

DOCUMENTATIE

Tema 1

Elev: Rad Vladut Cosmin George
Grupa: 30225
An: 2 CTI

CUPRINS:

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

1. Obiectivul temei

Obiectivul principal al temei a fost de a crea un calculator de polinoame. Am realizat aceasta tema cu ajutorul limbajului de programare Java. A fost necesara citirea polinoamelor si implementarea unor operatii: adunare, scadere, inmultire, impartire, derivare si integrare. Pentru a avea un design mai interesant si pentru a se observa mai usor am implementat si o interfata grafica.

Ca si obiective secundare:

- Am impartit codul in clase si in pachete
- Au fost respectate paradigmele Programarii Orientate pe Obiect
- Am lucrat cu TreeMap pentru o implementare mai usoara
- Testare Unitara pentru fiecare operatie a fost implementata cu JUnit

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

In cadrul interfetei grafice am reusit sa adaug niste butoane pentru fiecare operatie in parte. Pentru fiecare dintre aceste operatii ar trebui respectata introducerea corecta a polinoamelor ca in exemplul prezentat in interfata: $2x^2 + 7x^1 - 5x^0$.

In interiorul calculatorului de polinoame exista mai multe clase care provin din mai multe pachete diferite. Avem pachetul org.examples din care face parte clasa:

-Main

In pachetul polinom avem clasele :

-POLYNOM

-OPERATII

Pentru interfata am facut un pachet separat numit gui care contine clasa

-GUI

Pentru testare am facut o alta clasa in pachetul Java numita

-OPERATIITest

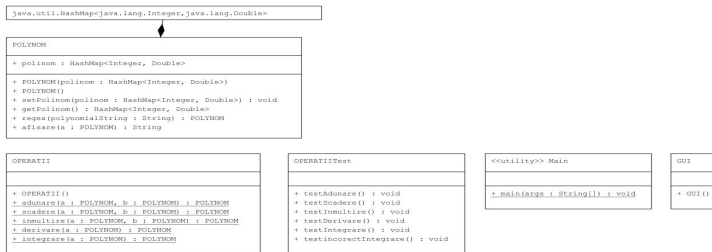
Dupa rularea clasei Main.c, se va deschide o fereastră si va aparea urmatoarea imagine in care pe urma utilizatorul va introduce polinoamele pe care se fac operatiile:

The image shows a Java Swing window with a light gray background. At the top, there are two labels: "Polinom 1" and "Polinom 2". Below each label is a white rectangular text input field. Underneath the input fields is a 2x2 grid of blue buttons with white text. The buttons are labeled: "Adunare" (Addition), "Scadere" (Subtraction), "Inmultire" (Multiplication), "Impartire" (Division), "Derivare" (Derivation), and "Integrare" (Integration). Below the grid is a label "Rezultat".

Polinom 1	Polinom 2
<input type="text"/>	<input type="text"/>
Adunare	Scadere
Inmultire	Impartire
Derivare	Integrare

Rezultat

3.Design- diagrama UML



4. Implementare

Clasa Main am utilizat-o pentru a verifica la fiecare pas daca operatiile pe care le am implementat sunt corecte si functioneaza pe oricare polinom le-as incerca. In ea am apelat si GUI care este interfata grafica implementata proiectului.

Clasa POLYNOM am folosit-o pentru a crea polinoamele. M-am folosit de metoda hashMap, astfel incat cheia(key) e reprezentata de coeficientul unui termen si value(valoarea) este puterea unui termen din polinom.

HashMap în Java este o colectie bazata pe harta si consta din perechi cheie-valoare. Un element HashMap poate fi accesat folosind o cheie, adica trebuie sa stim cheia pentru a accesa elementul HashMap.

Un HashMap foloseste o tehnică numita „Hashing”. In hash, un sir mai lung este convertit intr-un sir mai scurt prin aplicarea unor algoritmi sau „functie hash”. Un sir este convertit într-un sir mai scurt, deoarece ajuta la cautarea

care este mai rapida. Este, de asemenea, utilizat pentru indexarea eficienta.

