



Facultatea de Automatică și Calculatoare

Departamentul de Calculatoare

Documentație Tema 1

Calculator polinomial

Student:

Mălăescu Bianca

Grupa 30238



Cuprins

1. Obiectivul temei	3
2. Analiza problemei, modelare, scenarii, cazuri de utilizare	3
3. Proiectare	5
4. Implementare	8
5. Rezultate	8
6. Concluzii.....	9
7.Bibliografie.....	9



1. Obiectivul temei

Obiectivul temei constă în proiectarea și implementarea unui calculator polinomial cu o interfață grafică dedicată. Cu ajutorul acesteia, utilizatorul poate introduce polinoame, poate selecta operația matematică aplicată polinoamelor introduse (adunare, scădere, înmulțire, împărțire, derivare, integrare) și poate vizualiza rezultatul operației alese.

Se iau în considerare polinoamele unei singure variabile “x”, iar coeficienții polinomului sunt numere întregi.

Limbajul utilizat pentru realizarea aplicației este Java, unul dintre cele mai populare limbaje de programare care utilizează programarea orientată pe obiecte. Aceasta este folosit cel mai des pentru aplicații desktop, web sau de mobil.

Scopul acestui proiect este de a implementa funcțiile de bază pe polinoame, pentru a reliefa modul în care acestea pot fi create și manipulate prin intermediul paradigmei programării orientate pe obiecte. De asemenea, polinoamele și operațiile cu acestea se numără printre cele mai folosite structuri matematice în Computer Science.

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

Analiza problemei

În matematică, un polinom este o expresie construită dintr-una sau mai multe variabile și constante, folosind doar operații de adunare, scădere, înmulțire și ridicare la putere constantă pozitivă întreagă. $x^2 - 4x + 7$ este un polinom. Se observă în particular că împărțirea printr-o expresie ce conține o variabilă nu este permisă în polinoame.

Polinoamele sunt construite din termeni numiți monoame, care sunt alcătuite dintr-o constantă (numită coeficient) înmulțită cu una sau mai multe variabile. Fiecare variabilă poate avea un exponent constant întreg pozitiv. Exponentul unei variabile dintr-un monom este egal cu gradul acelei variabile în acel monom. Gradul unei variabile fără exponent este unu. Un monom fără variabile se numește monom constant, sau doar constantă.

De exemplu, următoarea expresie este un polinom:

$$3x^2 - 5x + 4$$

El constă din trei monoame: primul are gradul doi, al doilea are gradul unu, iar al treilea are gradul zero.

Când un polinom este dispus în ordinea naturală, el are termenii de grad mai mare înaintea celor de grad mai mic. În primul termen, coeficientul este 3, variabila este x, iar exponentul este doi. În al doilea termen, coeficientul este -5. Al treilea termen este o constantă. Gradul unui polinom este cel mai mare grad al unui termen al său. În acest exemplu, polinomul are gradul doi.

Operațiile care pot fi aplicate pe polinoame sunt: adunarea, scăderea, înmulțirea, împărțirea, derivarea și integrarea. Deoarece scăderea poate fi tratată ca o adunare cu termenul opus, și deoarece ridicarea la o putere întreagă pozitivă și constantă poate fi tratată ca înmulțire repetată, polinoamele se pot construi din constante și variabile folosind doar două operații: adunarea și înmulțirea.

Adunarea a 2 polinoame presupune adunarea monoamelor cu același exponent, corespunzătoare celor 2 polinoame. Asemenea se realizează și scăderea. Se scad monoamele cu același exponent corespunzătoare celor 2 polinoame.



Înmulțirea polinoamelor presupune adunarea produselor rezultate în urma adunării fiecărui monom corespunzător primului polinom cu fiecare monom corespunzător celui de-al doilea. Este important să nu se uite reducerea termenilor asemenea pentru ca rezultatul să fie unul concis.

Modelare

Pentru a modela problema propusă avem nevoie de o reprezentare a polinoamelor propuse, de o implementare a operațiilor necesare și de o metodă de furnizare a rezultatelor așteptate. Ținem cont de faptul că cele 2 polinoame conțin o singură variabilă "x", iar coeficienții și exponenții polinomului sunt numere întregi.

