DOCUMENTATIE

TEMA 1

NUME STUDENT:Cândea Claudiu

GRUPA:30227

# CUPRINS

[1. Obiectivul temei 3](#_Toc95297885)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](#_Toc95297886)

[3. Proiectare 5](#_Toc95297887)

[3](#_Toc95297887)[4. Implementare 6](#_Toc95297888)

[5. Rezultate 9](#_Toc95297889)

[6. Concluzii 9](#_Toc95297890)

[7. Bibliografie 10](#_Toc95297891)

1. **Obiectivul temei**

Obiectivul principal al temei este realizare unui calculator ce poate efectua urmatoarele operatii asupra polinoamelor: adunare, scadera, inmultire, impartire, derivare si integrare.

Obiective secundare:

1. Analiza problemei si identificare cerintelor
2. Proiectare calculatorului polinomial
3. Implementare calculatorului polinomial
4. Testare calculatorului polinomial

# Analiza problemei, modelare, scenarii, cazuri de utilizare

Cerintele functionale ale proiectului sunt urmatoarele:

-Utilizatorului ii este oferita posibiliatea introduceri in interfata a 2 polioname asupra carora se pot efectua diverse operatii;

-Utilizatorul poate selecta din interfata operatia pe care doreste sa o efectueze asupra polinoamelor introdus si poate vedea tot in interfata rezultatul acestei operatii;

-Calculatorul trebuie sa poate efectua adunarea, scaderea, inmultirea, impartirea asupra celor 2 polinoame introduse in interfata si operatiile de derivare si integrare asupra primului polinom;

Cerintele non-functionale ale proiectului sunt urmatoarele:

-Interfata trebuie sa fie intuitiva si usor de folosit de catre utilizator;

-Interfata trebuie sa ofere butoane care permit scrierea polinoamelor direct din interfata;

**Use Case**: Adunarea polinoamelor

**Actor principal**: Utilizatorul

**Scenariu de succes**:

1. Folosind tastatura sau butoanele din interfata utilizatorul introduce 2 polinoame in cele 2 casete text destinate acestor;
2. Utilizatorul selecteaza din interfata butonul „Add”;
3. In caseta text destinata „Result” se va afisa rezultatul adunarii celor 2 polinoame;

**Use Case**: Scadera polinoamelor

**Actor principal**: Utilizatorul

**Scenariu de succes**:

1. Folosind tastatura sau butoanele din interfata utilizatorul introduce 2 polinoame in cele 2 casete text destinate acestor;
2. Utilizatorul selecteaza din interfata butonul „Sub” ;
3. In caseta text destinata „Result” se va afisa rezultatul scaderii celor 2 polinoame;

**Use Case**: Inmultirea polinoamelor

**Actor principal**: Utilizatorul

**Scenariu de succes**:

1. Folosind tastatura sau butoanele din interfata utilizatorul introduce 2 polinoame in cele 2 casete text destinate acestor;
2. Utilizatorul selecteaza din interfata butonul „Mul”;
3. In caseta text destinata „Result” se va afisa rezultatul inmultirii celor 2 polinoame;

**Use Case**: Impartirea polinoamelor

**Actor principal**: Utilizatorul

**Scenariu de succes**:

1. Folosind tastatura sau butoanele din interfata utilizatorul introduce 2 polinoame in cele 2 casete text destinate acestor;
2. Utilizatorul selecteaza din interfata butonul „Div” ;
3. In caseta text destinata „Result” se va afisa rezultatul impartirii celor 2 polinoame;

**Use Case**: Derivarea polinomului

**Actor principal**: Utilizatorul

**Scenariu de succes**:

1. Folosind tastatura sau butoanele din interfata utilizatorul introduce 1 polinom in prima caseta text ;
2. Utilizatorul selecteaza din interfata butonul „Derivare” ;
3. In caseta text destinata „Result” se va afisa rezultatul derivarii polinomului;

