



**TECHNICAL
UNIVERSITY**
OF CLUJ-NAPOCA
ROMANIA

FACULTATEA: Automatică și Calculatoare
SPECIALIZAREA: Calculatoare și Tehnologia Informației
DISCIPLINA: Tehnici de programare
Grupa 30226 | An 2 semestrul 2

Student:

Galiș George-Laurențiu



Cuprins

1. Obiectivul temei.....	3
2. Analiza problemei.....	3
3. Proiectare.....	4
4. Implementare.....	6
5. Rezultate.....	13
6. Concluzii.....	15
7. Webografie.....	15



1. Obiectivul temei

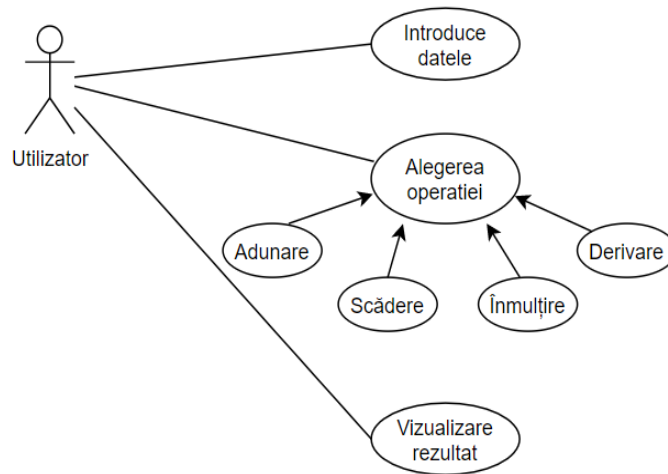
Obiectivul acestei teme pentru laborator este cel de a implementa un program care creeaza doua polinoame și face operații pe acestea. Printre operații se numără: adunarea, scădere, derivare și înmulțire. Toate aceste operații se fac ajutorul unei interfețe grafice. Această interfață grafică face ca programul sa fie mult mai ușor de folosit și de înțeles pentru utilizator deoarece totul se întâmplă vizual și ușor de înțeles.

2. Analiza problemei

Un polinom reprezintă o expresie matematică formată din monoame. Aceste monoame sunt formate la rândul lor dintr-un coeficient și exponent care sunt numere întregi.

Pentru a înțelege și proiecta mai ușor problema, se poate face o împartire pe situații în care este pus utilizatorul. În funcție de ce alegeri are utilizatorul, programatorul poate să-și facă o idee generală despre cum și în ce condiții ar trebui să se comporte programul.

Mai jos este reprezentată diagramă use-case care ajută la înțelegerea folosirii programului și opțiunile pe care le are de ales utilizatorul. Mai întâi utilizatorul trebuie să introducă coeficientii în dreptul exponentului respectiv, după care are de ales operația pentru polinoame și într-un final are de analizat și vizualizat rezultatul operației.



