

TEMA 1 – TEHNICI DE PROGRAMARE

SISTEM DE PROCESARE A POLINOAMELOR

Student:

Bruchental Maria Andreea

Grupa:30227

**Iunie, 2018**

Contents

[1. Obiectivul temei 3](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384320)

[2. Analiza problemei. Modelare. Scenarii. Cazuri de utilizare. 4](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384321)

[2.1. Analiza problemei 4](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384322)

[2.2. Modelare 4](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384323)

[2.3. Scenarii 5](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384324)

[2.4. Cazuri de utilzare 6](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384325)

[3. Proiectare 7](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384326)

[3.1. Diagrame UML 7](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384327)

[3.2. Structuri de date 8](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384328)

[3.3. Proiectare clase 9](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384329)

[4. Implementare si testare 11](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384330)

[4.1. Implementare 11](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384331)

[4.2. Testare 13](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384332)

[5. Rezultate 15](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384333)

[6. Concluzii, dezvoltari ulterioare 15](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384334)

[7. Bibliografie 17](file:///E:\FACULTATE\An%202\Sem2%20OLI\sem2\tp\documentatii\Documentatie1.docx#_Toc414384335)

# Obiectivul temei

**Cerinta:**  Propuneti, proiectati si implementati un sistem de procesare a polinoamelor de o singura variabila cu coeficienti intregi. Cu alte cuvinte, se cere proiectarea unui sistem care sa faca diverse operatii asupra unor polinoame date prin interfata intr-un mod ales de catre proiectant. Operatiile pot fi operatii asupra unuia sau doua polinoame.

**Obiectiv**: Se doreste crearea unei aplicatii Java, care sa realizeze o serie de operatii asupra unor polinome introduse de la tastatura sub forma de siruri de coeficienti. Operariile implementate sunt urmatoarele:

- citirea si afisarea unui polinom

- operatii efectuate asupra a doua polinoame:

* adunarea
* scaderea
* inmultirea
* impartirea

- operatii care pot fi efectuate asupra unui singur polinom, precum:

* derivarea
* integrarea

Pentru utilizare mai eficienta, voi proiecta si o intrefata grafica implementata cu elemente Swing care sa ii permita utilizatorului sa introduca polinoame si sa efectueze operatii asupra acestora . Interfata trebuie sa fie structurata in asa fel incat polinoamele sa se introduca intr-un mod cat mai logic cu putinta, iar efectuarea operatiilor sa se faca intr-un mod cat mai usor, care sa nu aduca dificultati utilizatorului. In ceea ce priveste modul de afisare al rezultatelor, acestea trebuie sa fie transpuse intr-un mod cat mai usor perceptibil de catre utilizator si intr-o locatie predefinita in acest scop. Voi implementa aplicatia dupa modelul MVC care ofera o mai buna structura a claselor si care ofera o separare a partii de functionare de cea de afisare. Interfata va fi si ea construita cu atentie, pentru ca fiecare buton sa se gaseasca la un loc cat mai potrivit iar fiecare fereastra de tip text da fie asezata in concordanta cu celelalte elemente deja stabilite.

# Analiza problemei. Modelare. Scenarii. Cazuri de utilizare.

### Analiza problemei

Polinomul este o functie cu o singura variabila,care poate fi exprimata sub urmatoarea forma generala:ax^n+bx^n-1+cx^n-2+…+z. Coeficientii unui polinom se vor pot pastra intr-un tablou unidimensional in ordine crescatoare sau descrecatoare dupa puterea lui x. Numim gradul polinomului puterea cea mai mare a lui x, pentru care coeficientul este diferit de 0. Asadar, sirul corespunzator coeficientilor va avea cel putin atatea elemente +1(pentru termenul liber), cat este graful polinomului.

Problema operatiilor asupra polinoamelor este una de o complexitatea destul de ridicata deoarece pot fi aplicate o serie extrem de larga de operatii asupra acestora. Pentru realizarea acestei teme avem nevoie de cumularea cunostintelor anterioare de Java dar si de matematica.

