

Tehnici de programare

Tema 1

Operatii pe polinoame



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Alexandru Madalina-Elena

Grupa: 30228

Anul 2

Cuprins:

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

1. Obiectivul temei

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

Consideratii teoretice: 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.

Polinoamele sunt construite din termeni numiti monoame, care sunt alcatuite dintr-o constanta (numita coeficient) inmultita cu una sau mai multe variabile. Exponentul unei variabile dintr-un monom este egal cu gradul acelei variabile in acel monom. De exemplu, $3x^2$ este un monom care are coeficientul 3 si gradul 2.

Un polinom este o suma de unul sau mai multe monoame. De exemplu, urmatoarea expresie este un polinom: $2x^3+5x-3$. Cand un polinom este dispus in ordinea naturala, el are termenii de grad mai mare înaintea celor de grad mai mic.

Proprietati:

- Suma a 2 polinoame este un polinom
- Produsul a 2 polinoame este un polinom
- Derivata unui polinom este un polinom
- Primitiva unui polinom este un polinom

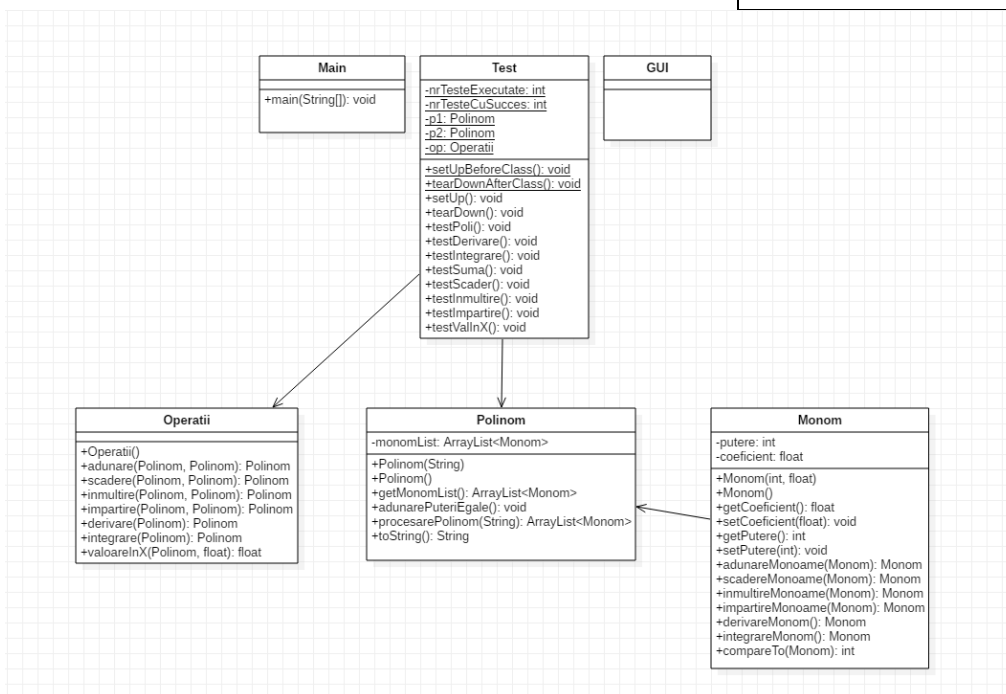
Abordarea problemei: Am proiectat o interfata grafica pentru o aplicatie care primeste ca variabile de intrare doua siruri de caractere de forma $a_1x^n+a_2x^{(n-1)}+...+a_{n-1}x+a_n$ pentru care avem 7 optiuni de operatii: valoarea intr-un numar, derivata, integrala, suma celor doua, diferenta lor, produsul si raportul lor. Acestea pot fi introduse cu puterile in orice ordine, iar pentru monoamele de gradul 0 se poate omite " x^0 " si pentru cele ce au coeficientul "1" sau "-1" se poate omite precizarea acestuia.

