**DOCUMENTATIE TEMA 1**

**CALCULATOR POLINOAME**

**Nume prenume:Toderean Alexandru**

**Grupa:30224**

**Profesor Laborator Assist. Claudia Pop**

Contents

[1. Cerinte Functionale 3](#_Toc476131445)

[2. Obiective 3](#_Toc476131446)

[2.1. Obiectiv Principal: 3](#_Toc476131447)

[2.2. Obective Secundare: 3](#_Toc476131448)

[3. Analiza Problemei 3](#_Toc476131449)

[4. Proiectare 3](#_Toc476131450)

[4.1. Structuri de date 3](#_Toc476131451)

[4.2. Diagrama de clase 3](#_Toc476131452)

[4.3. Algoritmi 3](#_Toc476131453)

[5. Implementare 4](#_Toc476131454)

[6. Testare 4](#_Toc476131455)

[7. Concluzii si Dezvoltari Ulterioare 4](#_Toc476131456)

[8. Bibliografie 4](#_Toc476131457)

# Cerinte Functionale

Dezvoltati un calculator de polinoame care sa perimita urmatoarele operatii:

* Citirea unui polinom de la tastatura sub forma 3X^2+2X^3-X+2
* Adunare, Scadere, Inmultire, Impartire, Derivare, Integrare
* Crearea unei interfete grafice pentru calculatorul de polinoame

# Obiective

## Obiectiv Principal:

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

## Obective Secundare:

|  |  |  |
| --- | --- | --- |
| **Obiectiv Secundar** | **Descriere** | **Capitol** |
| Dezvoltarea de use case-uri si scenarii |  | 3 |
| Alegerea structurilor de date |  | 4 |
| Impartirea pe clase |  | 4 |
| Dezvoltarea algoritmilor |  | 4 |
| Implementarea solutiei |  | 5 |
| Testare |  | 6 |

# Analiza Problemei

Diagrame de use case pentru cel putin o operatie

* <http://tynerblain.com/blog/2007/04/09/sample-use-case-example/>

# Proiectare

## Structuri de date

## Diagrama de clase

## Algoritmi

# Implementare

Descrierea fiecarei clase cu campuri si metodele importante

Descrierea implemantarii interfetei utilizator

# Testare

Se vor prezenta cateva scenario de testare (de exemplu 2-3 operatii pe polinoame)

Se va prezenta scenariile pentru testare cu JUnit

# Concluzii si Dezvoltari Ulterioare

# Bibliografie

**2.Obiectivele temei**

Obiectivele acestei temei sunt propunerea ,proiectarea si implementarea unui sistem de procesare a polinoamelor de o singura variabila cu coeficienti intregi.Acestea implica o serie de operatii a polinoamelor , mai exact: adunarea,scaderea,inmultirea si impartirea.

3.Analiza problemei

Complexitatea problemei se reduce la gasirea unui mod usor de a reprezenta polinoamelor cu o singura variabila. Ii vom reprezenta pe acestia cu ajutorul coeficientilor si cu ajutorul gradului lor. In acest fel procesarea polinoamelor nu este atat de dificila. Desigur sunt si alte operatii care se implementeaza mult mai greu sub aceasta forma, cum ar fi interpolarea, gasirea radacinior, la polinoamele de grad mare, sau gasirea celui mai mic divizor comun. Dar proiectul in cauza nu trateaza astfel de subiecte complicate. Toate operatiile care le vom efectua vor putea fi posibile cu ajutorul coeficientilor si al gradului.

Un polinom este o expresie construita dintr-una sau mai multe variabile si constante, folosind doar operatii de adunare,scadere,inmultire, impartire, derivare si integrare. Polinoamele sunt construite din termeni numiți monoame, care sunt alcătuite dintr-o constantă (numită coeficient) înmulțită cu una sau mai multe variabile. Fiecare variabilă poate avea un exponent constant întreg pozitiv. Exponentul unei variabile dintr-un monom este egal cu gradul acelei variabile în acel monom. Pentru că  x=x^1 , gradul unei variabile fără exponent este unu. Un monom fără variabile se numește monom constant, sau doar constantă. Gradul unui termen constant este 0. Coeficientul unui monom poate fi orice număr, inclusiv fracții, numere iraționale sau negative. Gradul întregului monom este suma gradelor tuturor variabilelor din el.Un polinom este o sumă de unul sau mai multe monoame. De exemplu, următoarea expresie este un polinom :  3x^2 - 5x + 4\,.

Pentru ca proiectul sa arate mai bine ar trebui sa aiba o interfata care va face conexiunile dintre utiizator si programul in sine. Acesta ar trebui sa respecte unele cerinte: spatiu pentru text, pentru citirea coeficientior, sau afisarea rezultatelor. De asemenea vom avea si panoul pentru butoanele care conduc la operatii, astfel utilizatorul are oportunitatea de a alege operatiile mult mai usor.