Polinoamele vor fi compuse din mai multe monoame, fiecare monom având un coeficient și un exponent.

Partea cea mai vastă a modelării presupune realizarea operațiilor între polinoame. Pentru a aduna două polinoame este necesară adunarea coeficienților monoamelor care au același exponent. Scăderea este similară, trebuie scăzuți coeficienții monoamelor care au același exponent. Înmulțirea polinoamelor presupune înmulțirea fiecărui monom din primul polinom cu fiecare monom din al doilea polinom, iar apoi se reduc termenii. Derivarea polinomului utilizează funcția de derivare a monomului, unde coeficientul este rezultatul înmulțirii dintre coeficientul inițial și exponent, iar exponentul scade cu 1. Pentru a integra un polinom folosim o funcție de integrare a monoamelor, aplicând o împărțire între coeficientul inițial și exponentul mărit cu 1 și o adunare cu 1 la exponent. Operațiile de derivare și integrare se aplică pe polinomul 1, corespunzător primului textbox din interfață.

Scenariu

Scenariul aplicației cuprinde partea de modelare a polinoamelor, implementarea operațiilor și interacțiunea utilizatorului cu aplicația prin interfața grafică. Prin intermediul acesteia, utilizator poate să introducă de la tastatură cele două polinoame, să aleagă operația dorită și să vizualizeze rezultatul în caseta text dedicată.

La introducerea textului corespunzător polinoamelor, utilizatorul trebuie să țină cont de formatul acceptat. Variabila polinoamelor trebuie să fie mereu x/X , precedată de un coeficient întreg (cu semn sau fără semn în cazul coeficienților pozitivi) și urmată de caracterul "^" și exponentul corespunzător. În cazul nerespectării acestui format, în consolă se va afișa un mesaj corespunzător.

Interfața reprezintă un mediu facil de utilizare și este foarte intuitivă. Conține cele 2 textbox-uri unde utilizatorul va completa polinoamele, 6 butoane corespunzătoare operațiilor care se pot efectua (Adunare, Scădere, Înmulțire, Împărțire, Derivare, Integrare) și textbox-ul corespunzător rezultatului care va fi afișat.

Pentru utilizarea cu succes a aplicației este necesar ca utilizatorul să respecte următorii pași:

- Introducerea corectă a polinoamelor, ținând cont de formatul acceptat explicat mai sus (În cazul neintroducerii unui format acceptat se afișează un mesaj în consolă și utilizatorul poate reîncerca introducerea polinoamelor)

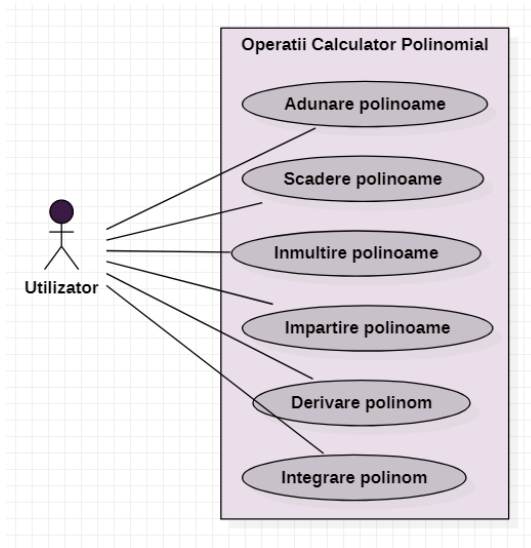
- Apăsarea butonului corespunzător operației dorite

- Vizualizarea rezultatului în caseta corespunzătoare

Resetarea valorilor se poate face prin editarea câmpurilor de text corespunzătoare polinoamelor și apăsarea oricărui buton. Fiecare apăsare de buton produce afișarea rezultatului corespunzător în caseta lui.



Diagrama cazurilor de utilizare



Cazurile de utilizare reprezintă cele 6 operații care pot fi aplicate asupra polinoamelor introduse. (Adunarea a 2 polinoame, scăderea a 2 polinoame, înmulțirea a 2 polinoame, derivarea unui polinom și integrarea unui polinom).

