

FACULTATEA: Automatică și calculatoare

SPECIALIZAREA: Calculatoare și Tehnologia informației

DISCIPLINA: Tehnici de programare

TEMĂ: Calculator de polinoame

**Realizator:** Bologa Horațiu

2 0 22

**CUPRINS**

1. Obiectivul temei . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
2. Analiza problemei, modelare, scenarii, cazuri de utilizare . . . . . . . . . . .
3. Proiectare . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

* decizii de proiectare . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* diagrame UML . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* structuri de date . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* proiectare clase . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* interfețe . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* relații . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* packages . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* algoritmi . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
* interfața utilizator . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

1. Implementare și testare . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
2. Rezultate . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
3. Concluzii . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
4. Bibliografie . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
5. **Obiectivul temei**

*Cerință:* Proiectați și implementați un sistem de procesare a polinoamelor de o singură variabilă cu coeficienți întregi.

Obiectivul primei teme de laborator este de a realiza un calculator de polinoame cu un grad de dificultate scăzut de utilizare, care are, de asemenea, și o interfață lizibilă, inteligibilă pentru utilizator.

Calculatorul de polinoame are la bază câteva operații fundamentale, precum:

* Adunarea
* Scăderea
* Înmulțirea
* Împărțirea
* Derivare (pentru fiecare polinom în parte)
* Integrare (pentru fiecare polinom în parte)

Un obiectiv secundar ar putea fi considerat parsarea polinoamelor, care sunt trimise ca și string-uri in căsuțele text asignate acestor operațiuni.

1. **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, spre exemplu x^2-4x+7 este un polinom.

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. Gradul unui termen constant este 0. Coeficientul unui monom poate fi orice număr, inclusiv fracții sau numere negative.

Pentru a utiliza calculatorul polinoamele trebuie introduse în căsuțele predestinate si sa se apese butonul memorare, pentru validarea datelor (acestea sa fie convertite in tip polinom din string).Cum polinoamele introduse se citesc de fapt ca niște stringuri nu numere, este necesară o parsare a datelor (spargem în monoame, care conțin coeficientul, număr real și puterea sau coeficientu). La parsare, am utilizat metoda split (după ”+” și ”-” pentru a separa în monoame și după ”x^” pentru a obține coeficientul și puterea).

Dupa introducerea si memorarea celor două polinoame se alege operația dorită si rezultatul în urma operației selectate va fi afișat în căsuța de tip text predestinată rezultatului.

Așadar, descrierea use-case pe pași este următoarea:

* Utilizatorul introduce polinoamele (fără spații, iar între coeficient și ”x”nu se folosește ”\*”; pentru ”x^1” se scrie doar ”x”, iar pentru constante nu e nevoie să se scrie și ”x^0”)ș
* Se apasă butonul ”OK” pentru validare
* Se alege operația / operațiile dorite
* Sistemul realizează operația aleasă
* Este afișat rezultatul pe ecran
* Se revine la primul pas dacă dorim să schimbăm polinoamele, iar dacă nu, se alege o altă operație

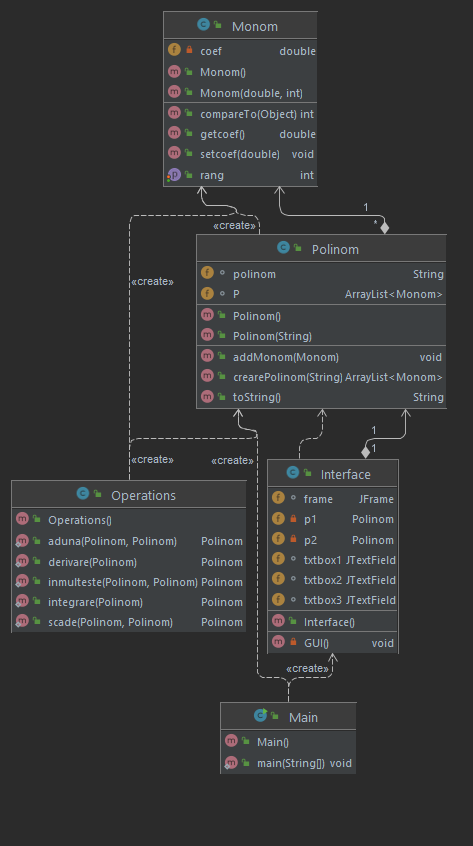
1. **1. Decizii de proiectare**

Pentru a putea manipula stringurile (care sunt polinomane) a fost necesară o parsare cu ajutorul metodei split, iar pentru stocarea monoamenlor rezultate pentru fiecare polinom a fost aleasa o structură de date de tip ArrayList.