3. Proiectare

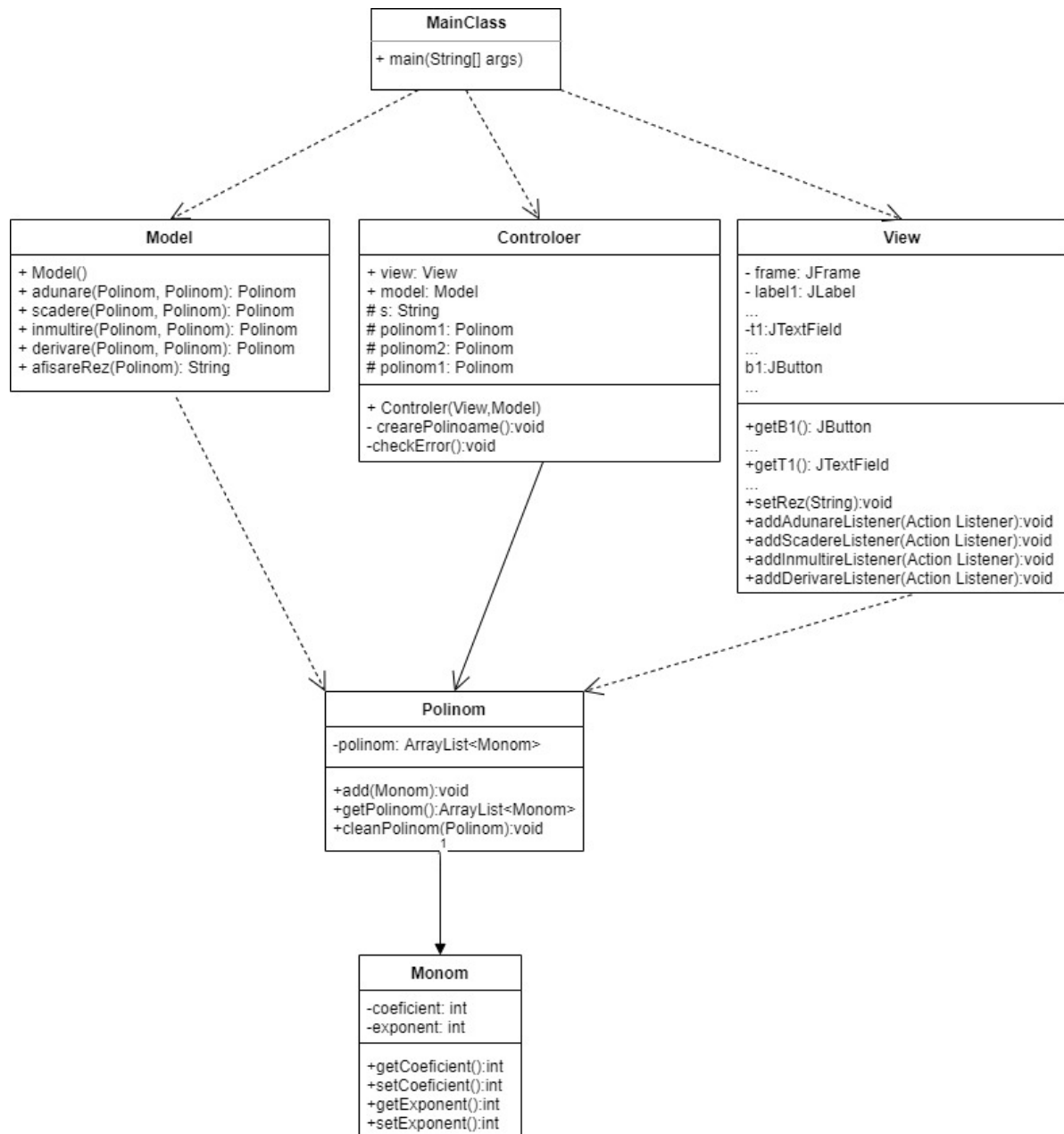
Metoda abordată pentru a citi datele care formează polinoamele respective este cât se poate de accesibilă deoarece se introduc doar coeficienții polinoamelor în câmpul exponentului predefinit deja de aplicație. Se putea alege și metodă introducerii manuale a polinoamelor de la tastatură prin text field-uri sub formă $(\text{coeficient}) \cdot X^{(\text{exponent})} + \dots = 0$ dar făcea citirea mult mai dificilă pentru utilizator.

Aplicația poate crea și face operații pe polinoame de grad maxim 5, dispuse de la gradul cel mai mare la gradul cel mai mic, deoarece exponentul nu influențează operațiile pe acestea, regulile fiind aceleași.



• Diagrama UML

Unified Modeling Language sau UML pe scurt este un limbaj standard pentru descrierea de modele si specificatii pentru software. UML a fost la bază dezvoltat pentru reprezentarea complexității programelor orientate pe obiect, al căror fundament este structurarea programelor pe clase, și instanțele acestora (numite și obiecte). Cu toate acestea, datorită eficienței și clarității în reprezentarea unor elemente abstracte, UML este utilizat dincolo de domeniul IT.





4. Implementare

- **Clasele:**

Clasa MainClass:

Este clasa în care se implementează structura Model-View-Controller. Această structura ajută la o implementare mult mai ușor de realizat în proiectele mai complexe, și fiecare structura are un rol bine definit.

Clasa View:

Această clasa formează interfață grafică și creează ceea ce vede și are de făcut utilizatorul cu aplicația.

Clasa Controller:

Această clasa este granița dintre interfață grafică (clasa View) și programul propriu-zis, unde se petrece interacțiunea utilizator-aplicație.

Clasa Model:

Este clasa în care au loc toate calculele. Operații de adunare, scădere, înmulțire și derivare.

Clasa Polinom:

Această clasa înglobează elementele care ajută la formarea unui polinom. Mai exact, fiecare polinom este format din unul sau mai multe monoame.

Clasa Monom:

Asemănător că la clasa Polinom, această clasa înglobează elementele principale la crearea unui monom: coeficient și exponent.



- **Metodele utilizate în clasa View:**

Această clasă este formată din getters and setters care preiau sau modifică informația din câmpurile ce formează interfața grafică cum ar fi campuri de text sau butoane.

