Tehnici de programare

Documentatie:

Tema1:Procesare de polinoame

Student:

Muntean Andrei

Gr. 30227

Cuprins:

1.Obiectivul temei

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

3. Proiectare (diagrame UML, structure de date, proiectare clase, interfete, relatii, packages, algoritmi, interfata utilizator)

4. Implementare si testare(Cazuri de testare)

5. Concluzii, ce s-a invatat din tema, dezvoltari ulterioare

**1. Obiectivul temei**

* **Tema problemei :**

Propuneti,proiectati si implementati un sistem de procesare a polinoamelor de o variabila.

* **Obiective :**

Scopul acestui proiect este acela de a implementa o serie de operatii unui polinom de o variabila. Acestea sunt adunarea, scaderea, inmutirea, impartirea, derivarea si integrarea.

* **Teorie :**

În [matematică](http://ro.wikipedia.org/wiki/Matematică), un polinom este o [expresie](http://ro.wikipedia.org/w/index.php?title=Expresie_matematică&action=edit&redlink=1) construită dintr-una sau mai multe [variabile](http://ro.wikipedia.org/wiki/Variabilă) şi [constante](http://ro.wikipedia.org/w/index.php?title=Constante&action=edit&redlink=1), folosind doar operaţii de adunare, scădere, înmulţire şi ridicare la putere constantă pozitivă întreagă. ce566337a9a0075cd983524fa1fe045e este un polinom. Se observă în particular că împărţirea printr-o expresie de conţine o variabilă nu este permisă în polinoame.

**Generalitati :**

Polinoamele sunt construite din [termeni](http://ro.wikipedia.org/w/index.php?title=Termen_%28matematic%C4%83%29&action=edit&redlink=1) numiţi [monoame](http://ro.wikipedia.org/w/index.php?title=Monom&action=edit&redlink=1), care constau dintr-o constantă (numită [coeficient](http://ro.wikipedia.org/w/index.php?title=Coeficient&action=edit&redlink=1)) înmulţită cu una sau mai multe [variabile](http://ro.wikipedia.org/w/index.php?title=Variabile&action=edit&redlink=1) (de regulă reprezentate prin cifre). Fiecare variabilă poate avea un [exponent](http://ro.wikipedia.org/w/index.php?title=Exponent&action=edit&redlink=1) constant întreg pozitiv. Exponentul unei variabile dintr-un monom este egal cu [gradul](http://ro.wikipedia.org/w/index.php?title=Gradul_unui_polinom&action=edit&redlink=1) acelei variabile în acel monom. Pentru că *x* = *x*1, gradul unei variabile fără exponent este unu. Un monom fără variabile se numeşte *monom constant*, sau doar *constantă*. Gradul unui termen constant este 0. Coeficientul unui monom poate fi orice număr, inclusiv fracţii, numere iraţionale sau negative. Un polinom construit cu o singură variabilă se numeşte univariat.

De exemplu,

 -5x^2y\,este un monom. Are coeficientul -5, variabilele sunt *x* şi *y*, gradul lui *x* este doi, iar gradul lui *y* este unu.Gradul întregului monom este suma gradelor tuturor variabilelor din el. În exemplul de mai sus, gradul este 2 + 1 = 3.

Un polinom este o sumă de unul sau mai multe monoame. De exemplu, următoarea expresie este un polinom:

 3x^2 - 5x + 4\,.

El constă din trei monoame: primul are gradul doi, al doilea are gradul unu, iar al treilea are gradul zero.

Când un polinom este dispus în ordinea naturală, el are termenii de grad mai mare înaintea celor de grad mai mic. În primul termen, coeficientul este 3, variabila este *x*, iar exponentul este doi. În al doilea termen, coeficientul este -5. Al treilea termen este o constantă. Gradul unui polinom este cel mai mare grad al unui termen al său. În acest exemplu, polinomul are gradul doi.