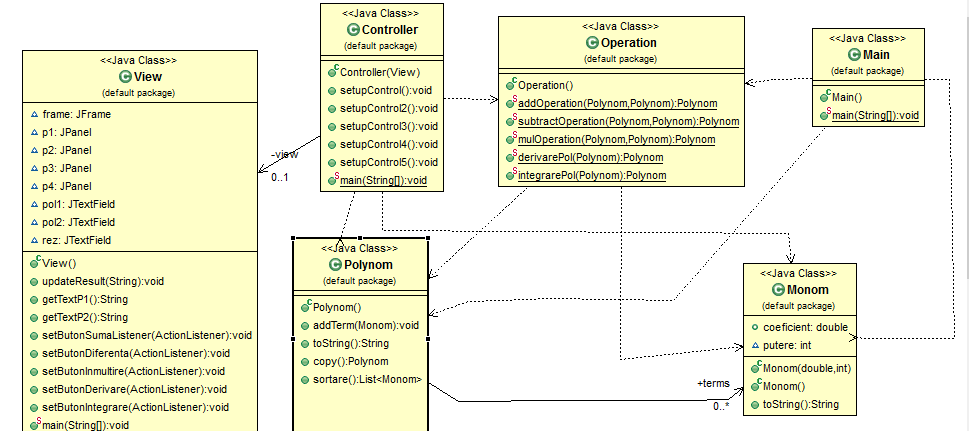
Pentru a implementa acest proiect in programarea orientata pe obiecte, vom avea nevoie de o clasa pentru a reprezenta polinoamele, Polynom,o clasa Monom in care-mi declar puterea si coeficientul monomului si un mod de afisare a acestuia,una pentru a implementa operatiile, Operation,una pentru interfata(View),una pentru control interfata(Controller),si inca una pentru a-mi testa operatiile inainte de punerea in interfata(Main). In interfata se vor face toate conectarile intre intrarile si iesirile care se refera la polinoame si operatiile aplicate lor. Prin intermediul controllerului se va genera rezultatul operatiilor in interfata.

Ca date de intrare avem coeficientii celor doua polinoame, acestea se introduc in partea de sus a ecranului. Aici vom afisa mesajul folosind Jtext . Prin aceleasi metode se vor afisa mesaje pentru introducerea coeficientilor celuilalt polinom. Dupa textul „Polinom1 si Polinom2” vor fi dreptunghiuri speciale pentru introducerea coeficientilor cu spatii intre ei. Vor fi implementati folosint JtextFields.

Partea din mijloc a panoului este alcatuita din 5 butoane care vor reda operatiile de adunare,scadere,inmultire,derivare si integrare.In parte dreapta sus va fi situat butonul Exit care opreste programul si iese din acesta,chiar jos la campul:”rezultat” se vor afisa rezultatele operatiilor pe polinoame.

**4.Proiectarea**

Pentru descrierea modelelor si a diagramelor de clase am folosit UML(Unified Modeling Language), Diagramele de clasă sunt folosite pentru reprezentarea concretă a unor instanțe de clasă, așadar obiecte, și a legăturilor concrete dintre acestea .



Proiectul contine 6 clase: Prima clasa este clasa Polynom care este alcatuita dintr-o lista de obiecte de tip Monom. Clasa Monom reprezinta un termen din polinom , caracteristicile monomul fiind puterea si coeficientul sau. Clasa operation imi implemeteaza operatiile care au loc asupra polinoamelor. Clasa View creaza interfata grafica, aici implementandu-se si actiunile care vor avea loc pe butoane in functie de operatii.Clasa Controller am folosit-o pentru setarea propriu-zisa a butoanelor si a intocmirii operatiilor in functie de butoane.Am mai creat o clasa main in care mi-am testat initial corectitudinea operatiilor implementate.

Utilizatorul va avea posibilitatea de a alege ce operatie doreste sa se efectueze asupra polinoamelor introduse prin apasare a unuia dintre cele 5 butoane.

**5.Implementarea**

Proiectul are doar un pachet care contine 6 clase: Polynom , Monom, Main(test) , Operation, View(interfata),Controller.

**Clasa Monom:**

-are 2 campuri prin care este reprezentat monomul,coeficientul si puterea.Avem un constructor pentru a modifica coeficientul si puterea prin intermediul parametrilor.Mai o metoda prin care afisam monoamele in functie de coeficient si putere.