In proiectul de fata am ales sa implementez doar cateva dintre operatiile principale, considerate ca fiind cele mai importante cat si cele mai utile. Pe langa operatiile implementate, mai existau si altele care puteau fi implementate pentru lucrul cu polinoame, cum ar fi: calcului radacinilor intregi, care se determinau dintre divizorii termenului liber, graficul etc.

A trebuit sa aleg un model de implementare care sa fie cat mai intuitiv pentru utilizator dar care si sa imi permita sa operez intr-un mod cat mai optim asupra polinoamelor. In acelasi timp, un mare grad de importanta il are modul in care sunt stocate polinoamele, clasele care le definesc si legaturile dintre acestea.

### Modelare

Punctul de pornire ar fi modul de introducere al polinoamelor prin intermediul interfetei si modul in care acestea sunt prelucrate mai departe pentru fiecare operatie. Fiind utilizata o structura de tip MVC, interfata este izolata de partea de logica tocmai pentru o mai buna delimitare a componentelor aplicatiei; aceasta in considerentul in care sa se poata realiza modificari cu o mai mare usurinta si fara a se afecta alte componente decat cele vizate. Am ales ca fiecare dintre cele doua polinoame sa fie introdus sub forma unui sir de coeficienti, separati prin “,” iar la apasarea unui buton acesta sa fie prelucrat de catre aplicatie. La simpla introducere a unui polinom, aplicatia va afisa un mesaj care va confirma/infirma succesul adaugarii polinomului, in functie de datele adaugate. In ceea ce priveste operatiile, introtdeauna polinomul P va fi primul operand, iar polinomul Q va fi cel de-al II-lea. Fiecare operatie care se face asupra a doua polinoame, va avea un buton corespunzator. Si cele asupra unui singur polinom vor avea butoane corespunzatoare doar ca in cazul acestora va fi nevoie si sa se selecteze unul dintre cele doua polinoame(polinomul P este selectat implicit), iar in cazul unei erori la introducere sau in cazul in care polinomul selectat nu este adaugat, se va emite un mesaj de eroare in fereastra de rezultat. Pentru operatiile care au nevoie de valori introduse de la tastatura, exista text field-uri din care se preiau valorile corespunazatoare, tot la apasarea unui buton.

Implementarea aplicatiei dupa modelul MVC face ca ea sa aiba pachete de sine statatoare pentru partea de interfata, partea de logica si partea de controler care face legatura intre cele doua. Astfel, in partea de model vom avea clasele ce tin de operarea asupra poinoamelor, in partea de view avem implementata interfata iar in partea de controler este realizata imbinarea celor doua. Cu ajutorul controlatorului, modelului sau a view-ului putem manipula datele. Depinde de noi cum manipulăm și interpretăm aceste "date".

### Scenarii

**Introducerea polinoamelor, alegerea operatiei si realizarea acesteia**

Pentru o implementare cat mai optima a aplicatie, va trebui sa stabilim care sunt exact datele de care avem nevoie, deci va trebui sa alegem un model dupa care vom stoca infomatia corespunzatoare fiecarui polinom. Pentru a efectua oricare din operatiile dorite, am nevoie de date precum ccoeficientii si gradul fiecarui polinom. Pentru aceasta am ales sa folosim introducerea polinoamelor ca si vector de intregi, fara a cere insa gradul polinomului, acesta fiind dedus din vectorul de coeficienti. Am ales ca polinomul sa fie introdus incepand cu gradul cel mai semnificativ deoarece este mai intuitive. Astfel daca introducem valorile 1,3,0,-1, polinomul introdus va fi 1\*x^3+3\*x^2-1. La fiecare buton apasat va fi afisat in fereastra de rezultate un mesaj corespunzator. Astfel, la introducerea polinomului, aplicatia va afisa un mesaj de success insotit de polinomul adaugat, tocamai pentru ca utilizatorul sa poata verifica daca datele introduse de el, au fost preluate intocmai.

