**FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE**

**DEPARTAMENTUL CALCULATOARE**

**Tema 1**

la disciplina

**Tehnici de programare**

**Titlu**

***„Calculator Polinomial”***

Nume si prenume : Trif Gheorghe Andrei

Grupa: 30225

An academic : 2018-2019

**1.Obiectivul temei**

Propuneti, proiectati si implementati un sistem de procesare a polinoamelor de o singura variabila cu coeficienti intregi.

|  |  |  |
| --- | --- | --- |
| Obiective secundare | Descriere | Capitolul |
| Dezvoltare de use case-urii si scenarii in legatura cu intelegera temei propuse. | Se va dezvolta un comportament pe care programul va incerca sa le urmeze. | 1 |
| Analiza problemei si modelarea acesteia | Se va prezenta proiectarea OOP a problemei ,diagrame UML ,pachete si structuri de date folosite. | 2 |
| Dezvoltarea algoritmilor | Se va perzenta operatile alese  si descrierea lor. | 3 |
| Impelmentarea solutiei | Se va prezenta codul sursa aferent proiectari. | 3 |
| Testare aplicatiei | Prezentarea ca de cazuri favorabile de rulare si modul de utilizare ,dar si anumite defecte. | 4 |
| Dezvoltare ulterioara | Noi idei in a inbunatati proiectul | 5 |

Scopul acestui proiect este de a implementa funcțiile de bază pe polinoame, pentru a descrie modul în care acestea pot fi create și manipulate prin intermediul paradigmei programării orientate pe obiecte. Printre operațiile de bază se vor număra cele de adunare, scădere, înmulțire și împărțire, dar și o serie de operații auxiliare precum integrare si derivare .

Ca prima idee am incercat sa creed monomul din interfata acest lucru putea dura ore pentru un polinom de grad mare astfel am hotarat sa introduc polinomul ca string si sa ii fac o conversie la intreg lucru care a fost mult mai usor.Astfel dupa ce am reusit sa spar acele stringur am continuat spre creearea listei se monoame.Dupa crearea unei liste aveam nevoie de operatii pe care interfata sa le execute si care sa imi genereze un anumit rezultat.Orice calculator merge pe un principiu asemanator de aceea cred ca mergand pe aceasta idee nu este o decizie proasta.

2.Analiza problemei si modelarea acestei

2.1.Analiza problemei

În matemeatica, un **polinom** este o expresie construită dintr-una sau mai multe variabile si constante folosind doar operatii de adunare ,scadere,inmultire,ridicare la putere.Polinoamele

sunt construite din termeni numiti monoame,care sunt alcatuiti dintr-o constanta numita coefcient inmultita cu una sau mai multe variabile.Fiecare variabila poate avea un exponent intreg pozitiv.In cadrul unui anumit monom exponentul este numit grad.Monomul cu gradul cel mai mare reprezinta "capul" polinomului , astfel putem spune ca polinom este de grad "p".

Un polinom de gradul n in nedeterminata X se scrie in forma canonica astfel:

P(X) = a0X n + a1X n−1 + ... + an−1X + an, unde a0 6= 0. Numerele a0, a1, ..., an sunt coeficientii polinomului,.

***Teorema împãrtirii cu rest***: "f,gÎC[x], g¹0 existã polinoamele unice q,rÎC[x] astfel încât f = gq + r, grad *r* < grad*g*.

***Schema lui Horner***: ne ajutã sã aflãm câtul q = b0Xn-1 + b1Xn-2 + . + bn-1 al împãrtirii polinomului f = a0Xn + a1Xn-1 + a2Xn-2 + . + an la binomul X-a; precum si restul acestei împãrtiri r = f(a);

Cateva proprietati ale polinoamelor:

1. Suma a două polinoame este un polinom;
2. Produsul a două polinoame este un polinom;
3. Derivata unui polinom este un polinom;
4. Primitiva unui polinom este un polinom;

Prin analiza problemei, ne referim la un prim set abstract de operații și proprietăți prin care încercăm să depistăm eventualele însușiri și comportamente ale proceselor necunoscute. Programarea orientată ne oferă aici un avantaj clar, tocmai fiindcă ea permite să taclăm problema de la un nivel superiror, fără a mai fi constrâși, într-o așa măsură, de caracteristicile tehnice.