**public** **double** coeficient;

**int** putere;

**public** Monom(**double** coeficient, **int** putere) {

**super**();

**this**.coeficient = coeficient;

**this**.putere = putere;

}

**public** String toString() {

**if**(coeficient>=0)

**return** "+" + coeficient + "x^" + putere ;

**else**

**return** coeficient + "x^" + putere;

}

**Clasa Polynom:**

-In ea este creata o lista de tip monom care reprezinta polinomul,un constructor pentru instantierea unui ArrayList de tip monom.O functie addTerm cu parametrul Monom care imi adauga monoame in lista mea de monoame,aceasta verifica daca monomului current I este gasit un termen care are aceleasi puteri,daca I este gasit atunci se insumeaza cu acesta,daca nu se aduna la lista de monoame.

-Din nou mai avem o metoda toString care imi afiseaza lista de monoame,si mai am inca o metoda numita copy care-mi creaza o copie a unui polinom,este necesare la unele operatii.

**public** List<Monom> terms;

**public** Polynom() {

**super**();

**this**.terms = **new** ArrayList<Monom>(); //creare polinom

}

**public** **void** addTerm(Monom nM) {

**boolean** ok = **false**;

**for** (Monom m : terms) {

**if** (m.putere == nM.putere) { //adaugare monoame in polinom

m.coeficient += nM.coeficient;

ok = **true**;

**break**;

}

}

**if** (ok == **false**) {

terms.add(nM);

}

}

**public** String toString() {

String s = "";

**for**(Monom m : terms) {

s += m.toString();

}

**return** s;

}

**public** Polynom copy (){

Polynom p = **new** Polynom();

**for**(Monom m: terms)

p.addTerm(**new** Monom(m.coeficient, m.putere));

**return** p;

}

**Clasa Operation:**

-aici am implementat cele 5 operatii:adunare,scadere,inmultire,derivare,integrare.Am creat o metoda static pentru fiecare operatie.

**public** **static** Polynom addOperation (Polynom p1, Polynom p2){

Polynom rez = p1.copy();

**for**(Monom m: p2.terms)

rez.addTerm(m);

**return** rez;

}

**public** **static** Polynom subtractOperation (Polynom p1, Polynom p2){

Polynom rez = p2.copy();

Polynom copyP1 = p1.copy();

**for**(Monom m: copyP1.terms){

m.coeficient \*= -1;

rez.addTerm(m);

}

**return** rez;

}

**public** **static** Polynom mulOperation(Polynom p1,Polynom p2){

Polynom rez= **new** Polynom();

**for**(Monom m: p1.terms ){

**for**(Monom n: p2.terms){

rez.addTerm(**new** Monom(m.coeficient\*n.coeficient, m.putere+n.putere));

}

}

**return** rez;

}

**public** **static** Polynom derivarePol( Polynom p1)

{

Polynom x= **new** Polynom();

**for** ( Monom m : p1.terms )

**if** ( m.putere > 0 )

x.addTerm( **new** Monom ( m.coeficient\*m.putere , m.putere-1));

**return** x;

}

**public** **static** Polynom integrarePol( Polynom p1)

{

Polynom x = **new** Polynom();

**for** ( Monom m : p1.terms )

x.addTerm(**new** Monom( m.coeficient /(m.putere+1) , m.putere+1));

**return** x;

}

-**Clasa View si Controller** implementeaza impreuna interfata grafica a proiectului,am creat un buton pentru fiecare operatie,la apasarea butoanelor consecutive campul de text se schimba in realizand operatiile dorite,pentru afisarea celor 3 campuri de text si a celor 5 butoane am folosit o fereastra de 400 pe 400 pixeli.

-**Clasa Main**  este o clasa in care am creat instante pentru obiectele Operation,Polynom,Monom,dupa care am testat operatiile pentru a fi sigur ca pot fi implementate pe partea de grafica.

**6.Rezultat**

Rezultatele se afiseaza in ultimul JtextField din interfata grafica . Pe baza informatiilor primite in campurile de mai sus , se creaza polinoamele , iar la apasarea unui buton se face operatia corespunzatoare . Rezultatul se va afisa sub forma : ak\*X^k+ ....+ a1\*X^1 a0 , unde a0,a1, ...ak sunt coeficientii polinomului si 1...k sunt gradele . Mentionez ca polinoamele nu se vor afisa in ordinea gradelor deoarece nu am implementat o functie de sortare.

**7.Concluzii. Dezvoltari ulterioare**

Dezvoltari ulterioare care pot fi facute: marirea numarului de operatii: valoarea intr-un anumit punct, gasirea radacinilor, ridicarea la o anumita putere,impartirea polinoamelor. Deasemenea putem realiza si extinderea numarului de polinoame, extinderea interfetei(un numar mai mare de butoane, a cutilor pt text, mesaje).Pentru implementarea operatiilor mai complicate este nevoi de mai multe cunostinte a modulelor matematice. Dar trebuie sa ne reamintim ca acest lucru nu este scopul unui programator ci a unui matematician.

**8. Bibliografie**

1. JDK-Windows Help
2. SUN online documentation,sun tutorials
3. *“Java de la 0 la expert”*
4. *http://stackoverflow.com/*