Un polinom de gradul unu este numit liniar, unul de gradul doi este pătratic, iar unul de gradul trei este un polinom cubic. Mai rar, polinoamele de gradul patru se numesc cuartice iar cele de grad cinci quintice. Un polinom cu un singur termen este numit monom, unul cu doi termeni binom, iar unul cu trei termeni trinom. Un polinom care are coeficientul 1 pentru termenul de grad maxim se numeşte monic.

O expresie ce poate fi adusă la o formă polinomială prin aplicarea secvenţială a unor legi [comutative](http://ro.wikipedia.org/w/index.php?title=Comutative&action=edit&redlink=1), [asociative](http://ro.wikipedia.org/w/index.php?title=Asociative&action=edit&redlink=1), şi [distributive](http://ro.wikipedia.org/w/index.php?title=Distributive&action=edit&redlink=1) este în general considerată tot un polinom.De exemplu \frac{x^3}{12}

este un polinom pentru că este echivalent cu \tfrac{1}{12}x^3. Coeficientul este \tfrac{1}{12}.

Dar,  {1 \over x^2 + 1} \,

( 5 + y ) ^ x \,nu este polinom pentru că include împărţirea printr-o variabilă. La fel şi

pentru că are o variabilă la exponent.

Deoarece scăderea poate fi tratată ca o adunare cu termenul opus, şi deoarece ridicarea la o putere întreagă pozitivă şi constantă poate fi tratată ca înmulţire repetată, polinoamele se pot construi din constante şi variabile folosind doar două operaţii: adunarea şi înmulţirea.

O funcţie polinomială este o funcţie definită ca [evaluând](http://ro.wikipedia.org/w/index.php?title=Expresie_matematic%C4%83&action=edit&redlink=1) un polinom. De exemplu, funcţia *f* definită prin  f(x) = x^3 - x \,

este o funcţie polinomială. Funcţiile polinomiale sunt o clasă importantă de [funcţii derivabile](http://ro.wikipedia.org/w/index.php?title=Func%C5%A3ie_derivabil%C4%83&action=edit&redlink=1).

O ecuaţie polinomială este o [ecuaţie](http://ro.wikipedia.org/wiki/Ecua%C5%A3ie) în care un polinom este considerat egal cu un alt polinom.

 3x^2 + 4x -5 = 0 \,

este o ecuaţie polinomială.

**Proprietetile polinoamelor :**

Forma algebrica a unui polinom:



*Gradul unui polinom:*



*Egalitatea polinoamelor:*



**Operatii cu polinoame :**



Proprietati:

* comutativa f+g=g+f
* asociativa (f+g)h=f(g+h)
* exista element neutru fata de adunare f+f0=f;
* exista polinom opus oricarui polinom f+(-f)=0



Proprietati:

* comutativa
* asociativa
* element neutru 1

**Alte proprietati :**

Valoarea unui polinom:



**Proprietati:**





Functia polinomiala asociata unui polinom:



Observatie: Daca f:C, nu putem vorbi despre graficul lui f, si despre monotomie.Se poate vorbi

despre injectivitate, bijectivitate si inversa.

**Teorema impartirii cu rest :**

**Teoremă.** (Teorema împărţirii cu rest a polinoamelor) Fie f,gC[X], g≠0. Atunci există şi sunt unice două polinoame q,rC[X] astel încât f = g · q + r, unde grad(r) < grad(g). Polinomu f se numeste deimpartit, polinomul g, impartitor, polinomul q este catur iar r restul impartirii.

Dacă r = 0, adică f = gq, atunci spunem că polinomul f se divide prin polinomul g (sau că f este multiplu de polinomul g sau că g este un divizor al polinomului f) sau că g divide polinomul f. Dacă f se divide prin g, atunci scriem fg (citim: f se divide prin g) sau g | f (citim:g divide pe f).

**Algoritmul de impartire :**

Pentru a efectua împărţirea polinomului f prin polinomul g ≠ 0 vom utiliza algoritmul care apare în demonstraţia teoremei împărţirii cu rest, ilustrat cu ajutorul unor exemple.

