

DOCUMENTATIE

Tema numarul 1

Popovici Eusebiu-Ionut

Grupa 30224

CUPRINS

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

1. Obiectivul temei

Obiectivul principal al temei este de a realiza un calculator de polinoame care sa lucreze prin intermediul unei interfete grafice, unde putem vedea rezultatele asteptate.

Pasii care trebuie urmati pentru a indeplini obiectivul principal sunt urmatoarii:

- realizarea clasei “polinom”(eventual si a clasei monom), unde trebuie sa ne declaram un Map in care sa stocam polinomul, unde key-ul Map-ului va fi puterea de tip integer si valoarea Map-ului va fi coeficientul de tip double(Capitolul 4).

- realizare clasei “operations”, unde vom scrie metodele care sa ne efectueze operatiile necesare(adunare, scadere, inmultire, impartire, derivare, integrare) si sa ne returneze rezultatul corect sub forma unui polinom. Tot in aceasta clasa am adaugat si o metoda denumita “matching”, unde am extras, folosind un regex, coeficientii dintr-un string si sa ii stocam in polinomul nostru(Capitolul 4).

- realizarea interfetei grafice, de unde vom putea testa operatiile implementate(Capitolul 4).

- realizare clasei de testare unitara, unde vom avea cate un test bun si unul gresit pentru fiecare operatie(Capitolul 5).

- toate aceste clase vor fi implementate in 4 pachete diferite, conform specificatiilor.

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

Aplicatia este structurata astfel incat prin intermediul interfetei sa putem introduce polinoamele dorite, iar prin apasarea unor butoane sa putem vedea rezultatele dorite. Prin apasarea butoanelor, in spatele interfetei se vor executa niste procese de back-end, reprezentate de urmatoarele operatii:

1. Adunarea a doua polinoame:

- utilizatorul introduce coeficientii celor doua polinoame;
- sistemul aduna cele doua polinoame si afiseaza rezultatul.

2. Scaderea a doua polinoame:

- utilizatorul introduce coeficientii celor doua polinoame;
- sistemul scade cele doua polinoame si afiseaza rezultatul.

3. Inmultirea a doua polinoame:

- utilizatorul introduce coeficientii celor doua polinoame;
- sistemul inmulteste cele doua polinoame si afiseaza rezultatul.

4. Impartirea a doua polinoame:

- utilizatorul introduce coeficientii celor doua polinoame;
- sistemul imparte cele doua polinoame si afiseaza rezultatul.

5. Derivarea unui polinom:

- utilizatorul introduce coeficientii a unui polinom, intr-un singur text-field, la alegere;
- sistemul deriveaza polinomul si afiseaza rezultatul.

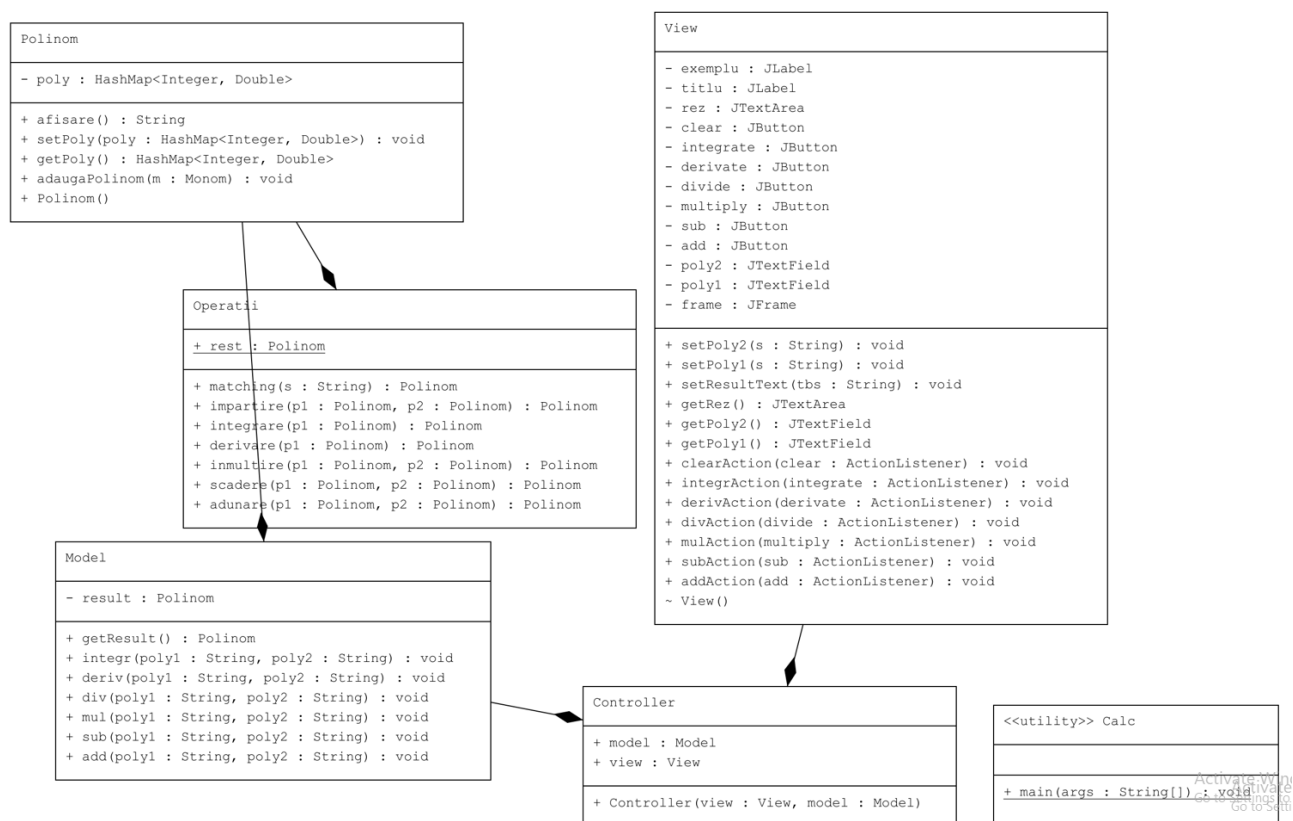
6. Integrarea unui polinom:

- utilizatorul introduce coeficientii a unui polinom, intr-un singur text-field, la alegere;
- sistemul integreaza polinomul si afiseaza rezultatul.

3.Proiectare

Proiectarea OOP a aplicatiei va aborda principiile programarii orientate pe obiecte,functionalitatile fiind incapsulate in clase,care se vor afla in diferite pachete pentru ca proiectul sa fie usor de citit si de inteles.

Diagramele UML vor fi folosite pentru a arata relatiile dintre clase si pachete.Diagramele UML folosite pentru clase ilustreaza clasele utilizate in aplicatie si relatiile dintre acestea,in timp ce in diagramele folosite pentru pachete ne prezinta pachetele in care se afla fiecare clasa.



Structurile de date folosite sunt alese astfel incat aplicatia sa ruleze cat mai optim si pentru a stoca si a gestiona datele pe care le utilizam in aplicatie.De exemplu,polinoamele noastre vor fi stocate intr-un Map,care ne usureaza modul de lucru la operatii.

4.Implementare

Pentru inceput am creat clasa “Polinom”,care initializeaza in constructor un HashMap si am implementat un setter si un getter prin care sa putem seta valorile polinomului si prelua valorile acestuia in alte clase.Aceasta clasa mai contine doua metode: o metoda denumita “adaugaPolinom”,in care stocam in polinomul nostru niste valori,si o metoda denumita “afisare”,care parcurge map-ul si afiseaza un polinom cu o structura adecvata.

Urmatoarea clasa importanta este clasa “Operatii”,in care in 6 metode diferite efectuam operatiile pe polinoame.Majoritatea operatiilor se bazeaza pe lucrul cu HashMap-uri,in care facem diferite parcurgeri pe map sau pe key,in care adunam,scadem,inmultim,impartim coeficientii si in care adunam si scadem puterile dupa caz,in functie de operatie.O alta metoda importanta din aceasta clasa este reprezentata de metoda “matching”,in care folosim un regex,extragem dintr-un String coeficientii si puterile unui polinom,permitandu-ne sa stocam aceste valori in polinomul nostru cu structura de HashMap.

Interfata utilizator,denumita “Calculator” este relativ simpla.Aceasta contine 2 textField-uri in care bagam inputul polinoamelor,6 butoane pentru fiecare operatie,un buton de clear si un textArea in care vom afisa rezultatul.Pentru fiecare buton in parte am adaugat un ActionListener,in care tratam diferitele exceptii pentru inputuri gresite,iar in cazul unor inputuri bune(specifice exemplului din partea de sus a interfetei) vom apela metodele din clasa “Operatii” si vom afisa in textArea-ul din josul interfetei rezultatul dorit.Aici am folosit un alt regex care verifica ca polinoamele introduse in textField sa aiba structura din exemplu.

5.Rezultate

In clasa in care am implementat testarile unitare,avem in total 12 testari,cate 2 pentru fiecare operatie,o testare corecta si una gresita. In metodele in care implementam testarile corecte,se vor initializa doua sau un singur polinom,in functie de operatie,vom stoca in alt polinom rezultatul de dupa executarea operatiei si,folosind metoda assertEquals,vom compara continutul rezultatului asteptat cu ceea ce ar trebui sa se afiseze,astfel incat testul sa fie corect.In metodele in care implementam testarile corecte,rezultatul asteptat va fi comparat intentionat cu un rezultat eronat,astfel incat testul sa fie considerat gresit.

6.Concluzii

In concluzie,aceasta tema m-a invatat sa aprofundez notiunile pe care le-am invatat la JAVA pana acum,m-a ajutat sa inteleg notiunile de regex,pattern si matcher,dar si notiunile legate de testarile unitare.Totodata,aceasta tema m-a invatat cum sa-mi structurez si sa-mi organizez codul,lucrând cu pachete si mai multe clase,ca sa pot avea un cod cat mai lizibil si usor de descifrat,fara metode cu un numar mare de linii de cod si clase foarte incarcate.

Pentru a imbunatati pe viitor acest proiect am putea face interfata sa arate mai estetic si am putea crea o functie care sa ne rezolve operatiile pe polinoame,pas cu pas,astfel incat utilizatorul sa poata invata din folosirea acestei aplicatii cum sa calculeze si pe hartie,nu sa obtina rezultatele direct,neintelegand ceea ce se intampla in spatele interfetei.

7.Bibliografie

1. <https://regexr.com/>
2. https://www.w3schools.com/java/java_regex.asp
3. <https://junit.org/junit5/docs/current/user-guide/#junit-platform-suite-engine>
4. <https://www.educba.com/java-parse-string/>

