

# DOCUMENTATIE

## TEMA 1

NUME STUDENT: Igna Alexandra Andreea

GRUPA: 30225

# CUPRINS

1.	Obiectivul temei.....	3
2.	Analiza problemei, modelare, scenarii, cazuri de utilizare.....	3
3.	Proiectare.....	5
4.	Implementare.....	6
5.	Rezultate.....	8
6.	Concluzii.....	10
7.	Bibliografie.....	10

## 1. Obiectivul temei

Obiectivul principal al temei este crearea unei aplicatii care sa permita efectuarea operatiilor de adunare, scadere si inmultire a doua polinoame.

Obiectivele secundare sunt:

- Implementarea unei clase numita *Operator*, care are operatiile de adunare, scadere si inmultire care vor fi scrise in metodele **addPolinom**, **subtractPolinom** si **multiplyPolinom**.
- Implementarea unei clase *Polinom* care sa reprezinte un polinom si sa aiba metodele **addPolynomials**, **subtractPolynomials**, **multiplyPolynomials** (de la clasa *Operator*), **toString**, **toPolynomial**.
- Implementarea unei clase *PolynomialCalculatorGUI* care sa reprezinte interfata grafica a aplicatiei, si sa aiba metoda **actionPerformed** care sa fie apelata atunci cand utilizatorul apasa butonul de calcul.

Aceste metode vor fi explicate mai in detaliu in Capitolul 4 din documentatie.

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

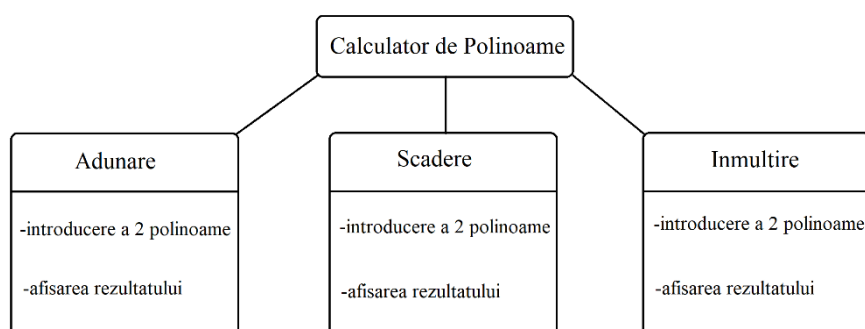
Cerinte functionale:

- Sa permita utilizatorului sa introduca polinoamele
- Sa permita utilizatorului sa efectueze operatii aritmetice cu polinoame: adunare, scadere, inmultire
- Sa permita utilizatorului sa afiseze rezultatul operatiilor efectuate

