**Universitatea Tehnica Cluj-Napoca**

**Facultatea de Calculatoare si Automatica**

**Departamenul de Calculatoare si Tehnologia informatiei**

Sistem de procesare a polinoamelor de o singura variabila cu coeficienti intregi

Nume: Cuciurean Catalin

Grupa: 30225

Materie: Tehnici de programare

**Obiectivul proiectului**

Polinoamele si operatiile cu acestea sunt unele dintre cele mai comune structuri matematice din Computer Science. Motivul principal este simplitatea algebrica a operatiilor – vorbim doar de adunari si inmultiri (care la randul lor sunt adunari repetate). Avantajul este ca polinoamele pot aproxima oricat de bine orice functie continua si derivabila, deci pot “interpola” functii mult mai complexe.

Tocmai de aceea, in multe aplicatii se vor regasi polinoame, intr-o forma sau alta, alaturi de un set de functii de baza cu care se pot manipula

Polinomul reprezinta o expresie algebrica constituita din mai multe monoame, legate intre ele prin semnul plus sau minus, suma algebrica a mai multor monoame.

Un monom fara variabile se numeste monom constant, sau doar constanta. Gradul unui termen constant este 0.

In matematica, un polinom este o expresie construita dintr-una sau mai multe variabile si constante, folosind doar operatii de adunare, scadere, inmultire si ridicare la putere constanta pozitiva intreaga.

Obiectivul acestui proiect este de a implementa operatii si functii de baza cu polinoame, , pentru a reliefa modul in care acestea pot fi create si manipulate prin intermediul paradigmelor programarii orientate pe obiecte.

Este folosita o interfata grafica cu utilizatorul care ofera posibilitatea de a introduce datele intr-un mod firesc, acestea fiind transformate ulterior in informatii concrete care pot utilizate in realizarea operatiilor.

**Analiza proiectului**

Prin analiza problemei, ne referim la un prim set abstract de operatii si proprietati prin care incercam sa depistam eventualele insusiri si comportamente ale proceselor necunoscute. Programarea orientata ne ofera aici un avantaj clar, tocmai fiindca ea permite sa taclam problema de la un nivel superiror, fara a mai fi constrasi, intr-o asa masura, de caracteristicile tehnice.

In partea de analiza se identifica clasele principale care urmeaza a fi implementate, si legaturile dintre ele. Dupa ce clasele au fost create urmeaza implementarea operatiilor.

**Scenarii**

Utilizarea normala a programului presupune introducerea primului polinom in formatul clasic LateX, in mod aleator, apoi se apasa butonul de OK pentru a vedea polinomul minificat si ordonat. Se repeta procesul si pentru polinomul numarul doi, iar dupa ce user-ul il poate vizualiza in zona nr. II, poate sa efectueze operatiile elementare existente sub TextField-uri. La fiecare din operatiile mentionate mai sus, utilizatorul poate sa atribuie valori specifice lui x pentru a vedea rezultatul numeric.

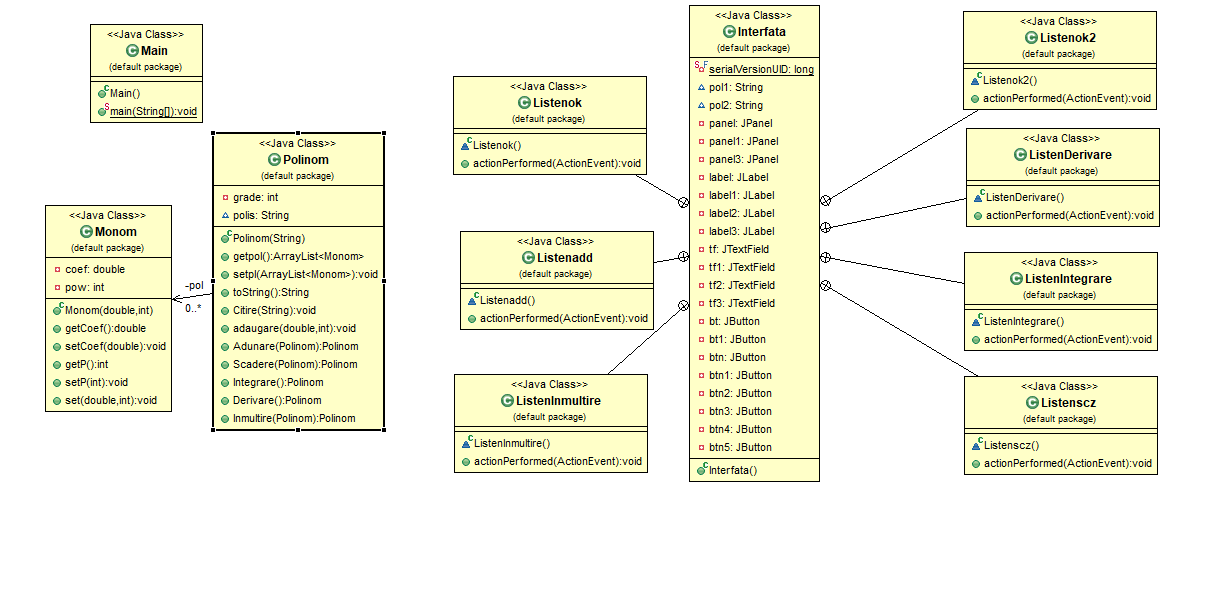
Aplicatia se poate folosi pentru realizarea operatiilor de adunare, scadere,inmultire,derivarea si integrarea intre primul polinom si al doilea polinom, impartirea nefiind inca implementata.