Această strategie de conceptualizare, mai poartă numele și de bottom-up design. Este foarte avantajoasă din prisma găsirii componentelor constituente, deoarece pot fi găsite, relativ ușor, structuri cu o legătură directă în lumea reală( obiecte, acțiuni etc.). Din păcate această versatilitate vine cu prețul complexității, ea crescând pe măsură ce se avansează pe nivelele inferioare.Bottom -up poate fi foarte utila pentru crearea unor solutii rapide dar nu foarte complexe fiind utilizata cel mai adesea de un singur programator care lucreaza independent .

De cele maimulte ori se porneste de la specificarea proiectului cautandu-se verbe sau substantive relevante pentru crearea de clase si pentru crearea de metode.Astfel odata ce am realizat pasul de mai sus vom avea o idee foarte clara despre comportamentul aplicatiei pe care urmeaza sa o creeam si care sunt funtiile sale de baza.

O astfel de aplicatie a fii folosita defoarte multe persoane sau poate chiar de o companie astfel un prim pas in creearea sa e sa determinam intrarile pe care un anumit utilizator ar dori sa le poata accesa.Acest lucru se face cel mai probabil prin cadrul interfetei dintre program si utilizator.Aceasta interfata trebuie sa fie foarte usor de folosit dar si foarte precisa astfel incat sa nu induca in eroare utilizatorii.Deoarece in zilele noastre mai exista persoane care nu sunt invatate cu astfel de lucruri si care la prima vedere ii poate crea o stare de confuzie.

In cadrul Calculatorului Polinomial se specifica foarte clar ca principala sa caracteristica o sa fie rezolvarea difieritelor operatii cu polinoame astfel eu in proiectarea mea am incercat sa usurez acest lucru prin creearea a doua zone de tip text in care vor fi introduse cele 2 polinoame deoarece introducerea lor in forma numerica poate creea anumite probleme utilizatorilor care nu cunosc anumite detalii despre ce inseamna un monom sau un polinom.

|  |
| --- |
| Polinom |
|  |
| C:  Polinom();  M:  Polinom1(String):ArrayList<Monoame>  getMonomIndex(int):Monom  size():int  getMo(ArrayList<Monom>):void  toString():String  doString():String |

2.2.Proiectarea claselor

|  |
| --- |
| Monom |
| coef:int  grad:int  coefD:double |
| C:  Monom(int ,int);  Monom();  Monom(double,int);  M:  getCoef():int  setCoef(int):void  setCoefD(double):void  getCoefD():double  setGrad(int)  getGrad():int; |

|  |
| --- |
| Intefata GUI |
| tf:TextField  tf2:TextField  rezultat:TextArea  egal:JButton  plusB:JRadioButton  minusB:JRadioButton  mulB:JRadioButton  pol1:JCheckBox  pol2:JCheckBox  pol11:JCheckBox  pol22:JCheckBox |
| C:  Interfata()  M:  Calculeaz():void |

|  |
| --- |
| Operatii |
|  |
| C:  Operation()  M:  addP(Polinom,Polinom):Polinom  minusP(Polinom,Polinom):Poinom  mulP(Polinom,Polinom):Poinom  derivP(Polinom):Poinom  integP(Polinom):Poinom |

Calculatorul polinomial este alatuit din 4 clase :Monom,Polinom,Interfata si clasa JFrame importata din java.swing .

In clasa Interfata este creata fereasta aplicatiei impreuna cu componentele sa le butoane ,casute si campuri text.Tot aici este instantiat butonul de egal care are rolul de a furniza rezultatul operatiei cerute de utilizator.Metoda Calculeaza reprezinta clculatorul in care are loc fiecare operatie aleasa de utilizator

Clasa Monom dupa cum bine am precizat mai sus un monom este o mica "bucatica" din polinom ca care are ca principale caracteristici un coeficient intreg sau real si un grad intreg.In aceasta clasa sunt definite doar metodele simple de acces la elemente get si set.

Clasa Monom mai prezinta si 3 tipuri de Constructori dintre care doi mai speciali.Primul este constructorul pentru monom cu coeficient si grad intreg folosit in operatii unde este anticipat ca la rezultate va fii tot un monom cu coeficient intreg.Al doilea constructor este folosit in crearea de monoame cu coeficient real in operati perecum integrarea sau impartirea.