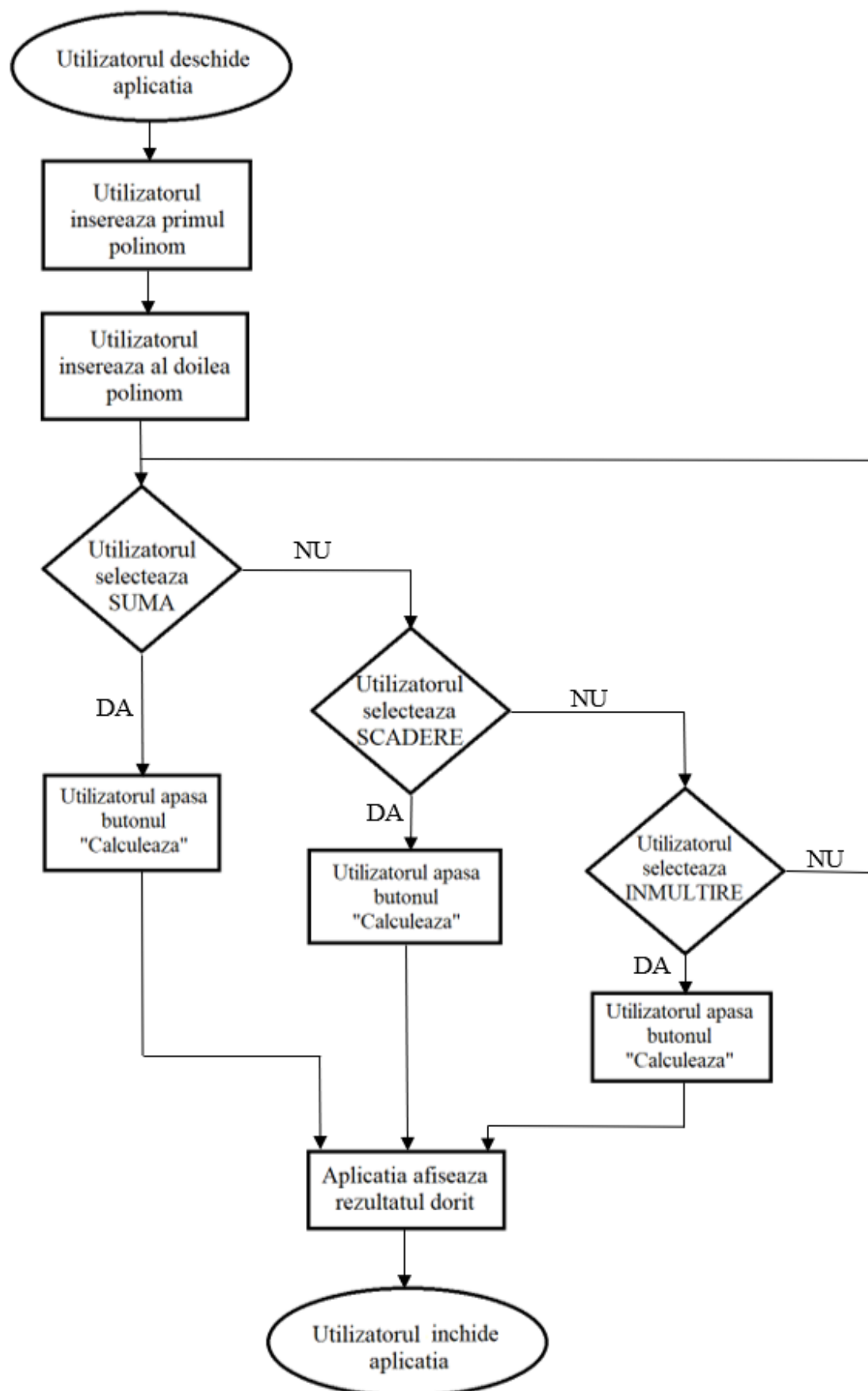
Cerinte non-functionale:

- Interfata grafica intuitiva si usor de utilizat
- Timp de raspuns rapid la operatiile efectuate
- Precizie in afisarea rezultatelor, cu un numar minim de erori

