- pentru cele două câmpuri specificate mai sus, au fost implementați getteri și setteri;

- această clasă mai conține două metode: una care returnează un monom nou care are gradul și coeficientul specificate prin parametrii și una care compară două monoame în funcție de grad (această metodă implementează metoda de comparare din interfața Comparable; a fost implementată pentru a putea utiliza Collections.sort)

O imagine care conține captură de ecran

Descriere generată automat

Clasa Polynomial:

- are un singur câmp, un HashMap care va conține monoamele din care este alcătuit polinomul(privat)

- getter și setter pentru câmpul respectiv

- metodele addTerm și addMonTerm, care adaugă la lista de termeni un termen nou, dat fie prin două valori care reprezintă coeficientul și gradul, fie printr-un obiect de tip monom; aceste metode sunt importante deoarece asigură faptul că polinomul nu va conține doi termeni diferiți de același grad; ele verifică gradul parametrului și daca in listă există deja un monom cu gradul respectiv, acestea vor aduna coeficientul parametrului;

- metoda refactor elimină termenii care au coeficientul 0 din polinom

- metoda getMaxGrade returnează monomul de grad maxim din polinom(sub forma de obiect Monomial)

O imagine care conține captură de ecran, așezat, telefon, celular

Descriere generată automat

O imagine care conține captură de ecran

Descriere generată automat

Clasa PolynomialController:

- nu are nici un câmp, conține metode statice care implementează operațiile care se efectuează asupra polinoamelor;

O imagine care conține captură de ecran, telefon

Descriere generată automat

- metoda add adună două polinoame a și b; rezultatul este returnat sub formă de polinom; această metoda parcurge pe rând listele de termeni ale celor două polinoame și adaugă fiecare termen în lista polinomului rezultat folosind metoda addTerm;

- metoda sub scade din polinomul n polinomul s; rezultatul este returnat sub formă de polinom; această metoda parcurge pe rând listele de termeni ale celor două polinoame și adaugă fiecare termen din polinomul n în lista polinomului rezultat folosind metoda addTerm; pentru polinomul s, termenii sunt adaugați tot cu ajutorul metodei addTerm, doar că toți coeficienții sunt înmulțiți cu -1;

O imagine care conține captură de ecran, telefon, celular, așezat

Descriere generată automat

- metoda print a fost implementată doar pentru verificări auxiliare și pentru testare, nu este folosită de aplicație; această metodă afișează polinomul în consolă;

- metoda prettyPrint, la fel ca și metoda print, a fost implementată doar pentru verificări auxiliare și pentru testare, nu este folosită de aplicație; această metodă afișează polinomul în consolă, eliminând termenii care au coeficientul 0; monoamele sunt afișate în ordine descrescătoare din punct de vedere al gradului; această metodă ia termenii polinomului din HashMap, îi introduce într-o listă de monoame și sortează descrescător lista, după care o va afișa în consolă;

- metoda retText returnează polinomul sub formă de string, pentru a putea fi afișat de interfața grafică; această metodă elimină termenii care au coeficientul 0 din polinom înainte să îl reprezinte sub formă de text; metoda ia termenii polinomului din HashMap, îi introduce într-o listă de monoame și sortează descrescător lista, după care lista va fi transformată în String;

O imagine care conține masă, ecran, laptop, computer

Descriere generată automat

- metoda mtp înmulțește două polinoame a și b, conform următoarei descrieri: se ia fiecare monom din a, se înmulțește pe rând cu fiecare monom din b, iar monoamele rezultate sunt adăugate în lista polinomului rezultat, cu ajutorul metodei addTerm; această metodă returnează rezultatul sub formă de polinom;

O imagine care conține captură de ecran, așezat, masă, telefon

Descriere generată automat

- metoda div împarte polinomul n la polinomul s și returnează câtul împărțirii; este creat un polinom nou care reprezintă rezultatul; de asemenea, se face o copie a polinomului n(deîmpărțitul) și se elimină zerourile din această copie; se memorează gradul maxim al polinomului s (împărțitorul); cât timp gradul maxim al copiei lu n este mai mare decât gradul maxim al lui s, se împarte monomul de grad maxim din copia lui n la monomul de grad maxim din s; rezultatul se adaugă la rezultat, apoi se înmulțește cu polinomul s și se scade din copia lui n;

se elimină zerourile din copia lui n după scădere și se repetă operația până când copia lui n are gradul mai mic decât s; la finalul buclei, în polinomul rezultat vom avea câtul împărțirii, iar în copia lui n vom avea restul; această metodă returnează polinomul rezultat;

O imagine care conține captură de ecran, telefon, așezat, monitor

Descriere generată automat

- metoda mod este face același lucru ca și metoda div, doar că returnează polinomul copie al lui n, care conține restul;

O imagine care conține captură de ecran

Descriere generată automat

- metoda dvt derivează polinomul a; derivarea se face termen cu termen, iar monoamele rezultate în urma derivării sunt stocate în lista de termeni a noului polinom, care va fi returnat;

- metoda itg integrează polinomul a; integrarea se face termen cu termen, iar monoamele rezultate în urma integrării sunt stocate în lista de termeni a noului polinom, care va fi returnat; de asemenea, în urma integrării se adaugă la rezultat un monom ce reprezintă constanta rezultată prin integrare. Această constantă a fost setată simbolic să aibă valoarea 0.1234;

O imagine care conține celular, telefon, așezat, masă

Descriere generată automat

- metoda mkFromString extrage dintr-un string dat ca parametru grupurile de simboluri care reprezintă termenii polinomului, îi transformă, îi adaugă în lista polinomului rezultat și returnează obiectul polinom rezultat. Pentru parsarea stringului se folosesc regular expressions. Formula aleasă este "([+-]?\d\*)x(\^(\d+))?|([+-]?\d+)" . Această formulă este avantajoasă deoarece coeficientul și puterea monomului vor fi preluate in grupuri separate, facilitând convertirea acestora din String în Double și astfel construirea polinomului rezultat.

