

DOCUMENTATIE

TEMA *1*

NUME STUDENT: HIRTESCU CIPRIAN-GABRIEL
GRUPA: 30225

CUPRINS

1.	Obiectivul temei.....	3
2.	Analiza problemei, modelare, scenarii, cazuri de utilizare	3
3.	Proiectare	4
4.	Implementare	5
5.	Rezultate	6
6.	Concluzii.....	6
7.	Bibliografie	6

1. Obiectivul temei

Obiectivul principal al temei este de a dezvolta o aplicatie in java denumita sugestiv “Calculator de Polinoame” , care este capabila sa efectueze diferite operatii specific pe polinoame.

Obiectivele secundare care au stat la baza indeplinirii obiectivului principal sunt:

- a. Abstractizarea informatiilor preluate din cerinta astfel incat sa se poata face diferenta intre clase, metodele pe care acestea le pot avea si alte functionalitati astfel incat sa se obtina o modularitate cat mai buna a codului rezultat , acest aspect va fi detaliat in capitolele urmatoare ;
- b. Determinarea atributelor specifice fiecarei clase, fapt care constituie una din partile fundamentale ale structurii proiectului;
- c. Determinarea metodelor , parte a proiectului in care determinam ce functionalitate poate sa aiba un obiect mai departe astfel incat sa devina folositor in alt context viitor;

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

Conform cerintei proiectului se pot determina atat cerinte functionale cat si cerinte non-functionale.

Cerinte functionale:

1. calculatorul de polinoame ar trebui sa permita utilizatorilor sa insereze polinoame ;
2. calculatorul de polinoame ar trebui sa permita utilizatorilor sa selecteze operatia matematica asupra datelor introduce anterior;
3. calculatorul de polinoame ar trebui sa efectueze operatia de adunare a doua polinoame respectiv sa scada doua polinoame;
4. calculatorul va trebui sa execute operatia de derivare respective de integrare a unui polinom ;
5. calculatorul va trebui sa rezolve inmultirea respectiv impartirea a doua polinoame;
6. calculatorul permite ca utilizatorul sa editeze un camp sau altul si sa foloseasca un rezultat anterior calculate.

Cerinte non-functionale:

1. calculatorul de polinoame ar trebui sa fie intuitiv si usor de folosit de catre utilizatori ;
2. operatiile ce trebuie executate ar trebui sa se faca intr-un timp relativ mic ;
3. rezultatul operatiilor trebuie sa fie corect si sa coincida unei abordari matematice

Scenariile posibile sunt legate de modul in care utilizatorii folosesc aplicatia , astfel incat in urma introducerii datelor in interfata , mai departe trebuie verificate , iar in caz de validare operatia

aleasa se va executa cu succes , in caz de esec , va aparea o atentionare cu privire la datele introduse.

De exemplu in cazul adunarii/scaderii/inmultirii/impartirii a doua polinoame abordarea este una similara. Se observa ca principalul actor care va interactiona cu aplicatia este utilizatorul .Detaliind afirmatia de mai sus din paragraful anterior , scenariul de succes ar cuprinde urmatoorii pasi: utilizatorul insereaza doua polinoame in interfata , apasand pe butoanele ce coincid la simbolurile dorite a fi introduse , mai departe se apasa pe butonul ce coincide la operatia aritmetica dorita , iar in cadrul locatiei corespunzatoare rezultatului va aparea valoarea calculata. Cazul de esec poate fi determinat de introducerea eronata a valorilor corespunzatoare celor doua polinoame sau de apasarea butonului corespunzator operatiei , inainte ca variabilele sa fie introduse.

Un alt caz care ar putea sa apara ar putea fi in cazul operatiei de impartire, al doilea polinom sa fie vid, ceea ce ar conduce la o operatie matematica invalida , ce va fi semnalata printr-un mesaj.