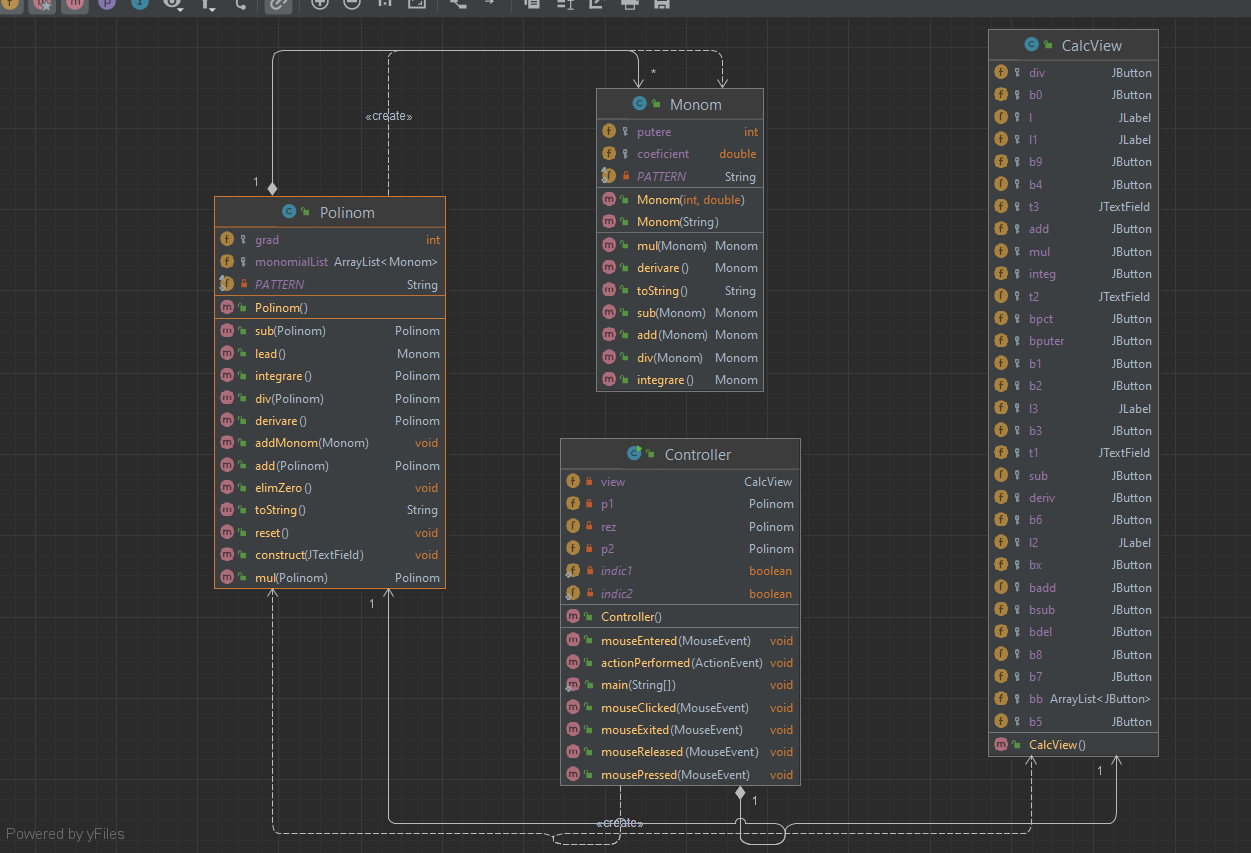
**Use Case**: Integrarea polinomului

**Actor principal**: Utilizatorul

**Scenariu de succes**:

1. Folosind tastatura sau butoanele din interfata utilizatorul introduce 1 polinom in prima caseta text ;
2. Utilizatorul selecteaza din interfata butonul „Integrare” ;
3. In caseta text destinata „Result” se va afisa rezultatul integrarii polinomului;

# Proiectare



Am inplementat proiectul folosit o arhitectura de tip Model-View-Controller(MCV). Clasele Monom si Polinom repezinta partea de model, specificant cum sunt reprezentate datele in program si ce operatii se pot efectua asupra acestora. Clasa CalcView reprezinta partea de view, fiind interfata grafica a programului ce ofera utilizatorului un mod mai simplu si mai placut de a interactiona cu o programul. Clasa Controller leaga partea de model care de partea de view.

Clasa View mosteneste clasa JFrame, astfel aceasta este o fereasta ce permite crearea unei interfete grafice.

Clasa Controller implementeaza interfetele ActionListener si MouseListener, astfel, prin implementarea metodelor acestor interfete , se asigura ca interfata grafica este interactiva si interactioneaza cu modelul.