Să se efectueze împărţirea polinomului f=6X5 – 17 X3 – X2 + 3 la polinomul g=3X2 – 6X + 2. Pentru a face aceasta dispunem ca mai jos polinoamele:

6X5 + 0X4 – 17X3 – X2 + 0X + 3 (Deîmpărţitul)|3X2 – 6X +2 (Împărţitorul)

-6X5 + 12X4 – 4X3 |2X3 + 4X2 + X –1 (Catul)

/ 12X4 – 21X3 – X2

- 12X4 + 24X3 – 8X2

/ 3X3 - 9X2 + 0X

- 3X3 + 6X2 – 2X

/ - 3X2 – 2X + 3

3X2 – 6X + 2

/ - 8X + 5 (Restul)

În continuare descriem procedeul utilizat:

**1)** Se ordonează polinoamele f şi g după puterile descrescătoare ale nedeterminatei X.

**2)** Se face împărţirea polinomului de grad mai mare (aici f) la polinomul de grad mai mic.

**3)** Se împarte primul termen al lui f la primul termen al lui g; se obţine astfel primul termen al câtului ( în exemplu avem: 6X5 : 3X2 = 2X3).

**4)**Se înmulţeşte rezultatul astfel obţinut (in exemplu 2X3) cu împărţitorul g şi se scade acest produs din deîmpărţitul f (adică se adună acest produs cu semn schimbat la f). Acest calcul ne dă primul rest al deîmpărţirii (în exemplu, primul rest este polinomul 12X4 – 21X3 – X2 +3).

**5)** Se repetă procedeul luând primul rest ca deîmpărţit.

**6)** Algoritmul se termină când gradul restului este strict mai mic decât gradul împărţitorului (în exemplu, câtul este q = 2X3 + 4X2 + X – 1 şi restul este r = -8X + 5).

**Divizibilitatea unui polinom. Teorema lui Bezout :**

**Teoremă.** Restul împărţirii unui polinom f C[X], f ≠ 0, prin polinomul g = X–aC[X] este egal cu valoarea numerică a polinomului f pentru x = a, adică r = f(a).

**Teorema lui Bézout.** Polinomul fC(X), f ≠ 0, se divide prin g = X–aC(X) dacă şi numai dacă

f (a) = 0.

**Cel mai mare divizor comun a doua polinoame :**



C.m.m.d.c este ultimul rest diferit de 0.

OBS. In cazul determinari prin algoritmul lui Euclid a C.m.m.d.c a doua polinoame la pasul in care se optine restul un numar se deduce ca polinoamele sunt prime intre ele.

**2. Analiza problemei**

Impementara acestei probleme are o necesitate foarte mare deoarece în multe domenii întalnim procesarea unor polinoame. Această aplicaţie ar uşura mult lucru celor care se întalnesc cu aceasta problemă.

Exista multe aplicaţii care pot implementa diferite operaţii cu polinoame, dar aceste programe pot fi folosite doar de utilizatori bine familiarizaţi cu ea, ceea ce nu permite unui începător sa lucreze cu ea. Este de dorit elaborarea unei aplicaţii care ar avea o interfaţă care ar permite sa fie utilizate de către începători.

Această aplicaţie poate fi implementată în mai multe limbaje de programare, dar cred ca java este unul din cele mai bune pentru această aplicaţie deoarece permite elaborarea unei interfeţe pentru utilizator. Java are la dispoziţie mai multe clase care permiterea scrierea unui cod relativ scurt pentru aplicaţii cu interfeţe.

Prima problema întalnită la analiza cerinţei este introducerea polinomului sau polinoamelor de calculat. Cel mai uşor cred ca ar fi mai întâi introducerea gradului polinomului şi apoi introducerea pe rand a fiecărui coieficient a polinomului, începând cu coeficientul lui x la puterea cea mai mare. În cazul cand avem operaţii cu două polinoame introducerea se va face succesiv, mai întâi primu polinom si elementele sale şi dupa al doilea polinom cu elementele sale. Următoarea probleme ar fi alegerea unei operaţii care se va efectua asupra polinoamelor. Aceasta se va alege dintro coloana de butoane care vor fi etichetate corespunzător operaţiei care o vor implimenta.