Clasa Operatii contine operatiile de baza pe care monoamele le pot suporta intre ele:adunare , scadere,inmultire ,impartire,etc.

Clasa Polinom nu este nimic mai mult decat o lista de monoame care daca stai sa te gasdesti este absolut banala si care este foarte natural deoarece o insiruire de monoame intre care apar operanzii +,- si fiecare grad este diferit reprezinta un polinom.Aceasta clasa are o metoda un pic mai speciala fiind vorba despre o metoda care primeste un string si returneaza o lista de monoame.

Intre cele doua clase Monom si Polinom exista o relatie de compozitie deoarece este foarte evident un Polinom nu poate exista fara o lista de monoame.

3.Impementarea Claselor

3.1.Clasa Monom

Aceasta clasa nu este o clasa complexa ii ea fiind definiti doar constructori necesari pentru a creea lista de monoame.Ca metode prezinta gettere si setter.

Ex: -5X^2 reprezinta obiectul clasei monom avand un grad si un coefiicient ca atribute.

**public** Monom(**int** coef1,**int** grad1) {

**this**.coef=coef1;

**this**.grad=grad1;

}

Acesta este constructorul principal folosti la aproximativ toate operatiile.

3.2.Clasa Polinom

Aceasta clasa reprezinta Polinomul in sine este alcatuit dintr-o lista de monoame.

**private** ArrayList<Monom> monom;

**public** Polinom() {

**this**.setMo(**new** ArrayList<Monom>());}

unde setMo este un geter care returneaza lista de moname.

**public** ArrayList<Monom> getMo() {

**return** monom;

}

Ca sa declar lista am folosit libraria java.util.ArrayList care contine diverse metode pentru lucrul cu liste.

Metodele toString() si doString() sunt folosite pentru a transforma inapoi in string monomul rezultat in urma unei operatii iar singura diferenta este ca toString suprascrie aceeasi metoda din clasa Object si se ocupa de rezultatele operatiilor cu coeficient intreg pe cand doString nu suprascrie nimic este o metoda de transformare care transforma polinoamele rezultate in urma operatiilor la care se intuieste ca rezultatul va fi cu virgula.Aici singurul lucru mai special este v=Math.*round*(i.getCoefD()\*100.0)/100.0; acesta linie care taie un anumit nr de zecimale ale nr ex:1.998898989 va fii 1.99.

Metoda size() returneaza luncimea listei de monoame.

Metoda Polinom1(String) are rolul de a crea efectiv dintr-un sting polinomul de care avem nevoie pentru a face opertiile

**public** ArrayList<Monom> Polinom1(String s) {

setMo(**new** ArrayList<Monom>());//definim lista

String v=s.replace("-","-n");//inlocuim toate - din string cu -n ca sa stim care coeficienti vor fi negativi

String[] monom = v.split("\\+|\\-");//spargem stringul in functie de plus si minus

**for**(String t : monom){//Foreach string din vectorul de stringuri

**if**(t.equals(""))//daca dupa ce am spart primul string e null trecem la urmatorul

**continue**;

String[] mon = t.split("\\^");//spargein functie de ^ si creeam un vector de string cu 2 elemente

**int** power = 0;

**int** coef = 0;

**if**(mon.length == 2)//in mon[1] vom avea puterea in tot deauna

power = Integer.*parseInt*(mon[1]);//convetim la int

**if**(mon[0].substring(0, 1).equals("n")){ // daca primul caracte din mon[0] e n stim ca vom avea un coefcient negativ

**if**(mon[0].substring(1, 2).equals("x"))//daca al doilea caracter e x vom avea coef=-1

coef=-1;

**else** {

coef=Integer.*parseInt*(mon[0].substring(1, 2))\*(-1);//luam coeficientul si il convetim

}

}

**else**

**if**(mon[0].equals("x")){//daca monp[0]=x coef ii 1

coef=1;

}

**else**

**if**(mon[0].length()==1){

coef=Integer.*parseInt*(mon[0]);//luam coeficientul pozitiva

power=0;

}

**else**{

coef=Integer.*parseInt*(mon[0].substring(0,mon[0].length()-1));}

**this**.getMo().add(**new** Monom(coef,power));// adaugam in lista noul monom

}

**return** **this**.getMo();//returnam Polinomul

3.3.Clasa Operatii

In aceasta clasa sunt implementate metodele de adunare ,scadere ,inmultire,integrare,derivare.Aceste metode sunt concepute pe baza ideei de rezolvare a acestor operatii in matmatica nimic complicat doar respectarea formuleleor si a algoritmilor matematici.