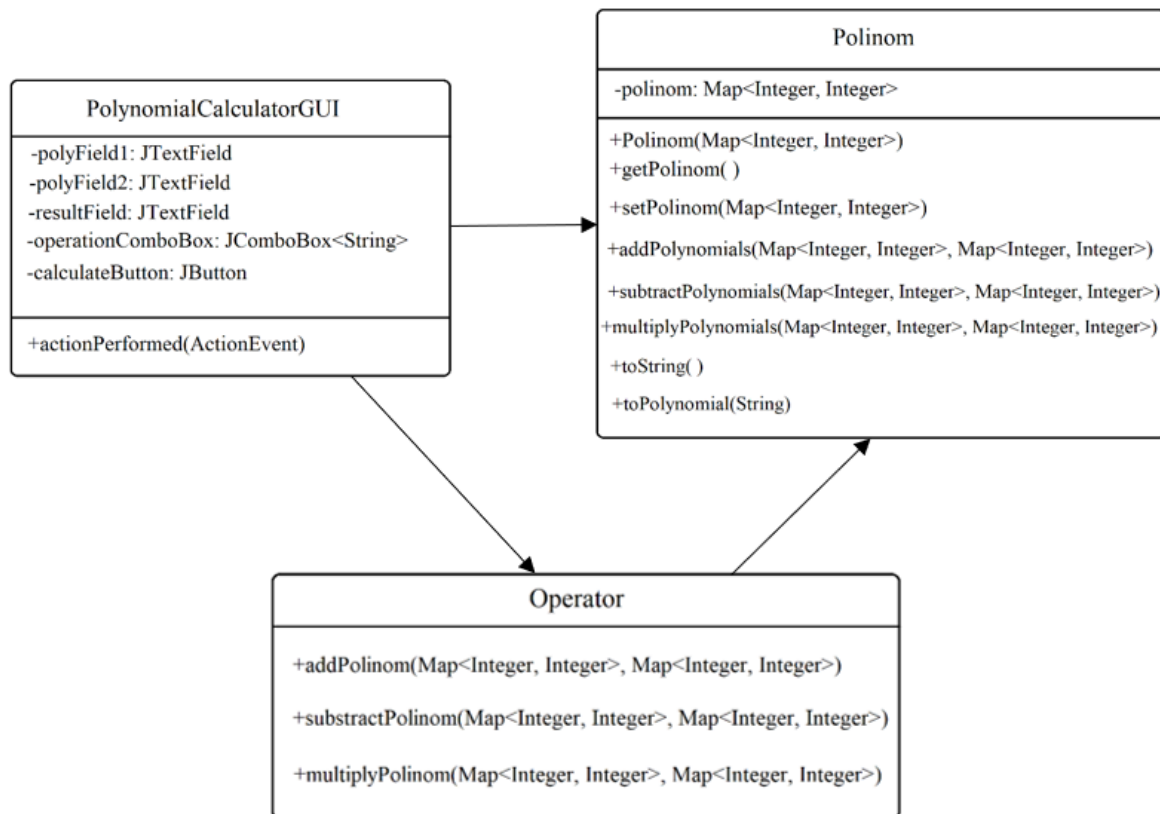
Diagrama cazurilor de utilizare pentru un calculator de polinoame:



Flow chart cu Descrierea use-case-urilor:



### 3. Proiectare



Relatia dintre clasa **Polinom** si clasa **PolynomialCalculatorGUI** este de asociere. Clasa **PolynomialCalculatorGUI** foloseste obiecte de tip **Polinom**, dar nu le detine in mod direct. In plus, modificarile facute in clasa **Polinom** nu afecteaza clasa **PolynomialCalculatorGUI** in mod direct, deci nu exista o relatie de dependenta sau de mostenire.

Structuri de date folosite: **Map<Integer, Integer>**, pentru a reprezenta coeficientii unui polinom.

Algoritmi folositi sunt:

- de citire a unui polinom din input.
- de adunare a doua polinoame.
- de scadere a doua polinoame.
- de inmultire a doua polinoame.

## 4. Implementare

### Clasa Polinom

- **addPolynomials(Map<Integer, Integer> p1, Map<Integer, Integer> p2)**  
metoda statica care primeste doua map-uri de tipul (exponent, coeficient) si returneaza un nou map care reprezinta suma polinoamelor asociate map-urilor primite ca parametru.
- **subtractPolynomials(Map<Integer, Integer> p1, Map<Integer, Integer> p2)**  
metoda statica care primeste doua map-uri de tipul (exponent, coeficient) si returneaza un nou map care reprezinta diferenta polinoamelor asociate map-urilor primite ca parametru.
- **multiplyPolynomials(Map<Integer, Integer> p1, Map<Integer, Integer> p2)**  
metoda statica care primeste doua map-uri de tipul (exponent, coeficient) si returneaza un nou map care reprezinta produsul polinoamelor asociate map-urilor primite ca parametru.
- **toString()**  
override la metoda toString care returneaza o reprezentare sub forma de string a polinomului, cu coeficientii si exponentele ordonate descrescator dupa gradul termenilor.
- **toPolynomial(String input)**  
metoda statica care primeste un string care reprezinta un polinom sub forma " $ax^n + bx^{(n-1)} + \dots + c$ " si returneaza un nou polinom corespunzator acelei reprezentari, sub forma unui map cu perechile (exponent, coeficient).