Toate datele de la intrare vor fi scrise în ferestre de mesaje care vor apărea corespunzător.

Această aplicaţie va efectua următoarele operaţii cu polinoame: testarea eglităţii a două polinoame, adunarea a două polinoame, diferenţa a doua polinoame, derivata unui polinom, înmulţirea a doua polinoame, aflarea rădacinii unui polinom (de grad nu mai mare de 2, deoarece polinoamele cu grad mai mare necesită operaţii mai complicate de calculare a rădăcinilor), afişarea unui polinom în formă canonică, calcularea unui polinom intr-un punct x specificat de către utilizator.

Această aplicaţie ar putea fi folosită in calcularea diferitor modele matematice a situaţiilor reale de zi cu zi. Mai multe formule de calcul din fizică pot fi reduse la calcularea unor polinoame, sau efectuare a diferitor operaţii asupra lor, deasemenea poate fi utila pentru vizualizarea în formă canonică a polinoamelor.

Pentru a implementa acest proiect in programarea orientata pe obiecte, vom avea nevoie de o clasa pentru a reprezenta polinoamele - **Polinom,** una pentru a implementa operatiile - **Operatii** ,una pentru metoda statica main si una pentru interfata - **Interfata**.

**3. Proiectare**

Pentru efectuarea acestei aplicaţii voi avea nevoie de 3 clase principale care vor avea metode specifice. Clasele vor fi următoarele:

* Polinom
* Interfata
* Operatii

Ulterior voi descrie amănunţit fiecare clasa, dar acum le voi descrie in linii mari.

Am inceput de la ideia ca îmi trebuie o clasa care ar crea un obiect de nume polinom, acest polinom va avea două variabile instanţă, gradul si un vector de coeficienţi, va avea constructori care vor iniţia aceste variabile instanţă în dependenţa de situaţie şi va mai conţine diferite metode de operare asupra acestui obiect de Polinom de tipul: set-,get- pentu setarea gradului, setarea coeficienţilor sau unui coeficient anume.

Toate operaţiile care se fac asupra polinoamelor le-am pus intr-o singură clasa cu numele Operatii. Aceasta clasă conţine toate operaţiile care se fac de aplicaţie asupra polinoamelor. Toate metodele sunt declarate staice pentru a putea fi folosite de alte clase fara a declara un obiectb de tip Operatii. Model: Operatii.*nume\_operatie([parametri,…])* nume\_operatie fiind numele unei metode din clasa Operatii împreuna cu parametrii ei daca sunt necesari. Majoritatea operaţiilor returnea un obiect de tip Polinom deasemenea unele returnează boolean in cazul operaţiei de egalitate si un obiect de tip String în cazul aflării rădacinilor, aceasta este cauzată de posibilitatea apariţiei unei sau mai multor rădacini si de diferite tipuri, neajunsul ar fi că nu putem face alte operaţii cu rezultatul de la operaţia aflarea rădăcinii.

Altă clasă pentru implementarea aplicaţiei este Interfaţa, dacă restu claselor fac operaţii care nu pot fi văzute de utilizator, această clasă are ca funcţie legatura dintre programul care execută operaţiile si utilizatorul. Această clasă extinde clasa Jframe, care reprezintă o fereastră cu butoane, fiecare buton este etichetat cu o eticheta care reprezintă operaţia care va fi efectuată. Butoanele în fereastră sunt amplasate într-o coloană cu 8 randuri, respectiv 8 operaţii care pot fi implementate:

Egalitatea(afisează la ecran un mesaj daca 2 polinoame sunt egale sau nu);

Adunare(efectuează operaţia de adunare asupra 2 polinoame şi afisează rezultatul);

Diferenţa(efectuează operaţia de scadere a 2 polinoame şi afisează rezultatul);

Inmulţirea(efectuează operaţia de înmulţire a 2 polinoame şi afisează rezultatul);