3.4.Clasa Interfata

In aceasta clasa este construita interfata prin apelare la JFrame.Astfel se creaza fereastra in care adaugam Jpanel-urile.

JFrame frame = **new** JFrame ("Calculator Polinomial");

frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

frame.setSize(640, 480);

frame.setContentPane(p);

frame.setVisible(**true**);

Am creat un Jpanel p in care am adaugat 3 sub panealuri si am instantiat ascultatori pentru butonul = egal.addActionListener(**new** ActionListener(),dar in care am adaugat si butanele precum si inputurile

JPanel p = **new** JPanel();

p.add(panel1);

p.add(panel2);

p.add(panel3);

Metoda Calculeaza face operatiile selectate de fiecare data cand = este apasat

**public** **void** actionPerformed(ActionEvent arg0) {

Calculeaza();

}

Metoda calculeaza verifica daca inputurile sunt goale caz in care nu avem ce calcula degeaba selectam o operatie

**if**(input1.isEmpty() && input2.isEmpty())

run=**false**;

**else**

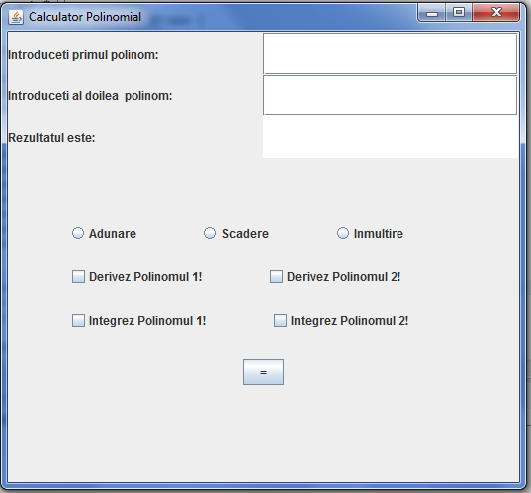
run=**true**;

Daca avem ceva in cele 2 inputuri atunci convertim cele 2 inputuri din string la polinom si verificam ce operatie sa selectat if(plusB.isSelected()) si facem operatia respectiva

4.Testarea aplicatiei

Este indicat ca polinomul sa fie introdus in ordine cu gradul cel mai mare prima data.

Astfel dupa introducerea in cele doua campuri a polinoamelor dorite se selecteaza una intre operatii si se apasa pe egal dupa apasarea pe egal campul rezultat nu mai poate fi editat



5.Dezvoltarea ulterioara

Din această primă temă, mi-am dat seama, din păcate la final de importanța sepărării, la nivel conceptual, a parsării string-urilor fată de interpretarea lor de către clasele Polinom și Monom. Această greșeală am facut-o în principal, fiindcă nu am reușit să estimez complexitatea reală a problemei. O clasă separată, Parsare, de exemplu, ar fi fost o alternativă mult mai viabilă fiindcă în acest fel nu ar fi fost nevoie să existe vreo legătură între procesul de determinare al monoamelor din string-uri și crearea efectivă a polinomului.

O alta idee de dezvoltare ar fi adaugare unei metode sau clase de verificare a inputului introdus deoarece in aceasta parte apar foarte multe probleme care nu se pot rezolva cu un if

Din aceasta tema am invatat ca este foarte important sa privesti problema bottom-up dar si sa te axezi mai mult pe caracteristicile reale ale unui obiect nu doar sa sari repede la partea de cod.O buna intelegere a ceea ce urmeaza sa faci simplifica mult partea de scriere de cod.

Bibliografie

[1]. https://ro.wikipedia.org/wiki/Polinom

[2].http://www.referatele.com/referate/matematica/online5/Valoarea-unui-polinom--Operatii-cu-Polinoame--Egalitatea--Gradul-Forma-Polinoamelor-referatele-com.php

[3]. https://www.mkyong.com/tutorials/junit-tutorials