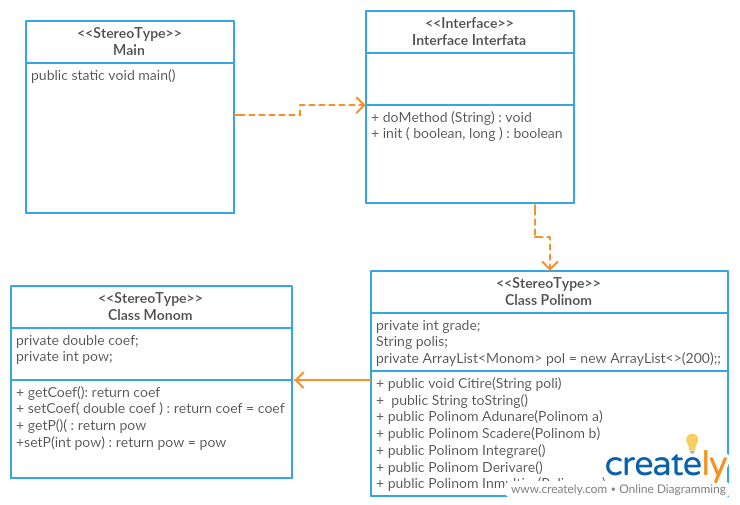
Polinomul introdus nu trebuie sa fie neaprat ordonat dupa puteri deoarece am facut o metoda care face asta deja iar daca se introduc mai multe monoame de acelasi fel se va face o sintetizare la un singur monom cu coeficientii adunati.

Totusi, cazul de mai sus este putin probabil si adesea lucruri neasteptate pot sa apara. Dintre acestea se remarca :

* Introducerea de text, fara semnificatie numerica, in campurile test
* Introducerea de necunoscute suplimentare(pe langa x)
* Folosirea altor cuvinte cheie decat cele specificate
* Scriere corecta a polinoamelor, dar moduri sintactice care difera putin de la user la user( de ex. 2x + 2  vs. 2x^3+5x )
* Exemplu: “x^5+5x^3+x^1+x^0+1**.** Orice deviere de la format poate duce la introducerea unui alt polinom decat cel droit.

**Proiectare**

Diagrama ****



Clasa Monom - clasa fundamentala a proiectului ce defineste structura de monom, entitate ce contine doar un coeficient o putere si, optional, un String pentru a printa rapid monomul. Tot aici avem si constructorul default a unui monom dar si cel explicit si set-erele si get-erele adecvate.

* Clasa Polinom - permite crearea polinoamelor ca liste de monoame, entitati si mai fundamentale, dar contine si partea sortare si minificare a acestora. In acest fel se trateaza unul din cazurile de utilizare mai deosebite, cand utilizatorul nu respecta in totalitate formatul default al scrierii unui polinom. Tot aici, se afla si operatiile fundamentale: adunare, scadere, imultire, impartire, derivare, integrare .
* Clasa Interfata - cu ajutorul ei se creeaza interfata grafica folosita in tot proiectul.
* Clasa Main – in aceasta clasa se afla maninul proiectului, unde se apeleaza interfata definitea in clasa Interfata
* Clasele ce se leaga la clasa Interfata reprezinta implementarea pentru butoane si a casetelor de text.