### Clasa Operator

- **Metoda addPolinom(Map<Integer, Integer> p1, Map<Integer, Integer> p2)**

Mai intai, se creeaza o mapa goala numita "result". Se parcurge map-ul "p1" (reprezentand primul polinom) si pentru fiecare exponent (cheie) se verifica daca acesta exista si in map-ul "p2" (reprezentand al doilea polinom). Daca exista, se aduna valorile corespunzatoare din p1 si p2 si se adauga in mapa "result". Daca nu exista, se adauga doar valoarea corespunzatoare din p1 in map-ul "result". Apoi, se parcurge map-ul "p2" si se verifica daca exista un exponent care nu exista in map-ul "p1". Daca exista, se adauga valoarea corespunzatoare in map-ul "result". La sfarsit, metoda returneaza mapa "result" care contine suma polinoamelor.

- **Metoda `subtractPolinom(Map<Integer, Integer> p1, Map<Integer, Integer> p2)`**

Mai întâi, se creează o mapă goală numită "result". Se parcurge map-ul "p1" (reprezentând primul polinom) și pentru fiecare exponent (cheie) se verifică dacă acesta există și în map-ul "p2" (reprezentând al doilea polinom). Dacă există, se scad valorile corespunzătoare din p1 și p2 și se adaugă în mapă "result". Dacă nu există, se adaugă doar valoarea corespunzătoare din p1 în map-ul "result". Apoi, se parcurge map-ul "p2" și se verifică dacă există un exponent care nu există în map-ul "p1". Dacă există, se adaugă valoarea corespunzătoare în map-ul "result", dar cu semn schimbat. La sfârșit, metoda returnează mapă "result" care conține suma polinoamelor.

- **Metoda `multiplyPolinom(Map<Integer, Integer> p1, Map<Integer, Integer> p2)`**

Mai întâi, se creează o mapă goală numită "result". Apoi, se parcurg toate perechile de exponent și coeficient din map-urile "p1" și "p2", iar pentru fiecare pereche se calculează noul exponent și noul coeficient, astfel: noul exponent este suma dintre exponentul din primul polinom și exponentul din al doilea polinom, iar noul coeficient este produsul dintre coeficientul din primul polinom și coeficientul din al doilea polinom.

În continuare, pentru fiecare pereche nouă de exponent și coeficient obținute, se verifică dacă exponentul respectiv există deja în mapă "result". Dacă da, se adaugă la valoarea existentă (adică se adaugă noul coeficient la coeficientul anterior). Dacă nu există, se adaugă o nouă pereche de exponent și coeficient în mapă "result". La sfârșit, metoda returnează mapă "result" care conține produsul polinoamelor.

## **Clasa `PolynomialCalculatorGUI`**

-este o clasă GUI (Interfața grafică de utilizator) care utilizează biblioteca Swing pentru a crea o interfață de utilizator pentru un calculator de polinoame.

-această interfață conține câmpuri de text pentru introducerea a două polinoame și afișarea rezultatului, un meniu pentru selectarea unei operații (suma, scăderea sau înmulțirea), precum și un buton "Calculează" pentru a iniția operația.

- **`PolynomialCalculatorGUI()`**

Metoda constructor care este responsabilă pentru crearea interfeței grafice a utilizatorului (câmpuri de text, butoane, meniuri). Utilizează obiecte `GridBagConstraints` pentru a specifica poziționarea și dimensiunea fiecărei componente pe grila.

- **actionPerformed()**

Este o metoda care actioneaza atunci cand utilizatorul apasa butonul "Calculeaza". Dupa ce utilizatorul introduce polinoamele in casutele text, se creeaza doua obiecte Polinom utilizand metoda statica toPolynomial() din clasa Polinom si apoi aplica operatia corespunzatoare selectata din meniu folosind metodele addPolynomials(), subtractPolynomials() sau multiplyPolynomials(). Rezultatul este apoi afisat in campul de text rezultat.