* daca coeficientii nu au fost introdusi sau nu s-au introdus in mod corectse va genera un mesaj de eroare; la operatiile efectuate asupara a 2 polinoame, operatiile nu se vor putea efectua daca nu au fost introduse ambele multimi de coeficienti ai celor doua polinoame; la operatiile pe un singur polinom daca nu a fost introdus, respectiv selectat polinomul asupra caruia dorim sa efectuam o operatie, se va genera din nou mesaj de eroare;
* in cazul operatiilor: calculul valorii intr-un punct, calcului derivatei intr-un punct, calculul integralei pe un anumit interval, va trebui sa introducem si alte valori necesare efectuarii operatiei; daca acestea nu au fost introduse sau nu s-au introdus numai cifre, se va afisa din nou mesaj de atentionare;
* scenariul ia sfarsit cand in casuta text apare rezultatul efectuarii operatiei alese;

Potentilii actori sunt utilizatorul care introduce polinoamele si operatiile de efectuat iar actorii secundari sunt polinoamele introduse.

### Cazuri de utilzare

Consider ca desi nu este indeajuns de complexa si de completa, aplicatia ar putea fi utilizata in scoli pentru interactivizarea orelor de matematica prin folosirea lui dupa terminarea predarii polinoamelor pentru a se verifica execitii deja rezolvate sau pentru verificarea cunostintelor acumulate. In concluzie, pentru rezolvarea acestui proiect trebuie sa luam in considerare procesarea polinoamelor, modelarea fiecarei operatii, preluarea informatiilor necesare la operatii si afisarea rezultatului dorit. Rezolvarea aleasa este una destul de clara si de o complexitate medie. In obtinerea solutiei s-a urmarit realizarea unei interfete cat mai usor de inteles si de utilizat pentru utilizator.

**Cazul de utilizare**: Efectuarea unei operatii asupra unui polinom-valoarea intr-un punct

**Actor**: Utlizatorul

**Scenariul de success:**

Utilizatorul introduce polinomul asupra caruia se doreste sa se opereze.

Utilizatorul il adauga prin apasarea butonului corespunzator.

Utilizatorul selecteaza polinomul asupra caruia sa se opereze adaugat.

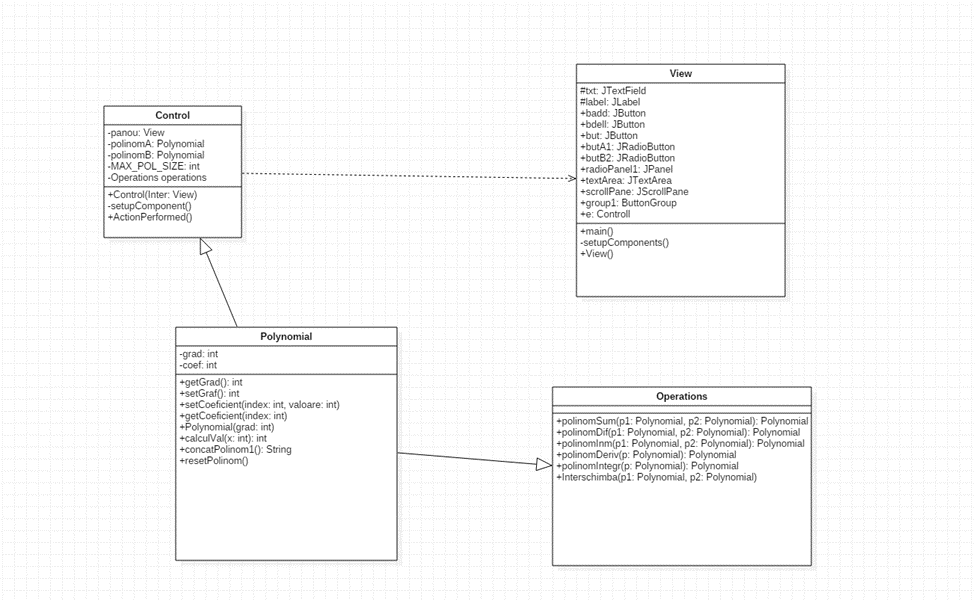
Rezultatul este afisat in fereastra de rezultate.