In clasa Polinom am folosi un ArrayList de Monoame pentru a reprezenta monoamele care formeaza polinomul.

Intre clasa Monom si clasa Polinom exista o relatie de agregare. Intre clasa Controller si clasele Polinom si CalcView exista, de asemenea, relatii de agregare.

# Implementare

**Clasa Monom**

Aceasta clasa a fost folosita pentru a modela in programul nostru conceptul matematic de monom(coeficient\*x^putere). Ca variabile instanta avem 1 camp de tip int ce reprezinta putera monomului (campul putere) si un camp de tip double ce reprezinta coeficientul monomului (camul coeficient). Coeficientul este de tip dobule deoarece in urma integrarii monomului coeficientii pot deveni valori rationale. Exista o constata de clasa, string-ul PATTERN utilizat la la citirea prin intermediul interfetei grafice a datelor folosing metoda pattern matching. Puterea si coeficientul au vizibilitatea protected si PATTERN este private.

Aceasta clasa are 2 constructori, unul primeste ca parametrii o valoare de tip int si una de tip double ce reprezinta puterea si coeficientul polinomului ce va fi creat si un al doilea ce primeste un String, folosit la crearea monoamelor prin intermediul datelor provenite de la utilizator prin interfata grafica.

Metoda add aduna 2 monoame astfel: daca puterile lor sunt egale atunci monomul rezultat va avea aceeasi putere iar coeficientul acestuia va fi suma coeficientilor monoamelor intiale. Daca puterile coeficientilor nu sunt egale se va arunca o exceptie.

Metoda sub scade 2 monoame astfel: daca puterile lor sunt egale atunci monomul rezultat va avea aceeasi putere iar coeficientul acestuia va fi diferenta coeficientilor monoamelor intiale. Daca puterile coeficientilor nu sunt egale se va arunca o exceptie.

Metoda mul inmulteste 2 monoame. Puterea monomului rezultat va fi suma puterilor monoamelor intiale si coeficientul acestuia va fi produsul coeficientilor monoamelor initlale.

Metoda div realizeaza impartirea a 2 monoame. Impartirea are loc doar daca coeficentul monomului ce va fi impartiorul este diferit de 0 si puterea acestuia este mai mica decat puterea deimpartitului. Puterea monomului rezultat va fi diferenta puterilor monoamelor intiale si coeficientul acestuia va fi rezultatul impartirii coeficientilor monoamelor initlale.

Metoda derivare realizeaza derivarea unui monom. Daca puterea monomului initial este 0, atunci se va returna un monom cu putere si coeficient 0, altfel se va returna un monom care are puterea mai mica cu 1 decat cel initial iar coeficientul sau va fi egal cu produsul dintre coeficientul si putera monomului initial.

Toate metodele si contructorii acestei clase au vizibilitatea public, metodele returneaza monoame..

**Clasa Polinom**

Aceasta clasa a fost folosita pentu a modela in aplicatia noasta conceptul matematic de polionom. Ca variabile instanta am folosti un arrayList de monoame pentru a reprezenta monoamele componente ale polinomului, aceasta list am numit-o monomialList si o variabil grad de tip int pentru a gradul polinomului, care reprezinta puterea cea mai mare la care se afla un monom in lista de monoame. Exista o constata de clasa, string-ul PATTERN utilizat la la citirea prin intermediul interfetei grafice a datelor folosing metoda pattern matching. Gradul si monomialList au vizibilitatea protected si PATTERN este private.

Clasa are un singur constructor care creeaza un polinom cu grad = -1 si cu un arrayList gol.

Metoda publica construct primeste ca paramteru un JTextField si folosind metoda pattern matching prelucreaza pe rand fiecare monom identificat in stringul continu de text field si il adauga in monomialList.

Metoda add efectueza adunarea a 2 polinoame folosindu-se de metoda de adunare a monoamelor descrisa in clasa Monom. Cu 2 for each se parcurg polinoamele si se aduna monoamele care au aceasi putere , iar monomul rezultat se adauga in lista de monoame a unui al treilea polinom ce reprezinta valoarea returnata de catrea aceasta metoda. Se mai foloseste si o variabila locala booleana pentru a indica daca monomul procesat din primul polinom a fost implicat intr-o adunare la parcurgea celui de-al doilea polinom, daca nu a fost atunci acesta va fi adaugat in rezulta fara a se mai efecuta o operatie asupra lui. In final se mai parcurg inca o data cu 2 for each-uri cele 2 polinoame initiale, insa de data aceasta for-ul exterior va parcurge cel de-al doilea polionom. Scopul acestei parcurgeri este de a gasit monoamele celui de al doilea polinom ce nu am fost implicat in vreo adunare si de le adauga in rezulat, de asemenea si aici se foloseste o variabila locala booleana.