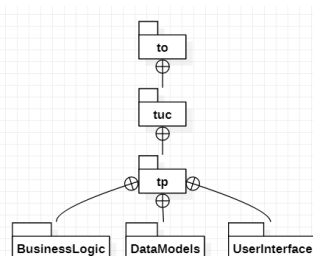
3.Proiectare

Arhitectura proiectului este bazată pe șablonul arhitectural Model – View – Controller.

MVC este un model de arhitectură utilizat în ingineria software. Succesul modelului se datorează izolării părții logice (business) de consideratele interfețe cu utilizatorul, rezultând o aplicație unde aspectul vizual sau/și nivelele inferioare ale regulilor de business sunt mai ușor de modificat, fără a afecta alte nivele. MVC este un model de arhitectură software care separă reprezentarea informațiilor din interacțiunea cu utilizatorul cu informațiile în sine.

Modul de funcționare: În primul rând, browser-ul trimite o solicitare către Controller. Apoi, Controller-ul interacționează cu Modelul pentru a trimite și a primi date. Controller-ul interacționează apoi cu View pentru a reda datele. Vizualizarea este preocupată doar de modul în care să prezinte informațiile și nu de prezentarea finală. În cele din urmă, View-ul va trimite prezentarea sa finală către Controller, iar Controller-ul va trimite acele date finale către rezultatul utilizatorului. Important este că View și Modelul nu interacționează niciodată între ele. Singura interacțiune care are loc între ele este prin intermediul Controller-ului.

Aplicația este organizată în mai multe pachete, care se pot observa în diagrama de pachete de mai jos.