2. Analiza

Scenariu de utilizare:

- Nu este obligatoriu sa se execute toti acesti pasi!
- Operatiile pe un polinom se pot face si la introducerea doar unuia dintre ele, dar pentru ca cele pe doua polinoame sa functioneze, ambele trebuie specificate!
- Forma naturala este generata si automat la alegerea unei optiuni de calcul!
- Introducerea primului polinom
- Se apasa butonul *OK* pentru a se genera forma naturala a primului polinom
- Introducerea celui de-al doilea polinom
- Se apasa al doilea buton *OK* pentru a se genera forma naturala a celui de-al doilea polinom
- Se introduce o valoare in care se doreste a fi evaluat primul polinom si se apasa *OK*
- Se introduce o valoare in care se doreste a fi evaluat al doilea polinom si se apasa *OK*
- Se apasa butoanele de derivare si integrare
- Se apasa butonul de adunare
- Se apasa butonul de scadere
- Se apasa butonul de inmultire
- Se apasa butonul de impartire

Diagrama UML



3. Proiectare

Descriere clase:

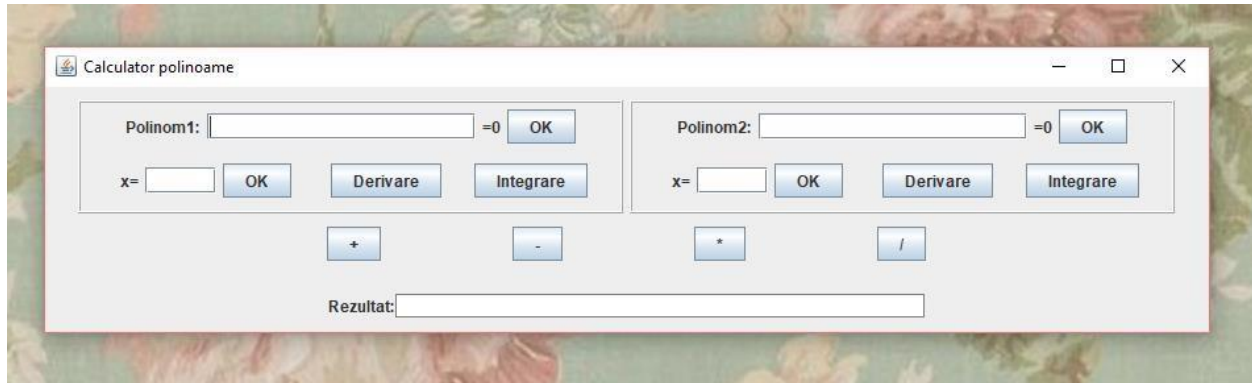
- *Main*: In aceasta clasa se creeaza un nou obiect GUI, care va fi interfata vizuala a aplicatiei
- *GUI*: Aceasta este clasa in care se structureaza interfata vizuala; interfata este formata dintr-un panel mare, in care sunt introduse cele 3 parti esentiale, *introducerePolinoame*, *optiuniCalcul* si *afisorRezultat*:
 - *introducerePolinoame* este un panel alcatuit din 2 paneluri, unul pentru fiecare polinom, acestea continand la randul lor cate 2 paneluri, unul pentru introducerea polinomului(un text field si un buton de ok pentru generarea polinomului introdus in forma sa naturala) si unul pentru optiunile de calcul separate pe polinom(valoarea sa intr-un numar dat, derivare si integrare);
 - *optiuniCalcul* este un panel cu 4 butoane, fiecare reprezentand o optiune de calcul intre cele 2 polinoame introduse, adunare, scadere, inmultire si impartire;
 - *afisorRezultat* este un panel ce contine un text field in care nu se pot introduce date si in care este afisat rezultatul ultimei operatii cerute
- *Monom*:
 - Clasa ce reprezinta monoamele polinomului
 - Are 2 attribute: *putere(int)* si *coeficient(float)*
 - Sta la baza polinomului
 - Contine mai multe metode, cum ar fi *gettere* si *settere* pentru *coeficient* si *putere* (*getCoeficient*, *getPutere*, *setCoeficient*, *setPutere*), operatii pe monoame(*adunare*, *inmultire*, *derivare*, *integrare*, *impartire*), constructori si o metoda pentru sortarea lor
 - *Metode*:
 - *public int getPutere()*, *public float getCoeficient()*: metode ce returneaza puterea, respective coeficientul unui monom
 - *public void setPutere(int putere)*, *public void setCoeficient(float coeficient)*: metode pentru a fixa puterea, respectiv coeficientul unui monom
 - constructori: doua metode, una cu doi parametri (unul de tipul float si unul de tipul int) pentru a se initializa cele doua attribute, si una fara parametri, in care attributele sunt setate la 0
 - *public Monom adunareMonoame(Monom m)*: se aduna coeficientii monoamelor in cazul in care au puteri egale si returneaza un monom
 - *public Monom derivareMonom()*: se deriveaza monomul dupa regula matematica de derivare a monoamelor: $(c \cdot x^p) = (c \cdot p) x^{p-1}$ si returneaza un monom