Aplicatia are posibilitatea de a remarca prezenta termenilor redundanti , a caror aparitie nu afecteaza in niciun fel functionalitatea , dar trebuie tratati cu atentie ; in afara de acest aspect , aparitia termenilor cu aceeasi putere vor fi adunati astfel incat sa apara sub un singur element de forma „coeficient X exponent” ;

3. Proiectare

Conform cerintei am putut identifica din punct de vedere al proiectarii OOP doua clase principale: Monom ale carei attribute principale sunt coeficientul si exponentul corespunzator unui element de forma “ $a \cdot X^b$ ” si o clasa Polinom care este formata dintr-o lista de monoame de forma anterior prezentata. Pe langa acestea , am putut distinge inca trei clase necesare modularizarii si pastrarii unei arhitecturi de forma MVC; acestea fiind : clasa Model ce reprezinta componenta care executa calculele asupra datelor introduse , clasa View , care continue butoanele , etichetele , text-field-urile , alaturi de frame-ul suport care le inglobeaza pe toate . Clasa Controller este folosita in scopul de a intruni partea de calcul aflata in Monom cu partea de interfata reprezentata de View, avand rol in a adauga “ascultatorii” pentru butoane si de a verifica datele introduse in text-field-urile asociate polinomului principal, secundar si rezultat. Legat de partea de calcul din cadrul Model-ului pe partea de algoritmi s-au folosit detalii legate de adunarea/scaderea a doua polinoame care se face termen cu termen ,unde variabilele au acelasi exponent . Derivarea si integrarea se executa conform regulilor $(a \cdot X^n)' = a \cdot n \cdot X^{(n-1)}$, respectiv : $\int a \cdot X^n = a / (n+1) \cdot X^{(n+1)}$.

Pentru inmultire se parcurge fiecare termen al primului polinom cu fiecare dintre termenii celui de al doilea polinom , se inmultesc si se insumeaza in final.

Impartirea celor doua polinoame se va efectua doar pentru acele polinoame a caror raport dintre coeficientul primului polinom si coeficientul celui alt sunt proportionali , adica polinoamele sa se poata imparti si sa aiba raportul coeficientilor un numar intreg, altfel impartirea nu se va putea face. Astfel incat se verifica daca primul polinom are coeficientul \geq decat cel alt iar in caz de adevar se considera un termen al catului format din coeficient , ca raport al coeficientilor

primului polinom si al celui alt si celui de-al doilea iar ca exponent rezultatul scaderii dintre cele doua valori ale exponentilor .

4. Implementare

Descrierea clasei Monom:

Clasa Monom creeaza un obiect de forma $a \cdot X^b$, avand ca si variabile instantate : coeficientul si exponentul declarate ambele ca int si avand modificatorul private . Prezinta de asemenea setter si getter pentru ambele componente ale sale , alaturi de metoda compareTo , necesara mai departe in clasa Model in momentul in care se doreste ordonarea descrescatoare in functie de valoarea exponentilor .

Descrierea clasei Polinom:

Clasa Polinom este formata dintr-o lista de monoame stocate sub forma anterior mentionata avand de asemenea modificatorul private si o metoda toString suprascrisa care intoarce polinomul citit sub o forma cat mai lizibila : $a \cdot X^n + b \cdot X^{n-1} + \dots$.

Descrierea clasei Model:

Clasa Model reprezinta partea de calcul din cadrul intregului proiect, prezinta o singura variabila instantata care este de fapt rezultaul operatiei .

1. Metoda eliminareRedundante primeste ca si parametru un polinom si executa o trecere asupra intregului polinom in urma careia se aduna sub forma unui singure variabile , coeficientii acelor variabile ce au aceeasi valoare la exponent, iar in cazul in care aceasta suma este zero , valoarea va fi eliminata permanent din intregul polinom.

2. Metoda adunaSiScade primeste ca si parametri doua polinoame alaturi de o valoare intreaga denumita sugestiv semn . In cadrul acesteia se executa adunarea / scaderea celor doua polinoame primite drept parametri, operatia corespunzatoare fiind dictata de semn a carei valoare poate fi 1 sau -1 , in acest caz am privit scaderea ca o adunare cu valoarea negata a celui de al doilea termen , rezultatul fiind unul corect , astfel scazand numarul de metode si totodata scazand si complexitatea.