1. **2. Diagrama UML, relații**

Unified Modeling Language (UML) este un limbaj standard pentru descrierea de modele și specificații pentru software. Diagrama de clase este un tip de diagramă ce este utilizată pentru a descrie structuri statice, adică a entităţilor sau claselor existente într-un sistem (adăugându-se și ce conțin acestea).

Diagrama UML pentru calculatorul de polinoame construit:



Între polinom și monom există o relație de agregare de tip ”este parte a”, monomul fiind parte a polinomului (un polinom nu ar putea exista fără monoame), iar un polinom are mai multe monoame (relație de tip “many-to-one”).

În rest, după cum se vede, este foarte folosită relația de dependență de tip ”folosește”.

1. **3. Clase, interfață împreună cu algoritmi**

În clasa **Interface** avem 9 butoane (8 pentru operații și unul pentru validarea polinoamelor) și unul pentru memorarea polinoamelor. Operațiile care sunt definite de cele 8 butoane sunt : ”ADUNARE”, ” SCADERE {p1 - p2}”, ”ÎNMULȚIRE”, ”IMPARTIRE {1 / 2}”, ”DERIVARE {p 1}”, ”INTEGRARE {p 1}”, ”DERIVARE {p 2}”, ”INTEGRARE (pol 2)”.

În interfață se mai găsesc si încă 3 zone destinate textului: 2 pentru a putea introduce, pe rând, fiecare polinom și unul pentru afișarea rezultatului, care nu poate sa fie editat. De asemenea, avem și labeluri, care au rolul de a oferi o mai buna organizare a ferestrei si sa ofere o interfață prietenoasa utilizatorului.

Frame-ul conține un panel, care, la rândul lui, conține paneluri mai mici:

* Primul panel, care conține labelurile pentru polinoame, precum și casetele de text unde se introduc polinoamele și butonul ” MEMORARE” pentru a înregistra polinoamele introduse
* Al doilea panel conține labelul cu titlul ” Alegeti operatia dorita”
* Al treilea panel conține butoanele corespunzătoare operațiilor implementate
* Al patrulea panel conține labelul pentru rezultat, precum și caseta de text în care este afișat acesta.

În clasa **Monom**, definim coeficientul (double) și rangul / puterea (int), care sunt caracteristice fiecărui monom. Metodele implementate sunt *get-erele* și *set-erele* pentru coeficient și rang, dar și o metodă *compareTo*, care ne va ajuta in stabilirea unei ordini descrescătoare pentru fiecare monom din polinoame.

În clasa **Polinom,** este definit stringul Polinom (acesta care este de tip string si este trimis de la tastatură) și o listă de monoame care este specifica polinomului. Metoda cea mai importantă, pe care se bazează proiectul este *parsarea* stringului preluat de la tastatură. Acesta se bazează pe RegEx (expresie regulată), cu ajutorul căruia facem împărțirea în monoame, iar ulterior, pentru fiecare monom, găsim coeficientul și puterea. Tot în clasa Polinom, este implementată și o metodă care are ca și scop adăugarea unui monom în lista de monoame a polinomului, *addMonom*. Metoda *toString* sortează înainte de a efectua operațiile după rangurile pe care astea le au. Aceasta operație este împărțita in 3 cazuri (cazul in care coeficientul monomului este mai mic decât 0, cazul in care acesta este mai mare decât 0 și cazul in care este 0, acesta este folosit la integrare unde apar si numere care prin integrare devin 0)

