Universitatea Tehnică din Cluj-Napoca

Facultatea de Automatică și Calculatoare

Calculator Polinomial

Matei Anechitei

Grupa 30221

1. Obiectivul temei

Proiectati si implementati un calculator polinomial cu o interfata grafica dedicata prin care utilizatorul poate sa introduca polinoame, sa selecteze operatia ce trebuie realizata (adunare, scadere, inmultire, impartire, derivare, integrare) si sa afiseze rezultatul.

Detalii:

-Polinoamele ce trebuie introduse sunt cu o singura variabila, iar coeficientii sunt intregi.

-Primul pas il reprezinta introducerea polinoamelor de intrare in aplicatie prin intermediul interfetei grafice si selectarea uneia din cele 6 operatii numite anterior.

-Adunarea, scaderea, inmultirea si impartirea folosesc 2 polinoame, iar integrarea si derivarea folosesc doar primul polinom.

-In cazul in care polinoamele au fost introduse corespunzator, se va afisa in interfata grafica rezultatul obtinut.

-In cazul adunarii, scaderii, inmultirii si derivarii, rezultatul va fi afisat cu coeficienti intregi, iar in cazul impartirii si integrarii, cu coeficienti reali cu 3 cifre dupa virgula.

-Impartirea va afisa 2 polinoame (un cat si un rest), celelalte operatii vor afisa un singur polinom.

-In cazul unei erori de introducere a polinoamelor, mesajul “Invalid input!” va aparea in spatiul rezervat rezultatului.

-Interfata grafica este realizata dupa modelul standard MVC (Model-View-Controller).

2. Analiza problemei, modelare, scenarii, cazuri de utilizare

Pentru a putea incepe partea de proiectare, trebuie sa aducem la cunostinta notiunea matematica de polinom si proprietatile de baza ale acestuia.

Definitie:

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 cu o constanta pozitiva intreaga. (Figura 1)



Operatii:

1. Adunarea

Adunarea a doua polinoame se face prin adunarea coeficientilor termenilor care au aceeasi putere.

1. Scaderea

Scaderea poate fi considerata o adunare a primului polinom cu opusul celui de-al doilea, obtinut prin negarea coeficientilor sai.

1. Inmultirea

Inmultirea polinoamelor este realizata inmultind fiecare monom al primului polinom cu fiecare monom din al doilea polinom, apoi adunand toate aceste rezultate. Monoamele pot fi inmultite prin inmultirea coeficientilor si adunarea exponentilor.

1. Impartirea

Impartirea polinoamelor se realizeaza dupa urmatorii pasi:

1. Se ordoneaza polinoamele in ordinea descrescatoare a exponentilor.
2. Se imparte primul monom al primului polinom la primul termen din al doilea polinom si astfel obtinem primul termen al catului.
3. Se inmulteste rezultatul obtinut cu impartitorul si se scade acest produs din deimpartit.
4. Se repeta acest procedeu pana cand gradul monomului cel mai mare din rest este mai mic decat gradului monomului cel mai mare din impartitor.
5. Derivarea

Derivarea este o operatie ce se aplica pe un singur polinom. Derivarea unui polinom se face prin derivarea tuturor monoamelor. Un monom se deriveaza prin inmultirea coeficientului cu exponentul si decrementarea exponentului cu 1.

1. Integrarea

Integrarea, la fel ca si derivarea, se aplica pe un singur polinom. Integrarea unui polinom se face prin integrarea tuturor monoamelor. Un monom se integreaza prin impartirea coeficientului la succesorului exponentului si incrementarea cu 1 a exponentului.

Modelarea datelor:

Polinomale sunt formate dintr-o lista de monoame. Monoamele sunt de un singur tip, cu coeficienti reali, dar la afisare sunt transformati in coeficienti intregi (in functie de operatia aleasa). Astfel, se pastreaza o acuratete sporita in cazul impartirii si a integrarii, fara a renunta la claritate in celelalte cazuri.

3.Proiectare

Aplicatia a fost construita pe baza MVC (Model-View-Controller).

Model

In cadrul structurii MVC, componenta care se ocupa cu partea de modelare este componenta “Model”, in acest proiect fiind alcatuit din mai multe clase (“Monomial”, “Polynomial”, “CalculatorModel”).

Obiectele din clasa “Monomial” sunt caracterizate de exponent (numar intreg) si coeficient (numar real).