- **Metodele utilizate în clasa Controller:**

În această clasă se găsesc metodele și subcasele care se ocupă cu partea din spate a aplicației și controlează în funcție de datele introduse de utilizator. Subclasele din această clasă implementează interfața ActionListener. Interfața le dă butoanelor un scop bine definit de programator cum ar fi: la apăsarea butonului de adunare, aplicația să afișeze rezultatul.

- **Metodele utilizate în clasa Model:**

Clasa Model conține metodele ce implementează operațiile făcute pe polinoame:

- **Adunare:** pentru început, inițializăm un monom și un polinom.
Avem argumente pentru această metodă :
(Polinom pol1, Polinom pol2). Pentru fiecare polinom, parcurgem element cu element, adică fiecare monom din fiecare polinom pentru a căuta monoame cu aceeași exponenți. Odată găsite se efectuează adunarea propriu zisă: creem un nou monom cu coeficientul reprezentat de suma celor 2 coeficienți , și exponentul același. Monomul rezultat este returnat de metodă. Ulterior, pentru monoamele care au fost notate cu parcurs negativ, adică nu au avut pereche că exponent, sunt introduse și ele în lista.



- **Scădere:** pentru început, inițializăm un monom și un polinom.
Avem argumente pentru această metodă :
(Polinom p1, Polinom p2). Pentru fiecare polinom, parcurgem element cu element, adică fiecare monom din fiecare polinom pentru a căuta monoame cu aceeași exponenți. Odată găsite se efectuează adunarea propriu zisă: creem un nou monom cu coeficientul reprezentat de suma celor 2 coeficienți , și exponentul același Monomul rezultat este returnat de metodă. Ulterior, pentru monoamele care au fost notate cu parcurs negativ, adică nu au avut pereche că exponent, sunt introduse și ele în lista. Diferența față de adunare e că aici la parcursul polinomului 2, cel care se scade, monoamele fără pereche primesc același coeficient dar cu semn negative.
(0 - m2.getCoeficient()).

- **Înmulțire:** pentru început, inițializăm un monom și un polinom.
Avem argumente pentru această metodă :
(Polinom p1, Polinom p2). Pentru fiecare polinom, parcurgem element cu element, adică fiecare monom din fiecare polinom și le înmulțim fiecare cu fiecare, înmulțind coeficientii perechilor de monoame, și adunând exponenții .

- **Derivare:** pentru început, inițializăm un polinom.
Avem argument pentru această metodă :
(Polinom p1) deoarece derivarea se realizează doar asupra unui singur polinom, încă cazul nostru polinomul 1. Aici se parcurge acest polinom, iar pentru fiecare monom găsit îi schimbăm coeficientul că fiind produsul dintre vechiul coeficient și exponent, iar exponentul devine vechiul exponent - 1.



- **Metodele utilizate în clasa Polinom:**

Clasa este formată din 3 metode:

```
public void add(Monom m) {  
    polinom.add(m);  
}  
  
public ArrayList<Monom> getPolinom() {  
    return polinom;  
}  
  
public static void cleanPolinom(Polinom p) {  
    p.polinom.clear();  
}
```

- **Metodele utilizate în clasa Monom:**

Clasa este formată din getters și setters pentru exponent și coeficient:

```
public int getCoefficient() {  
    return coeficient;  
}  
  
public void setCoefficient(int coeficient) {  
    this.coeficient = coeficient;  
}  
  
public int getExponent() {  
    return exponent;  
}  
  
public void setExponent(int exponent) {  
    this.exponent = exponent;}  
}
```



• GUI:

Interfața grafică sau Graphical User Interface este o interfață cu utilizatorul bazată pe un sistem de afișaj ce utilizează elemente grafice. Interfața grafică este numit sistemul de afișaj grafic-vizual pe un ecran, situat funcțional între utilizator și dispozitive electronice. Folosim o interfața grafică User-Friendly cu scopul de a putea fi folosit acest calculator de polinoame și de persoane non-specializate.

Interfața grafică cuprinde următoarele elemente:

Frame = “ramă” în care se adaugă toate elementele de care avem nevoie pentru bună funcționare a programului. E practic o fereastră care, după utilizare, apăsând butonul X din dreapta sus se poate închide (Exit_on_close).

Butoane = sunt în număr de 4, fiecare cu altă funcționalitate la apăsare:

- + face adunarea dintre cele 2 polinoame
- - face scăderea dintre cele 2 polinoame
- * face înmulțirea dintre cele 2 polinoame
- ‘ derivează primul polinom

TextField = spații dreptunghiulare în care se pot introduce date de la tastatură. Dar pe lângă asta pot fi folosite și pentru a afișa rezultatul fără a se putea introduce date de la tastatură.