# Proiectare

### Diagrame UML

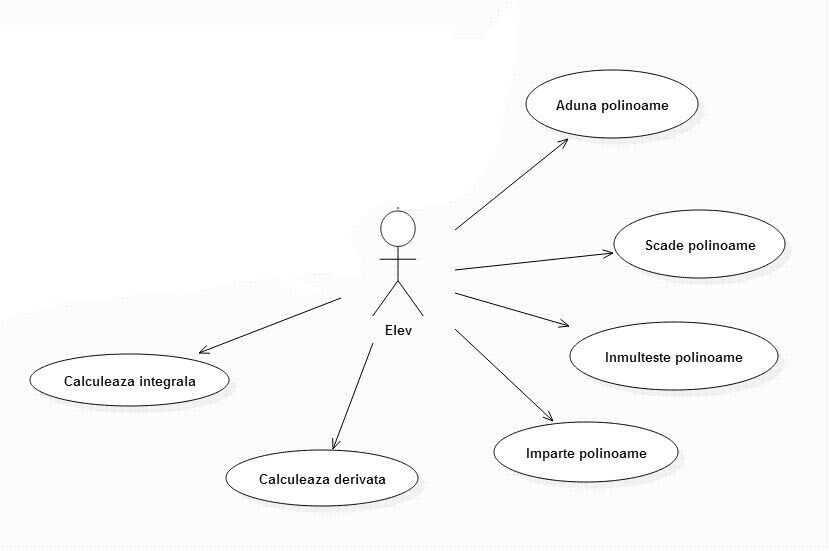
UML reuneşte cele mai bune tehnici şi practici din domeniul ingineriei progamării, care şi-au dovedit eficienţa în construcţia sistemelor complexe. Modelarea conceptuală este activitatea de identificare a conceptelor importante pentru sistem. În tehnica de programare orientată obiect, modelarea conceptuală se realizează prin diagrama claselor, întrucât clasele reprezintă concepte. O clasă se reprezintă printr-o căsuţă împărţită în trei. În partea de sus este notat numele clasei, în partea mediană atributele, iar în partea de jos operaţiile sale. Pentru o prezentre cat mai evidenta a structurii interne a programului, am ales sa constuiesc diagramele de clasa si cea use clasa.

Diagrama de clasa



Avand la baza modelul de proiectare MVC, controlul este clasa care reuneste interfata si polinomul. Astfel intre acesta si celelalte doua clase exista o relatie de asociere, intrucat interfata si polinomul participa in ascociatie avand un rol specific pe care il joaca. In acelasi timp si intre polinomul si interfata exista o astfel de relatie, deoarece interfata are la baza modul in care modelul este structurat si elemente ce tin de acesta.

**Diagrama use case**

****

### Structuri de date

Structurile de date folosite sunt printre cele mai simple.

* String – pentru preluarea, procesarea si apoi afisarea polinoamelor introduse in casuta text
* Tablou unidimensional – pentru pastrarea coeficientilor polinoamelor

### Proiectare clase

Conform modelului MVC, proiectul a fost impartit in trei parti mari, sub forma de pachete, fiecare fiind specializat pe o anumita sarcina. Astfel aplicatia are 3 pachete, polinoame, polinoameView si polinoameControler.

In pachetul polinoame exista 3 clase importante:

Clasa **Polinom:** este clasa care modeleaza forma unui polinom .