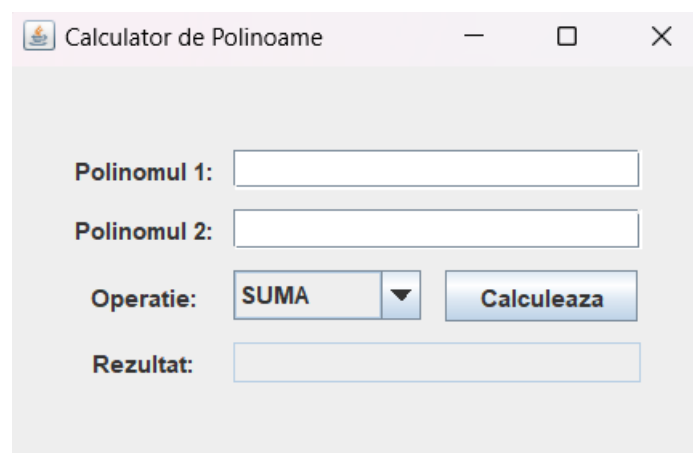
- **main()**

Este metoda principala care creeaza o instanta a clasei PolynomialCalculatorGUI si afiseaza fereastra.

## 5. Rezultate

Scenariile pentru testare: vom testa aplicatia calculand suma, scaderea si inmultirea a doua polinoame .

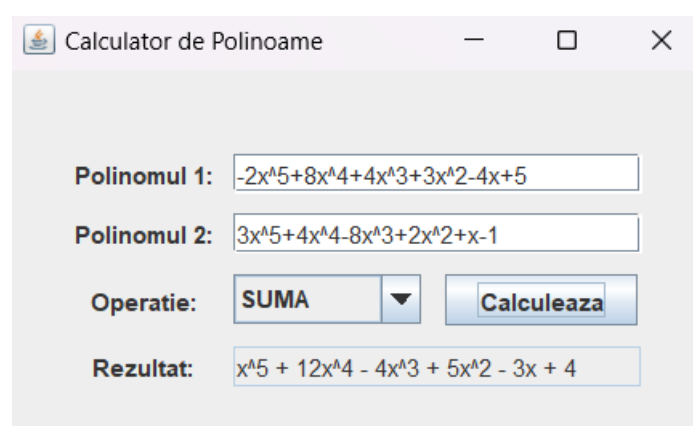
La deschiderea aplicatiei, putem observa interfata calculatorului de polinoame care are SUMA la "Operatie" pusa in mod implicit:



The screenshot shows a window titled "Calculator de Polinoame". It contains four text input fields labeled "Polinomul 1:", "Polinomul 2:", "Operatie:", and "Rezultat:". The "Operatie:" field has a dropdown menu with "SUMA" selected. To the right of the dropdown is a button labeled "Calculeaza".

Pentru a testa operatia de adunare a doua polinoame, utilizatorul va insera polinoamele  $-2x^5+8x^4+4x^3+3x^2-4x+5$  si  $3x^5+4x^4-8x^3+2x^2+x-1$  in casutele Polinomul 1 si Polinomul 2, si se apasa pe butonul "Calculeaza".

Rezultatul va fi  $x^5+12x^4-4x^3+5x^2-3x+4$ :



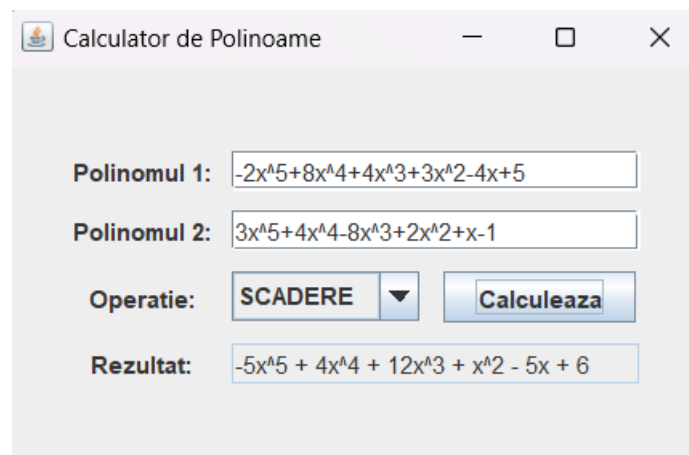
The screenshot shows the same window as before, but now the input fields contain the polynomials. "Polinomul 1:" contains  $-2x^5+8x^4+4x^3+3x^2-4x+5$  and "Polinomul 2:" contains  $3x^5+4x^4-8x^3+2x^2+x-1$ . The "Operatie:" dropdown still shows "SUMA". The "Rezultat:" field now displays the result:  $x^5 + 12x^4 - 4x^3 + 5x^2 - 3x + 4$ .