- public Monom integrareMonom(): se integreaza monomul dupa regula matematica de derivare a unu monom: $(c * x^p) = (c / (p + 1)) x^{p+1}$ si returneaza un monom
- public Monom inmultireMonoame(Monom m): se inmultesc 2 monoame si se returneaza rezultatul inmultirii lor: $(c1 * x^{p1}) * (c2 * x^{p2}) = c1 * c2 * x^{p1+p2}$
- public Monom impartireMonoame(Monom m): se impart 2 monoame si se returneaza rezultatul impartirii lor: $(c1 * x^{p1}) / (c2 * x^{p2}) = (c1 / c2) * x^{p1-p2}$
- public int compareTo(Monom o): metoda ce ajuta sortarea monoamelor unui polinom in ordine descrescatoare a puterilor pentru a obtine forma lor naturala
- *Polinom:*
 - Contine metodele de procesare ale polinoamelor de la intrare
 - Are un singur atribut, un array list de monoame
 - Contine mai multe metode, precum un getter, 2 constructori, o metoda de procesare a unui String, o metoda de generare a formei naturale a unui polinom, o metoda de convertire a listei de monoame intr-un String
 - *Metode:*
 - doi constructori: unul care primeste ca parametru un String pe care il va converti intr-o lista de monoame si unul care nu primeste niciun parametru si seteaza lista de monoame initiala ca fiind nula
 - public ArrayList<Monom> procesarePolinom(): metoda ce preia Stringul introdus in text field si il converteste in monoame; se parcurge tot stringul, iar in cazul intalnirii unui + sau a unui - se analizeaza urmatoarele caractere pentru a se stabili coeficientul si puterea monomului curent; exista cazuri speciale precum lipsa caracterului ^ sau lipsa coeficientului care sunt luate in considerare
 - public void adunarePuteriEgale(): metoda care normalizeaza polinomul introdus; aduna coeficientii monoamelor ale caror puteri apar de mai multe ori in lista de monoame si elimina valorile fara importanta(ex. "0")
 - public String toString(): metoda ce converteste lista de monoame intr-un String ce descrie polinomul in forma sa naturala(ex. output: $2x^3-2x+1$) pentru a putea fi afisat
- *Operatii:*
 - Cea mai importanta clasa din proiect
 - Contine metodele cu operatiile ce se pot face pe polinoame
 - Un constructor, patru operatii ce se pot face intre doua polinoame si 3 ce se pot executa pe un singur polinom, separate

- *Metode:*
 - public Polinom adunare(Polinom p1, Polinom p2): metoda de adunare a 2 polinoame; se parcurg, pana cand se ajunge la sfarsitul cel puțin al unui polinom, listele de monoame ale ambelor polinoame, iar in cazul intalnirii unor monoame cu puteri egale se apeleaza functia de adunare a monoamelor si se adauga rezultatul in polinomul ce retine rezultatul; la sfarsit se verifica daca mai exista membri in lista vreunui polinom si, in caz ca da, se adauga la sfarsitul polinomului rezultat monoamele ramase din acesta
 - public Polinom scadere(Polinom p1, Polinom p2): metoda de scadere a 2 polinoame; polinomul rezultat primeste rezultatul adunarii dintre descazut si opusul scazatorului(scazatorul inmultit cu -1)
 - public Polinom derivare(Polinom p): metoda de derivare a unui polinom; se parcurge fiecare monom din polinom si se deriveaza, apelandu-se metoda de derivare a monomului; se adauga monomul rezultat la polinomul rezultat
 - public Polinom integrare(Polinom p): metoda de integrare a unui polinom; se parcurge fiecare monom din polinom si se integreaza, apelandu-se metoda de integrare a monomului; se adauga monomul rezultat la polinomul rezultat
 - public Polinom inmultire(Polinom p1, Polinom p2): metoda de inmultire a 2 polinoamelor: fiecare monom din primul polinom se inmulteste cu monoamele celui de-al doilea, iar polinomul rezultat se adauga la polinomul final
 - public String impartire(Polinom p1, Polinom p2): metoda de impartire a 2 polinoame: returneaza un string de forma "cat = <valoarea catului>, rest = <valoarea restului>"; initial, restul este deimpartitul, urmand ca, pana cand restul ajunge sa aiba un grad mai mic decat impartitorul, impartitorul sa fie inmultit cu raportul dintre monomul cu grad cel mai mare din deimpartit si monomul cu grad cel mai mare din impartitor inmultit cu (-1); rezultatul acestui raport este adaugat la cat, iar rezultatul scaderii dintre rest si rezultatul inmultirii este noul rest;
 - public float valoareInX(Polinom pfloat x): se calculeaza matematic valoarea fiecarui monom in x, se parcurge fiecare monom al polinomului, si se adauga intr-o variabila rezultat de tip float care va fi returnata
- Test: clasa de Junit Test in care se testeaza toate operatiile disponibile in aplicatie
 - Clasa are douasprezece metode, dintre care una este pentru setarea conditiilor initiale, inaintea executarea testelor, una este pentru setarea actiunilor ce se iau in urma tuturor testelor, una pentru initializarile ce trebuie facute inainte de fiecare test, una ce delimiteaza terminarea unui test, iar opt reprezinta testele propriu-zis,