Clasa **Control** este clasa care contine Listenerii butoanelor. Pentru fiecare buton am creat metode de tip listener care sa sesizeze momentul in care s-a cerut efectuarea unei anumite operatii. In cazul text field-urilor in care se introduc valori de intrare, valorile adaugate sunt preluate si ele prin intermediul unor metode si transmise controlerului

Clasa **Execute** este clasa care contine metoda main din care practice porneste programul. In aceasta se reunesc cele trei component.

In pachetul **Interfata** exista o singura clasa care este total specializata pe interfata. Aceasta primeste ca parametru in constructor modelul si are definite ca variabile private, fiecare notiune, buton sau fereastra necesare in construirea interfetei. Am pozitionat fiecare element in fereastra cu ajuorul metodei setBounds iar pentru cele lacare a fost cazul, am setat diferite proprietati.

# Implementare si testare

### Implementare

Implementarea a fost realizata in mediul IntelliJ. Prima clasa proiectata a fost clasa Polinom, aceasta deoarece inainte de implementa restul aplicatiei, era recomandat sa stim exact cum arata obiectele cu care vom lucra. Mai apoi am proiectat restul claselor care contin metodele de jonglare cu obiectele de tip polinom.

### Testare

In ceea ce priveste testarea nu am terminat-o.

# Rezultate

In urma testarii aplicatiei prin interfata, se poate observa ca se pot efectua operatii asupra polinoamelor de orice grad cu precizarea ca fiecre coeficient sa fie separat de urmatorul prinr-o singura virgula. De asemenea coeficienti introdusi trebuie sa fie intregi pentru a putea fi procesati deoarece intregul proiect foloseste vectori de coeficienti de tip intreg. Desi la introducerea polinomului sunt ceruti si coeficientii 0, pentru a se putea afla gradul polinomului, la afisare acestia nu apar tocmai pentru ca utilizatorului sa ii fie mai usor sa preia rezultatele. Avand in vedere ca avem de-aface cu un vector de intregi, calcularea integralei nu va genera un rezultat exact, insa algoritmul este unul corect implementat si care cu cateva modificari, poate fi utilizat si pe alte tipuri de tablouri.

# Concluzii, dezvoltari ulterioare

In concluzie pot spune ca am reusit sa implementeaz un sistem de procesare a polinoamelor care realizeaza diferite operatii si care trateaza cazurile de utilizare destul de bine. Acest sistem este usor de utilizat deoarece are o interfata intuitiva, care nu cere prea multa informatie in ceea ce priveste polinoamele asupra carora se opereaza.

Fiind prima tema din cadrul acestei materii, dificultatea a fost mai ridicata, asta si din pricina faptului ca a trbuit sa implementez o interfata pentru utilizare.

Modelul MCV este o alta notiune pe care am reusit sa o aprofundez odata ce am realizat acest proiect. Am constatat ca este un model foarte util chiar si pentru proiectele mici, deoarece impune o organizare a claselor care sa impuna o departajare exacta intre ceea ce inseamna interfata si partea de logica. Aplicatia are o utilizare restransa deoarece putem folosi doar polinoame cu coeficienti intregi.

Dezvoltarile ulterioare pot tine atat de modul de implementare al operatiilor cat si de extinderea acestora. In acelasi timp, se pot realiza modificari si in cadrul interfetei.

-adaugarea de noi operatii(calcul radacini, generare grafic, etc);

-trecerea de la tablou de intregi, la tablou de double pentru o mai mare precizie in cadrul operatiilor

-introducerea unui meniu Help in care sa existe o documentatie despre folosirea aplicatiei

-gasirea unei alte metode de introducere a polinoamelor de la tastatura

# Bibliografie

[www.wikipedia.org](http://www.wikipedia.org)

<http://stackoverflow.com/questions/>

“Java de la 0 la expert”, Stefan Tanasa & Stefan Andrei, Editura Polirom, 2007

<http://www.tutorialspoint.com/design_pattern/mvc_pattern.htm>

http://math.stackexchange.com/questions/215734/pseudo-code-for-polynomial-long-division