Metoda caută într-o buclă match-urile din șirul de intrare și lucrează cu grupurile, astfel: dacă grupul 4 este nenul, însemnă că termenul dat este o constantă; se setează gradul ca fiind 0 și coeficientul (Double)grup(4); altfel, dacă grupul 4 e nul, vom verifica grupul 3; acesta reprezintă puterea; daca grupul 3 e nenul, monomul are gradul (Double)grup(3), altfel, monomul va avea gradul 1; mai departe, se verifică grupul 1, care reprezintă coeficientul în cazul în care grupul 4 e nul. Dacă grupul 1 e gol, coeficientul va fi 1; dacă grupul 1 e ”-”, coeficientul va fi -1; dacă grupul 1 e ”+”, coeficientul va fi 1, iar dacă nu e nici unul din cele specificate mai sus, coeficientul va avea valoarea (Double)grup(1). La final se adaugă termenul la lista polinomului și se merge mai departe la următorul substring găsit de obiectul Matcher;

Clasa App:

- conține metoda main; e folosită pentru lansarea ferestrei ApplicationWindow, care reprezintă interfața grafică utilizator;

O imagine care conține masă, așezat, laptop, computer

Descriere generată automat

Clasa ApplicationWindow:

- reprezintă o fereastră JavaSwing care contine 3 label uri, care indică utilizatorului în care text field să introducă polinoamele de intrare, 3 text field uri; în două dintre ele se vor introduce polinoamele de intrare, iar în al treilea va fi afișat polinomul rezultat; 1 textArea, care conține textul care informează utilizatorul despre cum trebuie să arate polinoamele și 5 butoane, câte unul pentru efectuarea fiecărei operații specificate în cerință;

- fiecare buton are un actionListener, în care este descrisă acțiunea efectuată la apăsarea butonului; de asemenea, la apăsarea fiecărui buton, sunt verificate datele introduse de către utilizator. Dacă aceste date nu sunt valide, se va afișa un OptionPane în care i se va aduce la cunoștință utilizatorului că nu a introdus datele corect;

- în cazul împărțirii, dacă la efectuarea operației polinoamele nu se împart exact, polinomul care reprezintă restul împărțirii va fi afișat într-un optionPane;

5.Rezultate

Aplicația întoarce rezultatul așteptat în orice scenariu. Pentru a asigura acest lucru, la transformarea datelor introduse de utilizator, s-au folosit regular expressions pentru identificarea termenilor polinoamelor. Dacă un text field este lăsat gol și utilizatorul încearcă sa efectueze o operație, îi va fi afișat un mesaj, iar operația respectivă, care ar putea genera probleme, nu este efectuată. Totuși, chiar dacă s-ar efectua operația pe polinoame goale, aplicația ar returna rezultatul 0, sau, în cazul integrării, ar returna constanta care se adaugă la integrare.

De asemenea, prin folosirea funcției parseDouble, sunt acoperite cazurile în care utilizatorul ar introduce coeficienți sau puteri ale monoamelor foarte mari, deoarece această funcție nu va arunca nici o excepție, oricât de mare ar fi numărul introdus.

Dacă se introduc polinoame random și se încearcă efectuarea unei operații asupra lor, în cazul în care printre caracterele din șirul de caractere introdus se află vreo secvență de caractere care respectă pattern-ul descris la detalierea funcției mkFromString din clasa PolynomialController, aceasta va fi interpretată ca un monom valid, va fi preluată și operația va fi efectuată. Altfel, dacă în șirurile de caractere introduse de către utilizator nu se va găsi un astfel de subșir, text field-ul va fi interpretat ca fiind gol și va fi afișsat un mesaj care îl va ajuta pe utilizator să introducă un șir valid.

Pentru operațiile de adunare, scădere, înmulțire și împărțire este necesară introducerea de către utilizator a doua șiruri de caractere valide.

Pentru operațiile de integrare si derivare, este necesară introducerea de către utilizator a unui singur șir de caractere valid, în câmpul polinomului 1 (P1), câmpul P2 nefiind luat în considerare la apăsarea butonului respectiv.

}

6.Concluzii

Acest calculator de polinoame poate fi dezvoltat ulterior prin implementarea unor funcții pentru aflarea rădăcinilor polinoamelor de gradul 1, 2 și 3; de asemenea, se poate implementa calcului integralelor din polinoame, integrale pe mulțimea numerelor reale.

Tema m-a pus în situația de a parcurge din nou anumite operații cu polinoame, pentru a găsi soluția optimă prin care acestea pot fi implementate; de asemenea, am învățat să lucrez cu REGEX, dar și cu HashMap uri, pe care înainte nu le stăpâneam.

Am învățat să folosesc Bitbucket.

7.Bibliografie

<https://www.tutorialspoint.com/java/java_regular_expressions.htm>

<https://docs.oracle.com/javase/7/docs/api/index.html>

<https://www.geeksforgeeks.org/>

<https://stackoverflow.com/>

<https://mkyong.com/java/>