**Calculator de polinoame**

Logo, company name

Description automatically generated

Student: Alexa Andrei Stefan

Group: 30225

1. **Obiectivul temei**

Proiectarea si implementarea unui calculator de polinoame cu o interfata grafica dedicata prin care utilizatorul poate sa insereze polinoame, sa selecteze operatia matematica si sa vada rezultatul.

Sub-obiective:

* Analiza problemei si idetificarea cerintelor
* Proiectarea calculatorului de polinoame
* Implementarea calculatorului de polinoame
* Testare calculatorului de polinoame

De ce facem acest calculator de polinoame?

Deoarece rezolvarea diferitelor operatii asupra polinoamelor de grade mai mari, pe hartie, este dificila si deloc eficienta.

Deoarece rezolva o problema des intalnita in domeniul nostru de studii.

Deoarece este un bun inceput in proiectarea unor probleme mai complexe.

1. **Analiza problemei, modelare, scenario si cazuri de utilizare**
   1. **Analiza problemei**

Un polinom P cu necunoscuta X este definit in felul urmator:

unde:

* + - *a0*, *a1*, …, *an* reprezinta coeficientii polinomului
    - *n* reprezinta gradul polinomului

Un monom este un tip de polinom cu un singur termen.

Un polinom este o lista de monoame.

Un monom este definit de un coeficient si un exponent(aX^b) unde:

a - coeficientul

b – puterea

Daca stim sau aflam coeficientul si puterea, putem recrea monomul sau sa il folosim pentru operatii.

Unul dintre modurile prin care putem sa aflam monoamele si sa izolam coeficientul si exponentul este folosind regex.

* 1. **Modelare**

Interfata ii va prezenta utilizatorului 2 campuri pentru cele doua polinoame asupra carora se doreste a fi facuta operatia.

Utilizatorul va avea access la butoanele necesare pentru a introduce un polinom, sau utilizatorul poate selecta campul droit si sa utilizeze tastaura pentru a scrie un polinom.

Rezultatul o sa apara intr-un camp separat specific.

Primul polinom va fi introdus in primul camp, iar cel de-al doilea polinom, in al doilea camp.

Unele dintre operatii, precum cea de integrare sau cea de derivare se vor face doar asupra unuia dintre polinoame si anume cel din primul camp, specific primului polinom.

Rezultatul este calculat si afisat dupa ce utilizatorul introduce un polinom si selecteaza operatia pe care vrea sa o faca.

Operatiile sunt urmatoarele:

* Adunare(Polinom1, Polinom2)
* Scadere(Polinom1, Polinom 2)
* Derivare(Polinom1)
* Integrare(Polinom1)
  1. **Scenarii**

Cel mai bun scenario si, de asemenea cazul in care aplicatia trebuie sa calculeze un rezultat correct este atunci cand utilizatorul introduce crect polinoamele in cele doua campuri specifice sub forma corecta.

Un polinom correct ar trebui sa arate in felul urmator:

aX^b + cX^(b-1) +… - dX^(b-n)

Programul va ordona polinomul rezultat din cele doua asupra carora s-au efectuat operatii, de la monomul cu cea mai mare putere la cel cu cea mai mica putere, desigur adunand monoamele cu aceeasi putere.

* 1. **Cazuri de utilizare**

Cazul de utilizare si motivul folosirii unui calculator de polinoame este evident:

Utilizatorul vrea o modalitate simpla si de incredere pentru a face operatii asupra polinoamelor, care sa fie de asemenea mai rapida decat rezolvarea operatiilor pe hartie.

1. **Proiectare**
   1. **Decizii de proiectare**

Ideea de la care am pornit: Un calculator de polinoame simplu de folosit si care face operatiile correct.

Interfata trebuie sa fie user-friendly sisa contina butoane pentru toate numerele/caracterele necesare pentru a scrie un polinom correct si un buton de delete in caz de greseli la inserarea input-ului. Interfata permite de asemnea inserarea polinoamelor de la tastaura.

Proiectarea Aplicatiei este bazata pe clasica structura Model-View-Controller.

Acest model ajuta la impartirea Aplicatiei in pachete/clase mai mici cu care este usor de lucrat si pentru a avea un proiect bine structurat, avand un scop clar. Acest model face de asemenea cautarea problemelor si a implementarii unor noi aspecte mult mai usoara.