- **t1-t6:** spațiile în care se introduc coeficienții pentru primul polinom
- **t7-t12:** spațiile în care se introduc coeficienții pentru al doilea polinom
- **r :** spațiu în care se afișează rezultatul final (acest spațiu nu se poate modifica de la tastatură)

Label = este efectiv o etichetă, care poate fi titlul, informații, indicații etc. care pot ajuta utilizatorul să folosească programul. În cazul nostru e folosit doar pentru a indica ce exponent are fiecare coeficient și se află în dreptul unui TextField



Tema 1

Calcul Polinoame

Polinom 1: X^5 + X^4 + X^3 + X^2 + X^1 + X^0 = 0

Polinom 2: X^5 + X^4 + X^3 + X^2 + X^1 + X^0 = 0

Operatie:



• Funcționare aplicație:

În main-ul programului sunt instantiate clasele Model, View și Controller, deci de fiecare dată când se inițializează clasa main se va instanția o nouă interfață grafică.

Pentru fiecare button, s-a făcut o metodă nouă care implică ActionEvent. Astfel, de fiecare dată când are loc o acțiune la un buton, de exemplu a fost apăsat, ActionEvent-ul denumit e, transmite informația la ActionListener care așteaptă astfel de informații. Apoi, au loc evenimentele ce se află în metodă respectivă (în general: se declară un Polinom nou, denumit rezultat, la care se atribuie rezultatul operației alese).

În această aplicație, la apăsarea unui buton se apelează mai multe metode succesiv. De aceea am considerat implementarea acestui lucru în clasa Controller. Evenimentele la apăsarea unui buton sunt:

1. Se crează polinomul prin metodă createPolinoame() care citește din TextField-uri informațiile despre coeficienți
2. Metodele din clasa Model returnează rezultatul fiecărei operații și le pune într-o variabilă de tip Polinom
3. Se verifică dacă utilizatorul a introdus sau nu date în TextField-uri prin metodă checkError(). În cazul în care nu a introdus date, se va deschide fereastra:



4. Se setează TextField-ul unde trebuie să se afișeze rezultatul cu rezultatul returnat de metodă
5. Se șterg informațiile citite din fiecare polinom prin metodă cleanPolinom()

5. Rezultate

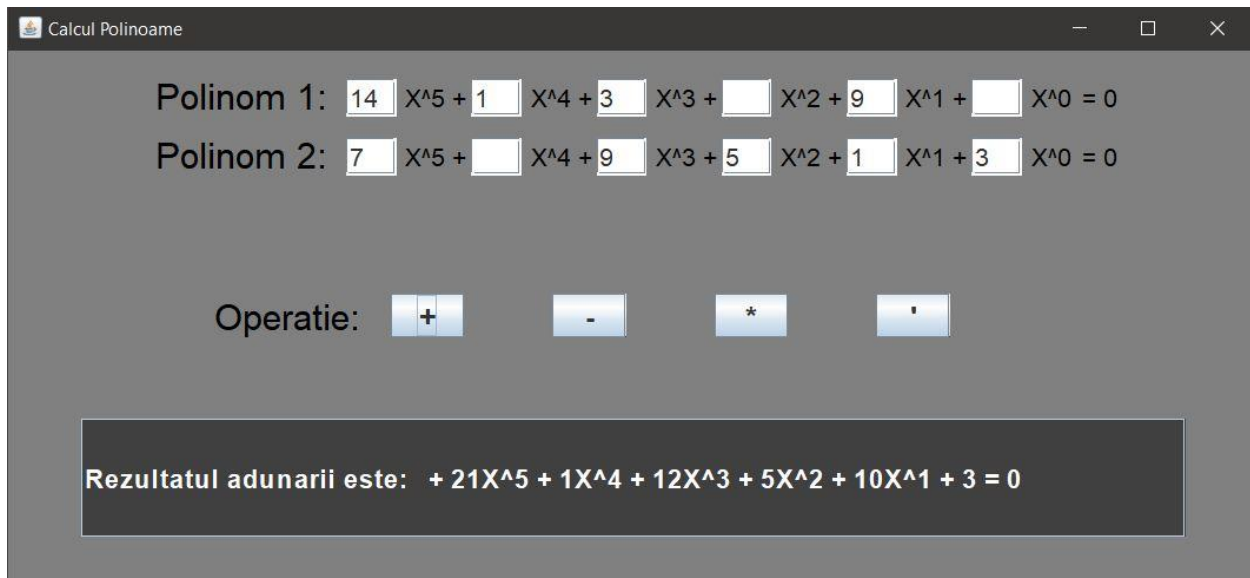
Implementarea operațiilor a fost făcută pentru aceleași intrări.