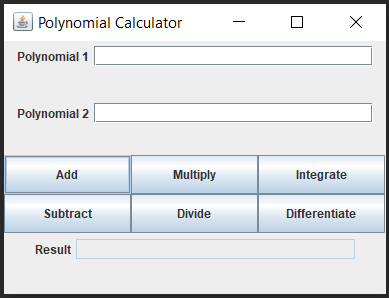
Obiectele din clasa “Polynomial” contin o lista de monoame, asupra carora se poate aplica si o sortare in functie de exponent, astfel incat sa avem ordinea adecvata. Tot in cadrul acestei clase sunt implementate cele 6 operatii si 2 functii care transforma lista de monoame intr-un sir de caractere usor de citit, in functie de nevoia de coeficienti intregi sau reali.

Clasa “ControllerModel” foloseste RegEx si transforma string-urile de intrare in liste de monoame pe care se pot executa operatiile.

View

Componenta “View” reprezinta interfata grafica cu care interactioneaza utilizatorul. Crearea ei a fost realizata cu ajutorul Swing si contine 3 TextField-uri (2 pentru polinoamele de intrare si 1 pentru rezultat) si 6 butoane, cate unul pentru fiecare operatie.

Clasa “View” extinde JFrame si, pe langa elementele de grafica, contine metodele care adauga cate un ActionListener pentru fiecare buton.



Controller

Controller-ul se ocupa cu preluarea string-urilor de intrare din “View”, le trimite “Model”-ului sa le prelucreze in functie de operatia ceruta, apoi actualizeaza interfata grafica. El contine pe langa un constructor si 6 metode de tip Listener pentru fiecare buton.

4.Implementare

In cele ce urmeaza voi prezenta clasele si cele mai importante metode din acestea.

Monomial.java

Clasa Monomial contine obiectele de tip monom, avand coeficient real si exponent intreg.

In afara de setter-e si getter-e, clasa mai contine 2 metode: toStringInt; toStringDouble.

Metoda toStringInt transforma un obiect Monomial intr-un string ce respecta conventiile matematice de scriere a unui monom, afisand-ul ca si cum ar avea coeficient intreg pentru claritate sporita.

Metoda toStringDouble are acelasi rol ca si metoda anterioara, doar ca afiseaza coeficientul sub forma zecimala, cu 3 cifre dupa virgula; metoda este folosita pentru afisarea rezultatului impartirii si a integrarii, unde natura coeficientilor depinde de la caz la caz.

Polynomial.java

Clasa Polynomial contine obiectele de tip polinom, fiind create dintr-o lista de tip ArrayList, alcatuita din obiecte Monomial. Polynomial implementeaza toate metodele corespunzatoare operatiilor, 2 metode pentru construirea string-ului trimis ca rezultat, o metoda de adaugare a monoamelor in lista si o metoda de aranajare a listei.

Metodele add, subtract, multiply, divide, differentiate si integrate sunt metodele ce efectueaza operatiile necesare pe polinoamele de intrare si returneaza rezultatul dorit, dar primesc ca intrare polinoame in forma lor de obiecte, si furnizeaza un raspuns in aceeasi forma, ca obiect de tip Polynomial.

Metodele printPolynomialInt si printPolynomialDouble se folosesc de metodele toStringInt, respectiv toStringDouble, si formeaza un string concatenand corespunzator monoamele din lista pe care il trimit ca rezultat pe interfata grafica.

Metoda arrangePolynomial are grija de mai multe aspecte cu privire la ordinea monoamelor in lista si eliminarea redundantelor. Aceasta elimina din lista monoamelor toate monoamele ce au coeficientul 0, uneste monoamele ce au acelasi exponent intr-un singur element al listei pentru fiecare exponent prezent, si sorteaza lista in ordine descrescatoare a exponentilor, folosind Comparable. In cazul in care lista este goala, adauga un element cu coeficient 0 si exponent 0, pentru a sti ca polinomul este nul, dar exista.

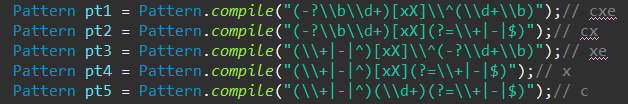
CalculatorModel.java

Clasa CalculatorModel are rolul de a transforma input-ul dat de utilizator sub forma de string in obiecte de tip Polynomial, folosindu-se de RegEx. Contine 2 clase, checkString si createPolynomial.

Metoda checkString elimina spatiile si caracterele “\*” din string, apoi foloseste un pattern de recunoastere a caracterelor care sunt considerate utile si, in cazul in care exista caractere necunoscute, trimite ca output mesajul “Invalid input!”.



Metoda createPolynomial actioneaza daca metoda checkString a confirmat ca string-ul poate fi transformat. Folosind RegEx, identifica toate monoamele dintr-un string care respecta unul din urmatoarele formate: Coeficient(\*)x^Exponent; Coeficient(\*)x; (+-)x^Exponent; (+-)x; Constanta. In urma identificarii unui monom, il introduce intr-o lista de monoame a unui obiect de tip Polynomial, apeleaza la final metoda arrangePolynomial si il trimite inapoi Controller-ului.