Metoda sub efectueza scadera a 2 polinoame folosindu-se de metoda de scadere a monoamelor descrisa in clasa Monom. Cu 2 for each se parcurg polinoamele si se scad monoamele care au aceasi putere , iar monomul rezultat se adauga in lista de monoame a unui al treilea polinom ce reprezinta valoarea returnata de catrea aceasta metoda. Se mai foloseste si o variabila locala booleana pentru a indica daca monomul procesat din primul polinom a fost implicat intr-o scadere la parcurgea celui de-al doilea polinom, daca nu a fost atunci acesta va fi adaugat in rezulta fara a se mai efecuta o operatie asupra lui. In final se mai parcurg inca o data cu 2 for each-uri cele 2 polinoame initiale, insa de data aceasta for-ul exterior va parcurge cel de-al doilea polionom. Scopul acestei parcurgeri este de a gasit monoamele celui de al doilea polinom ce nu am fost implicat in vreo scadere si de a le adauga in rezulat, insa coeficientul va avea semn schimbat, de asemenea si aici se foloseste o variabila locala booleana.

Metoda mul efectueaza inmultirea a 2 polinoame folosindu-se de metoda de inmultirea a monoamelor. Cu 2 for each-uri se parcurg cele 2 polinoame si se efectueaza inmultirea fiecarui monom din primul cu fiecare monom din al doilea. Monomul rezultat se adauga intr-un a treilea polinom ce reprezinta rezultatul returnat de metoda aceasta. Inainte de a fi aduga in rezultat se parcurge odata polinomul ce va fi returna pentru a vedea daca exista deja un monom in acesta care sa aiba aceeasi putere cu monomul rezultat recent, iar daca exist atunci se va realiza o adunare intre acestia si in rezulta va exista un singur monom la putere respectiva.

Metoda div efectueaza impartirea a 2 polinoame folosind-se de metoda de impartirea a monoamelor. Impartirea are loc doar daca impartitorul este diferint de 0, in cazul programului nostru monomialList.size diferit de 0, in caz contrar rezulatul returnat este un polinom cu valoare null. Impartirea are loc conform algoritmului descris cerinta temei.

Metoda derivare aplica operatia de derivare asupra unui polinom. Se parcurge fiecare monom al polinomului cu un for each si se aplica asupra lor metoda de derivare a monoamelor, iar rezultatele sunt adaugate intr-un polinom ce va fi returnat ca rezultat al acestei metode.

Metoda integrare aplica operatia de integrare asupra unui polinom. Se parcurge fiecare monom al polinomului cu un for each si se aplica asupra lor metoda de integrare a monoamelor, iar rezultatele sunt adaugate intr-un polinom ce va fi returnat ca rezultat al acestei metode.

Toate metodele ce implementeaza operatiile pe polinoame au vizibilitatea public.

**Clasa CalcView**

Aceasta clasa implementeaza interfata grafica a aplicatiei, mostenind clasa JFrame.

Ca variabile instanta are:

* 4 JLabel-uri;
* 3 JTextField-uri, 2 pentru a introduce polinoamele si unul pentru a afisa rezultatul operatiilor;
* 6 butoane (JButton) cu numele celor 6 operatii pe polinoame, a caror apasare ar trebui sa determine realizare operatiei indicate;
* 9 butoane (JButton) cu cifre de la 0 la 9 pentru a putea tasta valorile polinoamelor;
* 6 butoane (JButton) cu simbolurile +, -, ^, ., x pentru a putea tasta valorile polinoamelor;
* 1 buton (JButton) numit del a carui apasate va sterge continutul din cele doua casete text destinate introducerii polinoamelor;

Toate aceste obiecte au vizibiliatea protected.