Adunarea: etapele necesare efectuării operației de adunare sunt:

-introducere coeficienți pentru polinom 1, doar pentru gradul dorit

-introducere coeficienți pentru polinom 2, doar pentru gradul dorit

După ce cele două polinoame sunt formate, se apasă butonul + pentru efectuarea propriu zisă a operației de adunare dintre cele 2 polinoame. Rezultatul va fi afișat în textField-ul rezultat.



The screenshot shows a window titled "Calcul Polinoame" with a dark gray background. It contains two rows of input fields for polynomial coefficients. The first row is labeled "Polinom 1:" and the second "Polinom 2:". Each row has input boxes for coefficients of X^5 , X^4 , X^3 , X^2 , X^1 , and X^0 , followed by "= 0". Below the input fields, there is a section labeled "Operatie:" with four buttons: "+", "-", "*", and "/". At the bottom, a text box displays the result of the addition: "Rezultatul adunarii este: + 21X^5 + 1X^4 + 12X^3 + 5X^2 + 10X^1 + 3 = 0".

Scăderea: etapele necesare efectuării operației de scădere sunt:

-introducere coeficienți pentru polinom 1, doar pentru gradul dorit

-introducere coeficienți pentru polinom 2, doar pentru gradul dorit

După ce cele două polinoame sunt formate, se apasă butonul - pentru efectuarea propriu zisă a operației de scădere dintre cele 2 polinoame. Rezultatul va fi afișat în textField-ul rezultat.



Tema 1

Calcul Polinoame

Polinom 1: X⁵ + X⁴ + X³ + X² + X¹ + X⁰ = 0

Polinom 2: X⁵ + X⁴ + X³ + X² + X¹ + X⁰ = 0

Operatie:

Rezultatul scaderii este: + 7X⁵ + 1X⁴ + (-6)X³ + (-5)X² + 8X¹ + (-3) = 0

Înmulțirea: etapele necesare efectuării operației de înmulțire sunt:

-introducere coeficienți pentru polinom 1, doar pentru gradul dorit

-introducere coeficienți pentru polinom 2, doar pentru gradul dorit

După ce cele două polinoame sunt formate, se apasă butonul * pentru efectuarea propriu zisă a operației de înmulțire dintre cele 2 polinoame. Rezultatul va fi afișat în textField-ul rezultat.

Calcul Polinoame

Polinom 1: X⁵ + X⁴ + X³ + X² + X¹ + X⁰ = 0

Polinom 2: X⁵ + X⁴ + X³ + X² + X¹ + X⁰ = 0

Operatie:

Rezultatul inmultirii este: + 98X¹⁰ + 27X⁶ + 9X² = 0



Tema 1

Derivarea: etapele necesare efectuării operației de derivare sunt:

- introducere coeficienți pentru polinom 1, doar pentru gradul dorit
- apăsare buton Submit P1 pentru “culegerea” datelor dar și pentru afișarea polinomului format în textField-ul afisarePolinom1.

După ce polinomul este format se apasă butonul ‘**Derivare**’ pentru efectuarea propriu zisă a operației de derivare. Rezultatul va fi afișat în textField-ul rezultat.

The screenshot shows a Java application window titled "Calcul Polinoame". It contains two polynomials defined by coefficients in text boxes:

Polinom 1: X⁵ + X⁴ + X³ + X² + X¹ + X⁰ = 0

Polinom 2: X⁵ + X⁴ + X³ + X² + X¹ + X⁰ = 0

Below the polynomials are four buttons for operations: "+", "-", "*", and "/".

At the bottom, a text box displays the result of the derivative of P1: "Rezultatul derivarii lui P1 este: + 70X⁴ + 4X³ + 9X² + 9 = 0".

6. Concluzie

În concluzie, acest proiect a avut scopul de a aprofunda cunoștințele în tot ce înseamnă limbajul Java, implementarea paradigmelor OOP, crearea unui program cu o interfață grafică. Mai mult de atât cred ca a avut și scopul de a reaminti tehnicile de programare învățate semestrul trecut despre tot ce înseamnă sintaxa și aranjarea unui cod ca să poată fi înțeles foarte ușor și de un alt programator.

7. Webografie

1. Wikipedia
2. Youtube