De mentionat este ca singura variabila acceptata de catre program este x (non-case sensitive). Orice alta variabila va returna rezultatul “Invalid input!”.

CalculatorView.java

Clasa ControllerView se ocupa de interfata grafica. Ea contine 3 casete etichetate “Polynomial 1”, “Polynomial 2” si “Result” si 6 butoane etichetate dupa numele operatiei dorite. Caseta eticheta “Result” nu poate fi modificata de catre utilizator, ea afisand rezultatul furnizat de catre program.

CalculatorController.java

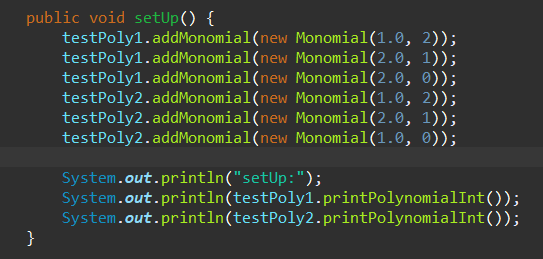
Clasa CalculatorController face legatura intre “View” si “Model”. Ea capteaza string-urile de pe interfata, le verifica cu metoda checkString, creeaza obiecte Polynomial cu ajutorul createPolynomial, apeleaza metoda corespunzatoare operatiei, apoi transforma obiectul rezultat Polynomial prin una din metodele de printare din clasa Polynomial intr-un string cu care actualizeaza interfata grafica.

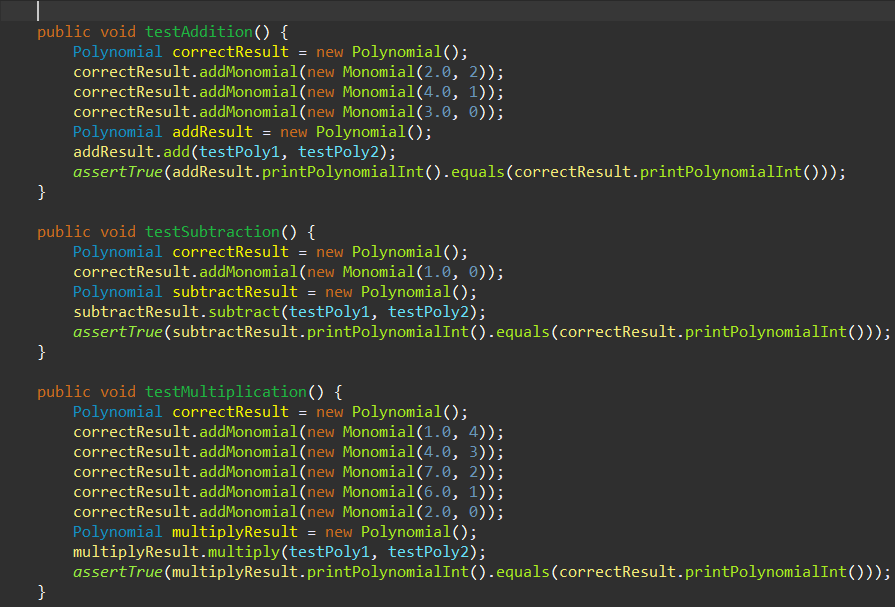


5.Rezultate

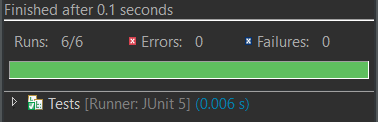
Testarea a fost realizata cu ajutorul Junit intr-o clasa separata numita intuitiv “Tests”, in care am verificat cele 6 operatii, dar si afisarea adecvata a polinoamelor.

Polinoamele de intrare sunt: x^2 + 2x + 2; x^2 + 2x + 1.









6.Concluzii

In urma implementarii acestui progam am invatat sa folosesc RegEx pentru a procesa input-uri sub forma de string-uri, am inteles structura arhitecturii de tip MVC, am exersat folosirea listelor in limbajul Java si am dobandit experienta in materie de programare orientata pe obiecte.

Imbunatatiri posibile:

* acceptarea oricarei litere ca si variabila de intrare;
* verificarea tuturor formelor acceptabile pentru scrierea din punct de vedere matematic al monoamelor (ex: x\*2);
* procesarea polinoamelor cu coeficienti reali;
* modernizarea interfetei grafice;
* mesaje de eroare specifice.

7.Bibliografie

* <https://ro.wikipedia.org/wiki/Polinom>
* <https://www.regextester.com/>
* <https://en.wikipedia.org/wiki/Polynomial_long_division>
* <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>