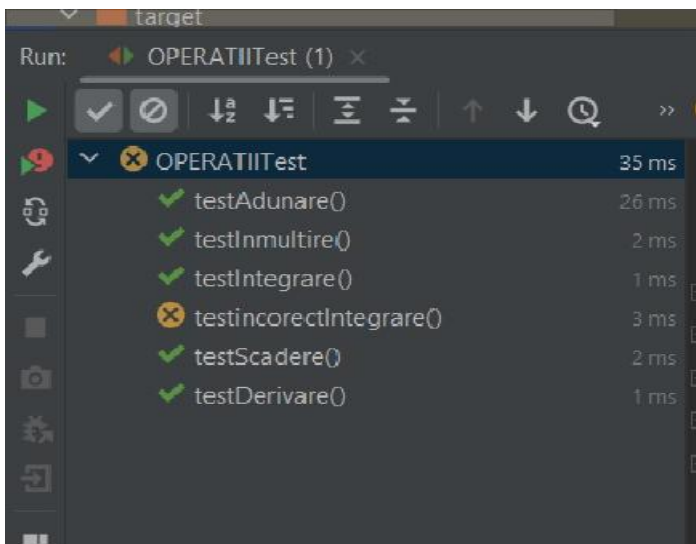
Clasa OPERATII a fost utilizata pentru a implementa operatiile necesare dorite, astfel am creat cate o functie pentru fiecare, asadar functioneaza urmatoarele operatii: *adunare, scadere, inmultire, impartire, derivare si integrare*. Fiecare din aceste operatii au fost implementate cu ajutorul hashingului, astfel nu am folosit alta clasa in plus. *Adunarea* a fost prima functie implementata unde am adunat cu usurinta coeficientii pentru elementele care au aceeasi putere. *Scaderea* este foarte asemanatoare adunarii, doar ca in loc sa adun coeficientii, i-am scazut, iar pentru elementele care se gasesc cu un coeficient in primul polinom si nu se gasesc in al doilea le-am adaugat la rezultat. *Inmultirea* am facut-o tot pe baza adunarii, astfel am inmultit separat fiecare element al unui polinom si am adunat pe urma coeeficientii care se repetau pentru aceeasi putere. *Derivarea* am facut-o foarte simplu, am implementat-o doar pentru primul polinom si am returnat coeficientul inmultit cu polinomul la puterea sa plus unu. Pentru *integrare* am returnat rezultatul inmultirii coeficientului cu impartirea polinomului la puterea sa plus unu cu puterea sa plus unu. Toate aceste operatii le am facut prin intermediul hashmap-ingului.

Clasa GUI este folosita pentru a reda o interfata grafica proiectului, astfel am folosit pachetul Java Swing cu atributiile lui pentru a putea face cu usurinta folosirea programului pe care am incercat sa-l implementez.

Pentru a putea sa inseram in text field-ul polinom un polinom fara sa ne dea eroare, am folosit regex ul pe care l am gasit sub path-ul acesta si care functioneaza pt ce avem noi nevoie, adica sa putem scrie polinomul ca si pe un script.

5. Rezultate

Pentru a ma asigura ca rezultatele sunt corecte, am mai implementat o clasa pe care am numit-o OPERATIITest care na ajuta la testarea operatiilor pe care le am implementat. Astfel, fiecare functie test primeste unu respectiv doi parametrii de tip polinom pe care ii definesc eu si le dau valori, si un al treilea care sa fie rezultatul efectuarii operatiei dorite. Verific daca dupa ce fac functia, rezultatul coincide cu polinomul ales ca rezultat si daca da testul este corect, daca nu este gresit si este afisat in consola. Am facut si teste care sa fie corecte si care sa fie gresite pentru a vedea cu usurinta si cu siguranta ca ceea ce se intampla este corect.



6. Concluzii

Aceasta implementare poate fi imbunatatita prin adaugarea anumito mesaje si butoan in plus la interfata graica pentru a o face mai usor de folosit, implementarea functiei impartire si multe altele care ar face-o mai usor de utilizat.

7. Bibliografie

- creerea unui GUI <https://www.youtube.com/watch?v=iE8tZ0hn2Ws>
https://www.youtube.com/watch?v=5o3fMLPY7qY&t=653s&ab_channel=AlexLee
- informatii despre map si hashmap <https://www.youtube.com/watch?v=H62Jfv1DJlU>
- folosirea regex-ului <https://regexr.com/>