Derivata(Derivează un polinom şi afisează rezultatul);

Aflarea rădăcinilor (află rădăcinile polinoamelor de grad 1 şi 2 şi afisează rezultatul, în cazul polinoamelor de grad >3 se afisează un mesaj respectiv);

Afisarea(afişează polinomul în formă canonică);

Rezolvarea polinomului în punctul x(rezolvă polinomul într-un punct x specificat de către utilizator şi afisează rezultatul);

Fereastra care apare la pornirea aplicaţiei are numele Operaţii cu polinoame, restul mesajelor sunt afişate cu ajutorul Joption Message, la fel si datele introduse sunt scrise in ferestre de mesaje.

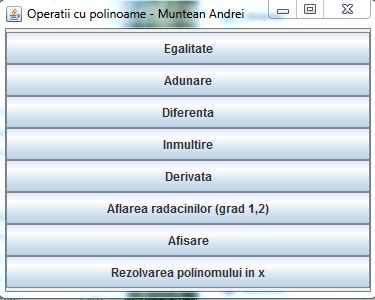
Metoda main care face ca aplicaşia să fie executabila am puso intr-o nouă clasa cu numele Principal, în care creez un obiect nou de tip Interfaţă.

* **Utilizarea**

Proiectul este realizat in NetBeans IDE.

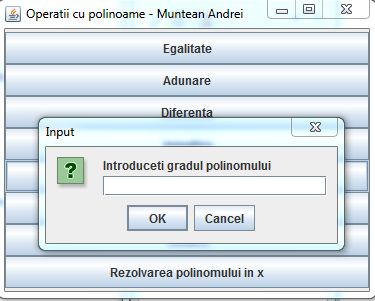
Pentru a utiliza programul utilizatorul trebuie sa respecte urmatorii pasi:

1. Sa porneasca programul Polinom.jar
2. In fereastra principala se va afisa un meniu un utilizatorul poate alege cum vor fi procesate polinoamele.

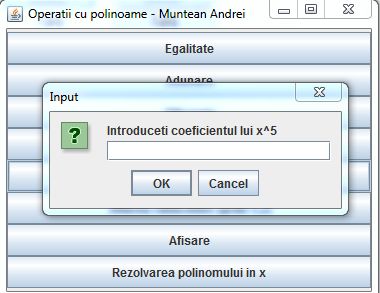
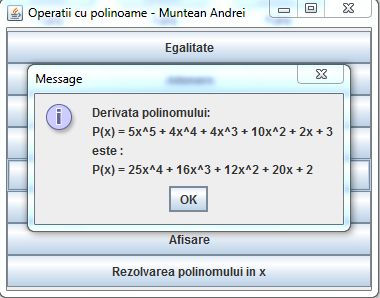