* 1. **Diagrama UML**

Clasa Polynomial nu poate exist afara clasa Monomial, polinoamele fiind definite ca o lista de monoame. Operatiile din clasa Model sunt aplicate pe polinoame. Metoda de parsing este folosita pentru a parsa input-ul de la utilizator in monoame (definite prin coeficient si exponent). Controller primeste evenimente de la Interfata.

**Graphical user interface

Description automatically generated with medium confidence**

* 1. **Structuri si tipuri de date**

Concepte importante pentru acest program:

Polinomul este vazut ca si un ArrayList de monoame. Folosind foreach putem itera prin toate monoamele polinomului si sa executam operatii pe ele.

Gradul unui polinom este exponentul cu cea mai mare valoare al polinomului.

Fiecare monom are un coeficient si un exponent.

Coeficientii sunt de tipul int asa ca unele operatii precum integrare sau derivare s-ar putea sa nu afiseze in totalitate correct. (pentru a rezolva aceasta problema putem pune double la coeficienti).

Exponentii sunt considerati intregi.

Cateva tipuri de date extra:

Campurile text din interfata sunt interpretate in cod ca si niste variabile care memoreaza input-ul ca si un string.

* 1. **Proiectarea claselor, interfetei si pachetelor**

Ideea cu care am inceput aici:

Clasele trebuie sa fie compacte, bine organizate, avand metode care excuta operatii specific si sunt numite orientativ.

Pachetele aplicatiei sunt urmatoarele:

Pachetul main care include toate sursele proiectului.

Pachetul Controller care include clasa Controller.

Pachetul Model care include clasele ce executa sarcini precum operatii/parsing-ul datelor.

Pachetul View include Interfata.

Proiectul final continue urmatoarele clase: Controller, Monomial, Polynomial, Model, Parsing, Interface si test.

Principalele idei ale claselor sunt urmatoarele:

In clasa Controller fac legatura dintre back-end si front-end. Metodele din clasa Controller sunt apelate cand utilizatorul foloseste interfata (de exemplu: utilizatorul apasa buton de adunare => metoda adunare este apelata in Controller, iar logica acestei metode este apelata dintr-o alta clasa numita Model).

Clasa Model contine metodele pentru operatiile de adunare, scadere, derivare si integrare.

Clasa Parsing contine metodele necesare pentru parsing si pentru a interpreta datele citite din campurile text ale interfetei. Aceasta clasa utilizeaza regex(pattern matching/creating) pentru a separa input-ul in grupuri care contin informatia necesara pentru a crea polinoamele in interiorul memoriei programului(stocam polinoamele ca o lista de monoame, fiecare avand un coeficient si un exponent).

Clasa Monomial contine caracteristicile unui monom si o metoda care transforma monomul inapoi intr-un string care poate fi afisat si citit de catre utilizator.

Clasa Polynomial contine caracteristicile unui polinom(un polinom este un ArrayList de monoame) si, de asemenea cateva metode care ordoneaza monoamele in ordinea corecta (de la cel mai mare exponent la cel mai mic) si adauga monoamele care au acelasi exponent.

Clasa test contine initializarea aplicatiei si cateva teste specifice.

* 1. **Algoritmi folositi**

In acest capitol voi descrie cativa din agloritmii folositi pe care eu ii consider important, folosid si putin pseudo-cod pentru ca este mult mai usor de urmarit.

Adunare:

Fiecare monom din primul polinomeste comparat cu fiecare monom din al doilea polinom. Daca au acelasi exponent, coeficientii lor sunt adunati, iar exponentul rezultatului ramane la fel (dupa ce facem aceasta operatie setam valoarea lui skip la TRUE). Cu noul coeficient si exponent, creez un nou monom care este adaugat la polinomul rezultat.Tot ce mai este de facut acum este sa adaugam rezultatului toate monoamele din polinomul 1 si 2 daca nu au fost deja adaugate in pasul anterior (daca skip este inca FALSE).

Scadere:

Acest algoritm este practic doar o adunare cu semnele de + si – ale celui de-al doilea polinom inversate. Schimbarile facute algoritmului de adunare sunt:

-In loc sa adun termenii cu acelasi exponent, scad din cel de al doilea pe primul.

-Monoamele din al doilea polinom nefolosite la pasul anterior, vor fi adaugate rezultatului cu semn schimbat.