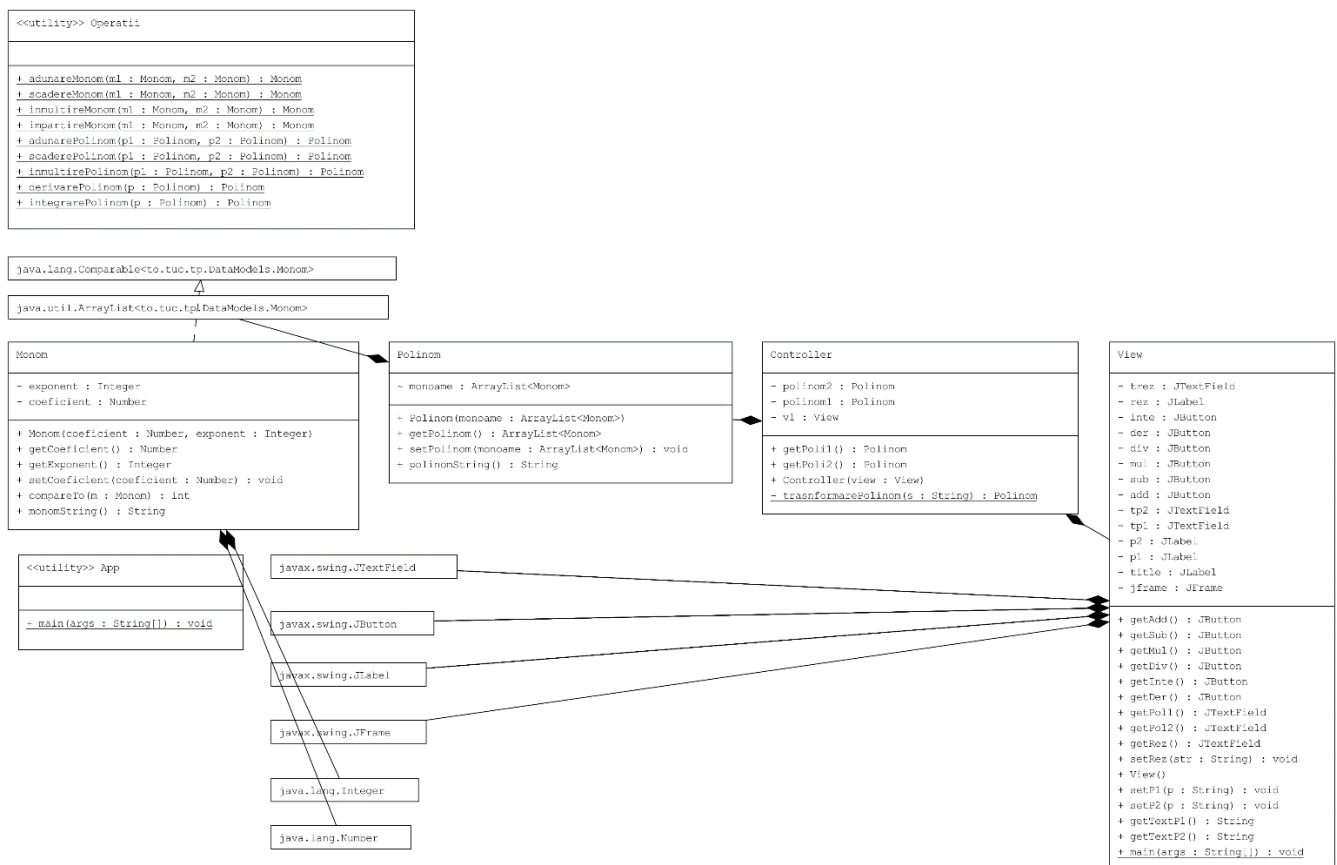


BusinessLogic conține partea de logică pe care o fac independent de alte pachete. În acest pachet se regăsește un singur fișier java care conține operațiile. În partea superioară a acestuia se află metodele care implementează operațiile pentru monoame, urmate de operațiile pentru polinoame. În operațiile corespunzătoare polinoamelor apelăm metode ale monoamelor.

DataModels conține cele 2 modele necesare implementării proiectului, monom și polinom. Așa cum am precizat în partea de modelare, polinomul îl vom trata ca o listă de monoame pentru a facilita implementarea și logica aplicației. Fiecare model are câteva metode caracteristice.

UserInterface reprezintă partea de interacțiune cu utilizatorul. În acest pachet regăsim fișierele corespunzătoare pentru View și Controller. În view este construită interfața grafică, iar partea de controller gestionează interacțiunea dintre View și Models. Aceasta preia input-ul și îl convertește în comenzi pentru view sau pentru modele. Totodată, în controller am implementat și una dintre cele mai importante funcții pentru aplicație, cea care preia string-ul introdus de utilizator pentru un polinom și îl convertește într-un obiect de tip Polinom pentru a putea realiza operațiile.

Diagrama UML de clase





În schema de mai sus este reprezentată diagrama UML de clase. Clasele cuprinse în proiect sunt:

-Clasa Monom: modelează conceptul de monom (care va compune polinomul) și are ca atribute coeficientul monomului și exponentul acestuia. Exponentul este un număr întreg, iar coeficientul l-am declarat ca Number. Deși polinoamele introduse de utilizator vor avea coeficienți întregi, în urma unor operații precum integrarea de exemplu, coeficientul monoamelor poate deveni de tip real (double), de aceea am ales tipul Number, ca să pot modela ambele situații.

-Clasa Polinom: modelează structura matematică a unui polinom, având ca atribute doar o listă de monoame. Am ales să lucrez cu ArrayLists pentru a facilita implementarea și logica conceptului de polinom, iar un avantaj al acestora este că au lungime variabilă.

-Clasa Operații: nu are atribute și conține doar metode pentru efectuarea operațiilor atât pe monoame, cât și pe polinoame. Metodele care efectuează operații pe monoame sunt folosite în cadrul metodelor pentru operații cu polinoame.

-Clasa View: cuprinde partea de interfață grafică. Am formatat fiecare label, buton și textbox manual pentru o imagine de ansamblu plăcută și pentru ca interfața să fie intuitivă.

-Clasa Controller: gestionează resursele din View și DataModels și face legătura dintre acestea. Are ca atribute un obiect de tip view și cele 2 polinoame care vor fi introduse de utilizator.

-Clasa App: a fost construită doar pentru rularea aplicației și nu conține decât o metodă.