3. Metodele denumite sugestiv „inmulteste” , „imparte” executa operatiile de inmultire si respectiv de impartire a doua polinoame in modul exemplificat in capitolul anterior.

4. Metodele „deriveaza” si „integreaza” realizeaza operatiile de derivare respectiv de integrare a doua polinoame , analog metodei din capitolul anterior.

Descrierea clasei View:

Clasa View reprezinta partea de interfata a acestui proiect , fiind constituita din o multitudine de butoane, etichete si text-field-uri ce sunt foarte usor de folosit si sugestive pentru utilizatorul aplicatiei.

Referitor la butoane exista butoane pe de o parte care au ca si scop introducerea datelor in locul potrivit , butoane pentru selectarea locatiei unde se doreste editarea , butoane pentru determinarea operatiei dorite si butoane pentru stergere si folosire a valorii anterior calculate. Dintre acestea se mentioneaza ca butoanele de la 0-9 alaturi de cele ce corespund operatiilor de adunare : "+" , scadere: "-", , inmultire: "*" , ridicare la putere "^" si cel care corespunde variabilei polinomului : "X" o data ce sunt apasate apar in text-field-ul selectat , urmand spre a fi validate. Butoanele : "Pol1" , "Pol2" determina locatia asupra careia butoanele anterioare vor actiona , selectand ori text-field-ul primului polinom ori al celuiilalt. Butoanele: „DEL” , „CLR” executa stergere partiala , respectiv totala a continutului aflat in unul din cele doua locatii de scriere , iar „Ant” copiaza valoarea aflata in text-field-ul asociat polinomului rezultat si il suprapune peste cel din text-field-ul primului polinom.

Toate acestea sunt intrunite in frame-ul: "frmPolynomialCalculator" , a carui vizibilitate este setata in cadrul Controller-ului.

Descrierea clasei Controller:

Clasa Controller este partea in care Model-ul si View-ul sunt instantiate si sunt folosite simultan pentru a forma un tot unitar care sa rezolve problema propusa.

In cadrul acesteia, sunt adaugati „ascultatorii” pentru butoanele plasate in frame-ul din View , astfel incat sa functioneze in modul dorit, pe langa acest aspect, se verifica daca datele introduse de utilizator corespund vreunui pattern cunoscut de catre calculator , in caz afirmativ calculele vor avea loc , altfel va fi afisata o fereastra ce explica ce exceptie a aparut la introducerea datelor.

5. Rezultate

In urma scenariilor testate cu JUnit s-a constatat ca operatiile se executa in mod corect , iar nedeterminarile de forma 0/0 ; polinom/0 sunt tratate special astfel incat sa nu afecteze functionalitatea generala a proiectului. Pe langa acest aspect se mentioneaza faptul ca toate operatiile se executa pe polinoame a caror coeficienti sunt numere intregi , rezultatul asteptat continand de asemenea coeficienti intregi , astfel ca in cazul impartirii daca cele doua polinoame nu au coeficienti proportionali rezultatul final va fi zero , ceea ce inseamna ca actiunea de impartire nu s-a putut efectua .

6. Concluzii

Pentru o dezvoltare ulterioara s-ar putea considera ca polinomul ar avea coeficienti reali , ceea ce ar conduce la o complexitate mai mare in cazul impartirii , restul operatiilor ramanand la fel.

7. Bibliografie

Calculator de Polinoame

Primul Polinom =

 X^3+X^2

Al doilea Polinom =

 $6*X^{23}$

Polinom Rezultat =

 $+6*X^{26}+6*X^{25}$

Inmulteste

Scade

Imparte

Deriveaza

Aduna

Integreaza

1

2

3

0

+

-

4

5

6

Ant

*

^

Pol1

7

8

9

X

DEL

CLR

Pol2