dintre care sapte reprezinta testele operatiilor pe polinoame mentionare anterior, iar una testeaza procesarea unui polinom si aducerea sa la forma naturala

4. Implementare si testare



- Cele 2 dreptunghiuri din partea de sus sunt destinate calculului separat pe polinoame; in text field-uri se introduce polinoame, iar butonul okay le preia, le proceseaza, si le reafiseaza in forma lor naturala, lucru care se va face si fara sa fie vizibil, la apelarea oricarei operatiei, chiar daca nu se apasa butonul ok
- Prima operatie particulara pentru polinoame este valoarea polinomului intr-un numar; numarul se introduce in text field-ul de dupa "x=", iar rezultatul va fi afisat dupa apasarea butonului ok
- A doua si a treia operatie(derivarea si integrarea) se vor face automat dupa apasarea butoanelor respective cu denumiri sugestive, iar rezultatul va fi afisat
- In a doua treime a ferestrei sunt butoanele pentru operatiile intre cele 2 polinoame introduse, adunare("+"), scadere("-"), inmultire("*") si impartire("/"); aceste operatii se vor face automat dupa apasarea butoanelor, iar polinoamele, respective Stringul rezultate vor fi afisate
- In ultima treime a ferestrei este afisorul rezultatului operatiei curente
- Operatiile ce se pot face pe un singur polinom(cele din prima treime a ferestrei) se pot face doar dupa introducerea unui singur polinom, in timp ce celelalte necesita introducerea ambelor polinoame

Testare: Am testat aplicatia folosind Junit Tests, avand in final facute 8 teste, pentru procesarea si aranjarea polinomului in forma sa naturala, pentru calculul valorii polinomului intr-un numar, pentru derivare, pentru integrare, pentru suma, pentru scadere, pentru inmultire si pentru impartire. In final, pentru polinoamele alese de mine, acestea au fost executate cu succes.

5. Rezultate obtinute

Pentru a putea folosi aplicatia, utilizatorul trebuie sa aiba JVM instalat si trebuie sa se respecte regulile stabilite de introducere a polinoamelor. Nu trebuie sa existe niciun spatiu intre caractere, singurele caractere acceptate sunt x , $+$, $-$, $^$ si cifrele de la 0 la 9. Pentru a folosi aplicatia, nu trebuie in mod obligatoriu sa se foloseasca butonul OK pentru a genera forma naturala a polinomului, ci se poate apela direct la operatii deoarece in momentul acestui eveniment se va genera automat forma naturala a rezultatului, indiferent de forma operanzilor introdusi.

Aplicatia este functionala in majoritatea cazurilor, cu exceptia situatiilor in care exista un caracter dintre $+$, $-$ si $^$ duplicat sau o combinatie formata din acestea introduse consecutiv, in care nu este depistata o eroare de introducere a polinoamelor. De asemenea, pot exista exceptii nedepistate de mine.

6. Concluzii

Aplicatia are implementate operatii de baza pe unul sau doua polinoame, introducerea lor fiind limitata de anumite conditii. Aceasta nu presupune nici metode foarte complexe si nici o interfata foarte complicata. Este usor de folosit si inteles pentru majoritatea utilizatorilor.

Aplicatia poate fi imbunatatita in multe moduri: se poate mari numarul de polinoame, se pot aduga metode, de exemplu calculul solutiilor unui polinom, generare de grafice, se poate largi accesibilitatea prin micșorarea numarului de conditii ce trebuie respectate la introducerea polinoamelor.

7. Bibliografie

1. Wikipedia
2. Diferite referate despre polinoame

3. Tutoriale de Java
4. Laboratoare de POO din semestrul 1
5. Diagrama a fost desenata folosind starUML