Pentru a testa operatia de scadere a doua polinoame, utilizatorul va insera polinoamele

“ $-2x^5+8x^4+4x^3+3x^2-4x+5$ ” si “ $3x^5+4x^4-8x^3+2x^2-1$ ” in casutele Polinomul 1 si Polinomul 2, si se apasa pe butonul “Calculeaza”.

Rezultatul va fi “ $-5x^5+4x^4+12x^3+x^2-5x+6$ ”:

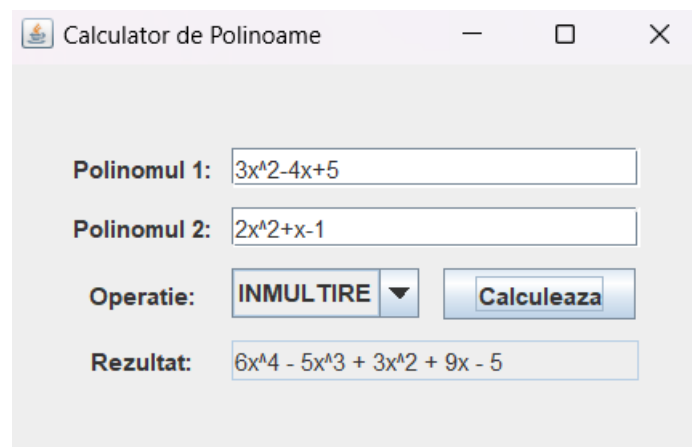


The screenshot shows a window titled "Calculator de Polinoame". It contains four input fields and two buttons. The first input field, labeled "Polinomul 1:", contains the polynomial  $-2x^5+8x^4+4x^3+3x^2-4x+5$ . The second input field, labeled "Polinomul 2:", contains the polynomial  $3x^5+4x^4-8x^3+2x^2+x-1$ . Below these is a dropdown menu labeled "Operatie:" with "SCADERE" selected. To the right of the dropdown is a button labeled "Calculeaza". At the bottom, a field labeled "Rezultat:" displays the result:  $-5x^5 + 4x^4 + 12x^3 + x^2 - 5x + 6$ .

Pentru a testa operatia de inmultire a doua polinoame, utilizatorul va insera polinoamele

“ $3x^2-4x+5$ ” si “ $2x^2+x-1$ ” in casutele Polinomul 1 si Polinomul 2, si se apasa pe butonul “Calculeaza”.

$(3x^2-4x+5) \cdot (2x^2+x-1) = 6x^4+3x^3-3x^2-8x^3-4x^2+4x+10x^2+5x-5$ , deci rezultatul va fi “ $6x^4-5x^3+3x^2+9x-5$ ”:



The screenshot shows the same "Calculator de Polinoame" window. The first input field, labeled "Polinomul 1:", contains the polynomial  $3x^2-4x+5$ . The second input field, labeled "Polinomul 2:", contains the polynomial  $2x^2+x-1$ . The dropdown menu labeled "Operatie:" now has "INMULTIRE" selected. The "Calculeaza" button is still present. The "Rezultat:" field at the bottom displays the result:  $6x^4 - 5x^3 + 3x^2 + 9x - 5$ .

## 6. Concluzii

În concluzie, am învățat cum să construiesc și să utilizez polinoame în Java și cum să construiesc o interfață grafică utilizând Swing. Această temă a fost utilă pentru a înțelege mai bine cum să utilizăm structurile de date și să construim aplicații utile.

Posibile dezvoltări ulterioare pentru acest program ar putea fi adăugarea de operații suplimentare, cum ar fi împărțirea, derivarea sau integrarea unui polinom. De asemenea, se poate adăuga validarea datelor de intrare pentru a se asigura că utilizatorii introduc datele corecte și pentru a evita alte erori.

## 7. Bibliografie

1. <https://www.javatpoint.com/java-swing>
2. <https://docs.oracle.com/javase/8/docs/api/java/util/Map.html>
3. <https://www.geeksforgeeks.org/stringbuilder-class-in-java-with-examples/>
4. <https://beginnersbook.com/2013/12/java-string-split-method-example/>
5. Prezentările suport din <https://dsrl.eu/courses/pt/>