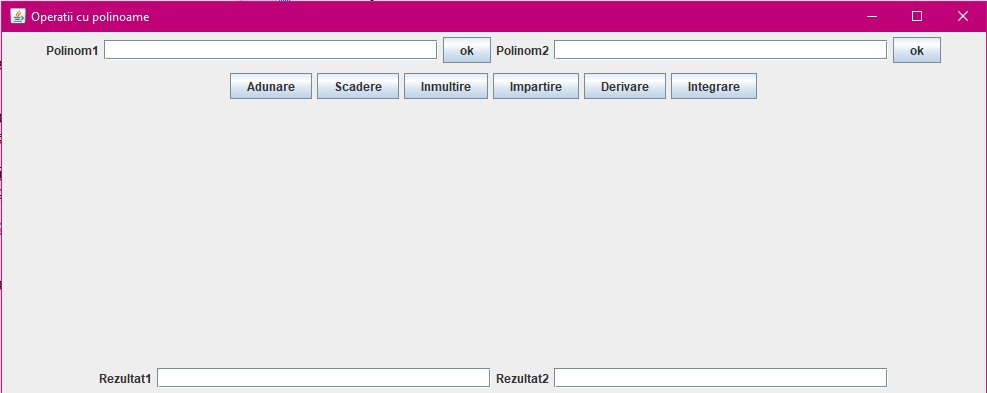
**Implementarea**

Clasa Monom contine metodele getCoef (), getPt(), setCoef (double), setPt(int) pentru settere si gettere, constructorii Monom(double,int).

Clasa Polinom contine constructorii Polinom(String) si Polinom(List<Monom>) Citirea unui string se face prin metoda Citire(string) prin care se citeste un string si cu ajutorul split-ului se delimiteaza monoamele. Sortarea monoamelor in polinoame se face prin metoda ArrayList<Monom> sort(ArrayList<Monom>) in functie de gradul monoamelor. Avem ca si settere si gettere ArrayList<Monom> getpol() si setpl (ArrayList<Monom>pol). Operatiile importante din aceasta clasa sunt:

String t0String(), Citire(String pol), Adunare(Polinom a), Scadere(Polinom b), DerivarePolinom(), Integrare() ,Imultire(Polinom a).

Clasa Interfata cintine interfata GUI a programului.



In prima casuta de text se introduce polinomul intai, iar in a doua casuta de text se introduce al doilea polinom, putoanele ok seteaza polinoamele, iar butoanele de adunare, scadere, imultire, impartire, derivare si integrare au rolul de a efectua operatiile ce le poarta numele asupra celor doua polinoame. In casuta de text rezultat 1 se afiseaza rezultatul operatiilor, iar a doua casuta de text rezultat 2 a fost implementata in cazul impartirii cu rest. Adunarea, scaderea, imultire si impartire efectueaza operatii asupra celor doua polinoame, adica face operatiile de adunare si scadere intre cele doua polinoame, iar operatiile de derivare si integrare efectueaza operatii doar asupra primului polinom.

### **Rezultate**

Cu exceptia unor mici cazuri cand programul este fortat la in zona de input-uri, programul nu va rula corespunzator deoarece nu exista parte de parsare pentru eventualele string-uri aleatoare.

In rest, atat rezultatele, cat si valorile se afla in grila celor asteptate, programul avand un oarecare grad de flexibilitate la nivelului modului de procesare a polinoamelor si parsare a expresiilor speciale.

**Concluzii si dezvoltari ulterioare**

Una dintre concluzii este ca liste gestioneaza mult mai bine memoria calculatorului decat vectorii desi timpul de acces este mai ridicat.

Ca si dezvoltari ulteriore, ideea de a converti dintr-o baza polinomiala intr-o alta pare foarte atractiva, necesitand algoritmi pentru manipularea matricilor de transformare a bazelor. Desigur, o baza generala ar pune mari probleme de implementare, dar si din punct de vedere matematic, asa ca implementarea unor baze standard(Laplace, Newton, Legendre) pare o alternativa fezabila.

Ce am invatat este rolul pe care separarea obiectivelor il are chiar la inceputul problemei. Nu o data, a trebuit sa elimin bucati semnificative de cod fiindca incercam sa compensez lipsa de rigurozitate din faza de planificare cu clauze suplimentare.