Interfața grafică

Calculator polinomial

Polinomul 1:

Polinomul 2:

Adunare	Scadere
Inmultire	Impartire
Derivare p1	Integrare p1

Rezultat:



Prin intermediul interfeței grafice construite, utilizatorul va transmite polinoamele dorite și va putea vizualiza rezultatul operației alese. Interfața este intuitivă și ușor de folosit.

4. Implementare

Aplicația “Calculator Polinomial” a fost implementată în limbajul de programare Java, utilizând șablonul arhitectural Model-View-controller. Am folosit ca framework Swing pentru a implementa partea de interfață grafică. Swing face parte din clasa de bază Java și este folosit pentru a crea aplicații desktop. Include componente ca butoane, textfield-uri, etichete etc. Compunerea acestor componente rezultă o GUI (graphical user interface).

Metodele claselor

-Clasa Monom: are ca metode getters și setters, o metodă care descrie compararea a 2 monoame (în funcție de exponent), folosită pentru sortarea monoamelor într-un polinom și o metodă de transformare a unui monom într-un String, pentru a putea fi vizualizat un monom mai ușor.

-Clasa Polinom: pe lângă getters și setters se mai regăsește o metodă de convertire a unui polinom într-un String. Aceasta se folosește de metoda similară din clasa Monom.

-Clasa Operații: cuprinde cele mai importante metode ale aplicației, operațiile. Primele metode fac referire la operațiile cu monoame, iar cele de mai jos sunt operații cu polinoame. Metodele operațiilor cu monoame se regăsesc în metodele operațiilor pentru polinoame, pentru un calcul mai ușor.

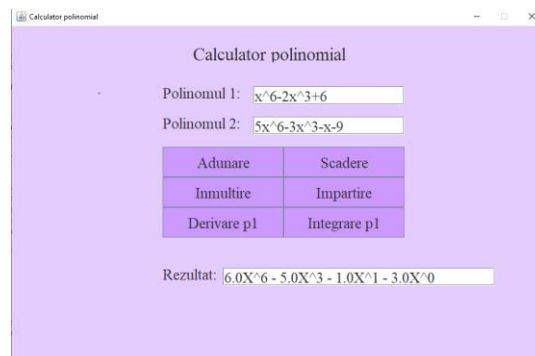
-Clasa View: conține doar getters și setters pentru a face legătura dintre interfața grafică și datele proiectului.

-Clasa Controller: gestionează evenimentele ActionListener pentru fiecare buton al interfeței și mai cuprinde o metodă foarte importantă, cea de transformare a unui string dat de utilizator în obiect de tip Polinom. Pentru realizarea acesteia am lucrat cu pattern matchings și biblioteca Regex. Expresia regulată folosită împarte polinomul dat ca string în 4 tipuri de grupuri de monoame, care sunt identificate în cod pentru a stabili coeficientul și exponentul corect. De asemenea, am construit și un validator pentru datele introduse de utilizator. Astfel, dacă formatul String-ului dat nu este cel dorit, nu se efectuează operații iar utilizatorul va fi notificat printr-un mesaj în consolă.

-Clasa App: conține doar o metodă main care punește aplicația. Sunt instanțiate un obiect de tip Controller și un View.

5. Rezultate

Aplicația funcționează bine, singura excepție fiind operația de înmulțire care nu a fost implementată. A fost testată doar manual, prin multe exemple care prezintă excepții sau rezultate deosebite.





6. Concluzii

Realizarea proiectului m-a familiarizat cu șablonul arhitectural Model-View-Controller și cu framework-ul Swing, foarte util pentru realizarea interfeței. Am lucrat pentru prima dată cu pattern matching folosind biblioteca Regex și am reușit să implementez cea mai mare parte a proiectului, exceptând operația de înmulțire care mi s-a părut mai complicată de modelat.

7. Bibliografie

- [1]: <https://www.javatpoint.com/java-swing>
- [2]: <https://stackoverflow.com/questions/36490757/regex-for-polynomial-expression>
- [3]: <https://regex101.com/>
- [4]: <https://en.wikipedia.org/wiki/Polynomial>
- [5]: https://users.utcluj.ro/~igiosan/teaching_poo.html
- [6]: <https://dsrl.eu/mantal/pt/>