Derivare:

Pentru derivare, iteram prin toate monoamele polinomului dat si inmultim coeficientul cu exponentul curent, iar apoi noul exponent al monomului va fi exponentul curent -1.

Integrare:

Pentru integrare, iteram prin toate monoamele polinomului dat si impartim coeficientul curent cu exponentul curent + 1, iar noul exponent va fi obtinut prin cresterea cu 1 al celui vechi.

Alte metode si algoritmi folositi:

Metoda toString imi construieste un String vizibil pentru utilizator.

Metoda parsePolynomial foloseste un model pentru acrea grupuri si sa izoleze partile inputului pentru a parsa datele in informatiile de care noi avem nevoie pentru a crea monoame corecte in programul nostrum.

Folosesc Collection sort pentru a sorta monoamele in ordine descrescatoare in functie de exponent. Acesta este foarte folositor pentru a afisa rezultatul in ordinea clasica in care un polinom este scris (de la cel mai mare, la cel mai mic exponent).

addMonomial : -aceasta metoda va lua polinoamele si va aduna monoamele daca acestea au aceeasi puetere.

* 1. **Interfata**

Graphical user interface

Description automatically generated

Pentru a crea interfata am folosit Eclipse. Interfata este implementata cu java swing.

Interfata trebuie sa fie simpla si usor de folosit pentru toti utilizatorii, include butoane prin care utilizatorul poate sa introduca date in campurile text, dar accepta de asemenea introducerea datelor direct de la tastaura .

Interfata apeleaza operatia care se doreste a fi executata cand utilizatorul apasa butonul specific acelei operatii.

Input-urile (polinoamele) sunt introduce in cele doua campuri text cu nume specifice (Primul polinom si Al doilea polinom), iar rezultatul este afisat in campul denumit “Rezultat”.

1. **Implementare**

Implementarea si deciziile pe care le-am luat pentru aceasta au fost deja discutate in capitolele anterioare (descrierea claselor, metodelor si interfetei), deci o sa rezum procesul de creare al aplicatiei aici.

Prima parte pe care am create-o a fost interfata. Pentru asta am folosit Eclipse (Java Swing), am detaliat procesul acesta in capitolul Interfata.

Urmatoarea parte a proiectului a fost cotroller-ul care stabileste ce se intampla cand se interactioneaza cu interfata. Am definit niste metode si niste corpuri temporare pentru ele. Bazat pe acestea am inceput sa creez metodele de care aveam nevoie.

Am create clasa Monomial si Polynomial, iar apoi o clasa care sa contina codul pentru parsarea input-ului de la utilizator, iar apoi am inceput sa implementez algortimii pentru operatii.

Pasul final a fost testarea intregii aplicatii, dar nu am folosit testarea unitara.

Cateva din metodele principale sunt:Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Metodele de integrare si derivare sunt destul de asemanatoare, diferenta fiind doar la calcularea rezultatului, la fel si pentru metodele de adunare si scadere unde diferenta este doar semnul.

Text

Description automatically generated

Folsesc acest algoritm pentru a elimina monomul in cazul in care coeficientul acestuia este 0.

1. **Rezultate**

Testarea aplicatiei a fost facuta doar incercand sa folosim aplicatia ca un utilizator normal pentru a observa rezultatele pentru diferite input-uri si operatii.

Graphical user interface

Description automatically generated

1. **Concluzii**

Acest proiect a fost o provocare noua pentru mine, deoarece am invatat sa lucrez cu diferite informatii noi fiind prima data cand lucrez cu Regex.

As fi putut sa implementez si celelalte operatii cerute in tema, iar pe langa acestea mi s-ar parea interesant sa incerc sa implementez si ridicarea la putere sau alte operatii asemanatoare.

Am descoperit de asemenea ca imi place sa lucrez cu Interfata.

1. **Bibliografie**

-Documentatia pentru proiect pusa la dispozitie

-Stack Overflow pentru Regex, de asemenea anumite sfaturi pentru implementarea operatiilor pe polinoame

[Polynomial Java Array add subtract - Stack Overflow](https://stackoverflow.com/questions/42198612/polynomial-java-array-add-subtract)

[Java regex, separate degree & coeff of polynomial - Stack Overflow](https://stackoverflow.com/questions/28859919/java-regex-separate-degree-coeff-of-polynomial/44188848)

<https://jenkov.com/tutorials/java-regex/matcher.html>