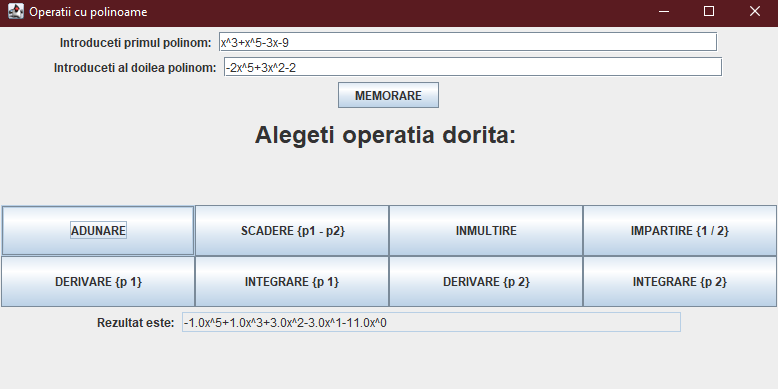
Clasa **Operations** este alcătuita din operațiile necesare calculatorului nostru. Operația de *adunare* are ca si scop parcurgerea celor doua polinoame si verificarea egalității coeficienților elementelor cu același rang, astfel fiind adăugat in rezultat. După aceasta parcurgere inițiala mai este necesară o parcurgere pentru monoamele care nu sunt regăsite în ambele polinoame. *Scăderea* este implementată pe același principiu, dar diferența se face la un semn (din + pentru a aduna termenii, se face – pentru a scădea) și la parcurgerea monoamelor care nu au corespondent în primul din polinomul de după – , acestea se trec cu semn schimbat. Pentru *înmulțire,* operația a fost implementata prin parcurgerea polinoamelor si înmulțirea fiecărui monom cu fiecare. La final a mai fost parcurs o data pentru a aduna monoamele cu aceeași putere. La operația de *derivare,* trebuie păstrată regula de parcurgere a polinoamelor și mers pe doua posibilități, dacă rangul este mai mare ca 1, atunci se înmulțește rangul cu coeficientul și se decrementează rangul și daca rangul este 0, atunci coeficientul devine 0. Pentru *integrare* algoritmul este asemănător cu cel a derivării (se incrementează rangul și după aceea se împarte cu coeficientul ), diferența fiind ca nu mai avem nevoie de cazul cu rangul =0.

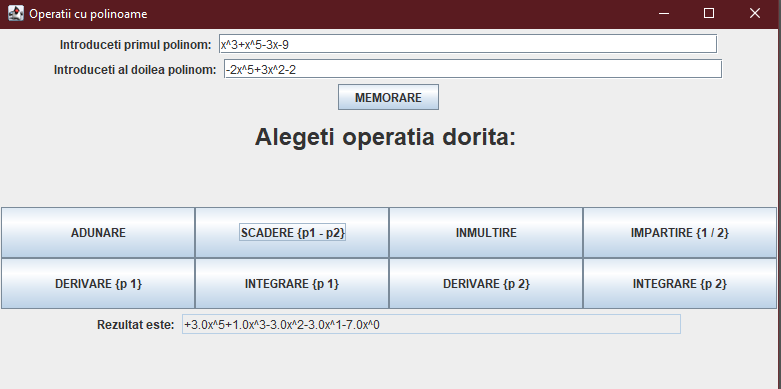
**4. Pachete**

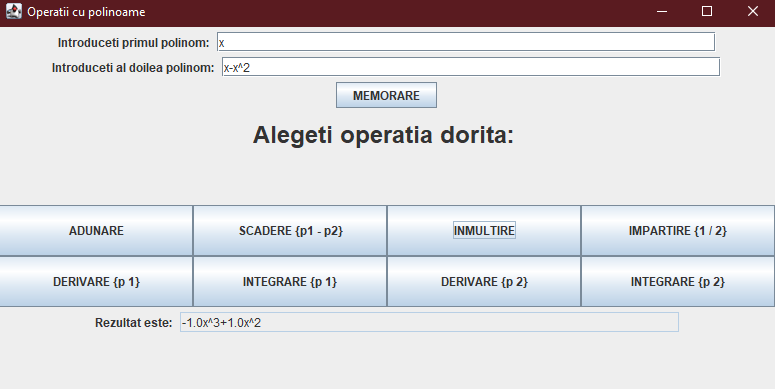
Toate clasele sunt acoperite de același pachet, acesta fiind importat in toate clasele proiectului : **package PT.ass1.test2.model.model;** . Pe langa acesta, clasele Monom, Polinom și Operations fac parte din pachetul model, Main din controller și Interface din view.

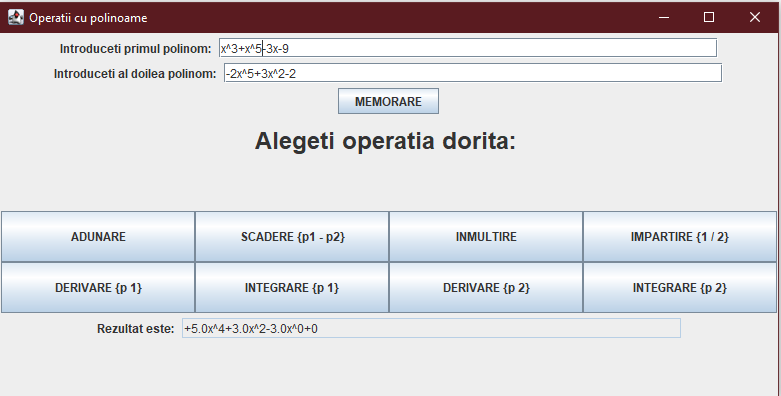
1. **Testare și rezultate**

În print screenurile ce urmează, a fost testată fiecare operație pe polinoame in ordinea in care sunt afișate butoanele pe ecran.