Aceasta clasa mai are un constructor care are rolul de a initializa si de a aseza in fereastra obiectele ce compun interfata grafica. Se seteaza dimensiunile ferestrei, numele ferestrei, optiunea de incghidere a inchidere a ferestrei, butoanele pentru cifre sunt puse intr-un arrayList pentru a fi manipulate mai usor, obiectele sunt impartitea in mai multe panel-uri pentru a face interfata grafica mai intuitiva, panel-urilor se se asociaza layout-uri, si in final fereastra se seteaza ca find vizibila.

**Clasa Controller**

Aceasta clasa asigura legatura dintre model si interfata grafica. Clasa implementeaza interfetele ActionListener si MouseListener. Ca variabile instanta are 2 polinoame, p1 si p2, un CalcView numit view si 2 valori boolene numite indic1 si indic2. Toate acestea au vizibiliatea private.

Constructoul acestei clase nu primeste niciun parametru. El initializeaza variabilele instanta si adauga ascultatori obiectelor din view. Intrucat Controller-ul implementeaza cele 2 interfete de ascultatori, ascultatorul adaugat obiectelor din view este obiectul „this”.

In aceasta clasa se afla metoda main care porneste aplicatia.

Metoda actionPerformed primeste ca parametru un actionEvent e, si in functie de susa evenimentului va avea loc o operatie:

* polinoame din view, atunci se va realiza operatia respectiva asupra polinoamelor introduse de utilizator in textField-uri;
* Daca sursa evenimentului a fost apasare unuia din butoanele cu cifre sau cu simboluri din view, atunci cifra sau simbolul respectiv va fi scris in unul din cele 2 textField-uri destinate polinoamelor ce reprezinta intput-ul aplicatie.
* Daca sursa evenimentului a fost butonul del din view, atunci se va sterge continutul din cele 2 textField-uri destinate polinoamelor ce reprezinta intput-ul aplicatie.

Metoda mouseClicked primeste ca parametru un MouseEvent e. Daca obiectul asupra caruia s-a efectuat click-ul a fost textField-ul pentru primul polinom atunci indic1 devine true si indic2 devine false, daca obiectul asupra caruia s-a efectuat click-ul a fost textField-ul pentru al doilea polinom atunci indic1 devine false si indic2 devine true. Cei doi indici sunt folositi pentru a indica in care textField se va scrie cand se v-a apasa un buton cu cifra sau cu simbol din view.

# Rezultate

In clasa de test ,TestClass, am testat operatiile pe polinoame: adunare, scadere, inmultire, impartire, derivare si integrare. Toate cele 6 teste au trecut cu succes.

Pentru testare adunarii am folosit polinoamele p1=+3x^3+7x^2-x^1+5 si p2=+x^2-x^1+1. Pentru testarea scaderii am efectuat operatia p1-p2, iar pentru testare impartirii am folosit operatia p1/p2. Pentru derivare si integrare am folosit polinomul p1, cu precizare ca am schimbat coeficientul monomului 7x^2 din 7 in 6. Pentru inmultire am folosit polinoamele 3x^2+2x^1 si 1x^2+x^1.

La testare inmultiri am urmarit ca in urma inmultirii monom cu monom sa rezulte monoame cu aceeasi putere pentru a verifica daca metoda de inmultire efectueaza si adunare acestora.

La testare impartirii am urmarit ca algoritmul sa indeplineasca fiecare etapa.

La testare integrarii am urmarit ca coeficientii afisati sa corespunda valorile rationale asteptate.

La testare derivarii am urmarit comporatmentul in derivare unui monom cu puterea 0.

La testare adunarii si scaderii am urmarit ca in rezultat sa apara si monoamele celor 2 operanzi asupra carora nu s-a efectuat nicio operatie din clasa monom.

# Concluzii

In urma acestei teme am invatat cum se aplica intr-un proiect real concepetele de programare orientate pe obiecte si modul in care se poate crea un program cu interfata grafica care permite utilizatorului sa interactioneze mult mai usor cu programul.

Ca dezvolare ulterioara se mai poate lucra la modul in care programul identifica polinoamele introduse de utilizator in interfata grafica, intrucat acum trebuie sa se specifice pentru fiecare monom coeficientul si putera, s-ar putea implementa ca programul sa recunoasca si faptul ca coeficientul este 1 atunci cand nu se scrie sau ca „x” defapt reprezinta x la putera 1.

# Bibliografie

<https://dsrl.eu/courses/pt/>

https://regex101.com/