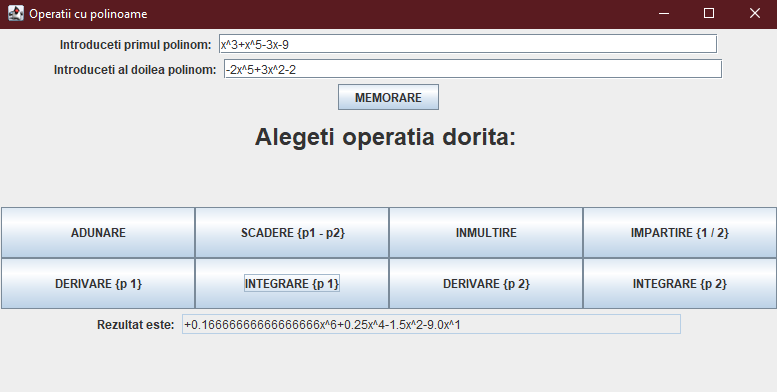
ADUNARE:

SCĂDERE:

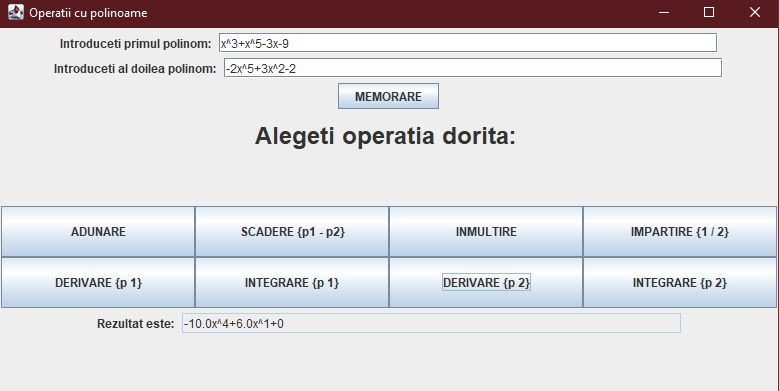
ÎNMULȚIRE:

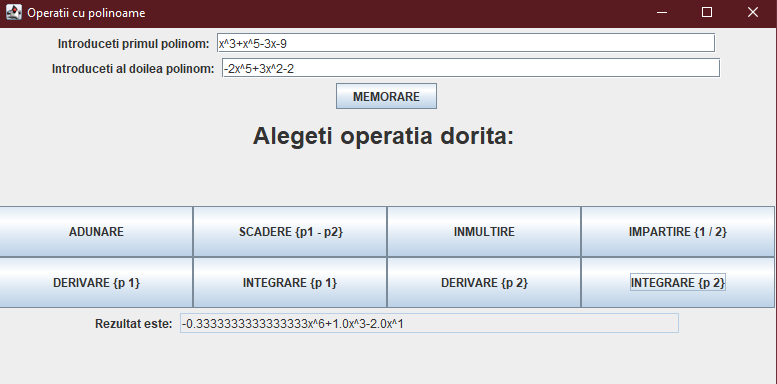
DERIVARE (POLINOM 1):

INTEGRARE (POLINOM 1):



DERIVARE (POLINOM 2):



 INTEGRARE (POLINOM 2):

1. **Concluzii și posibilități de dezvoltare**

Posibilitățile de dezvoltare sunt foarte multe atât pe partea de interfața (adăugându-se noi butoane si noi funcționalități, posibil și o interfața de tip drag and drop) si pe partea de implementare (adăugarea de operații noi, optimizarea celor existente)

Acest calculator a avut scopul de a fi ușor de utilizat si manevrat de către utilizator, acest netrebuind sa știe implementarea din spate pentru a-l putea utiliza și sa aibă operațiile de baza pentru polinoame.

1. **Bibliografie**

<https://stackoverflow.com/questions/3481828/how-to-split-a-string-in-java>

<https://ro.wikipedia.org/wiki/Polinom>

<https://regexr.com/>

<https://docs.oracle.com/javase/7/docs/api/>