1. Spre exemplu alegem „Derivata” :

Se introduce gradul polinomului -

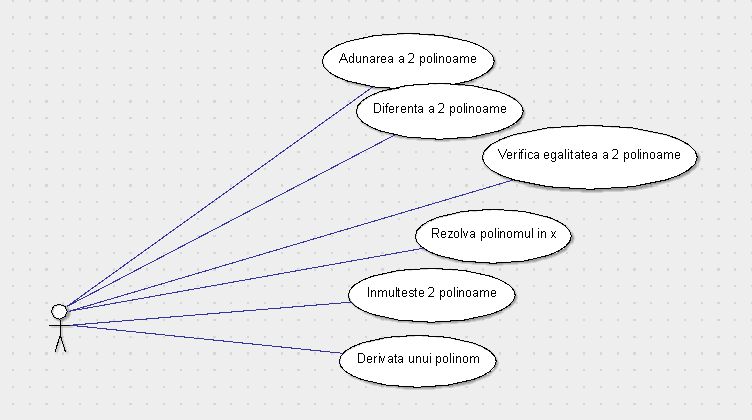


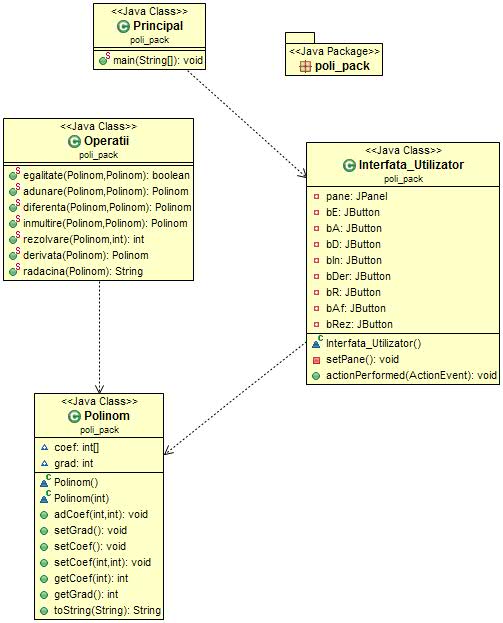
Se introduc coefecientii –

**

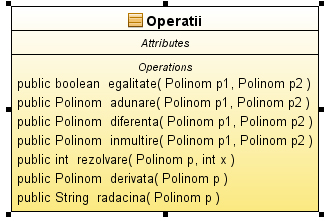
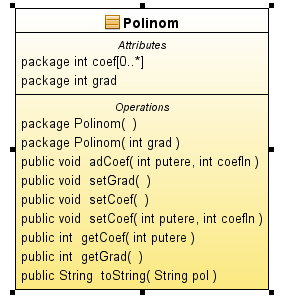
Dupa ce au fost introdusi coefecientii programul va calcula derivata polinomului introdus de utilizator –

**3.b) Proiectare :**

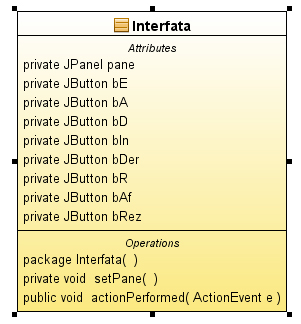
UML\_use-case:

* **Diagrame UMLClasele:**

Clasa **Operatii** se ocupa cu toate operatiile celor doua polinoame. Toate operatiile returneaza acelasi obiect de tip polinom. Operatiile de integrare si diferentiere nu au parametru ci sunt aplicate doar primului polinom.



In clasa **Polinom** avem constructorii **setCoef** si **setGrad,** operatii care transforma forma string a coeficientilor in array de intregi, sau care transforma un obiect polinom intr-u string pentru a fi afisat mai usor.

**Clasa Interfata –** clasa extinde clasa JFrame si implementeaza ActionListener. Acest obiect nou

format reprezinta o fereastra cu 8 butoane, are denumirea de "Operatii cu polinoame", si

fiecare buton este etichetat corespunzator unei operatzii cu polinoame

Constructorul creaza o fereastra de dim specif centarata in mijlocul ecranului

cu denum specif, are si metoda de setare a panelui cu butoane care ulterior e

adaugat la fereastra

**4.Implementarea si Testarea**

Proiectul este realizat NetBeans 7.2.1 win 32.

În continuare voi analiza ficare rând de cod, dar voi trece peste unele parţi deoarece detalierea unelor secvenţe este descrisă in codul sursă în forma de comentariu sau în javadoc care este ataşat împreuna cu proiectul.

* **Clasa Polinom**

Am inceput cu importarea clasei JoptionPane pentru utilizarea ei ulterior in metoda de setare a coeficienţilor şi a gradului polinomului.

Am iniţiat 2 variabile instanţă grad de tipul intreg si coef care este un tabel de valori intregi. Aceste 2 variabile vor fi folosite pentru:

Grad- pentru memorarea gradului polinomului;

Coef- pentru memorarea coeficienţilor polinomului;

Am creat doi constructori unul fară parametri dar cu o metoda în blocul lui care iniţiază variabilele instanţă şi altul cu un parametru care reprezintă gradul polinomului şi are in blocul de cod iniţierea variabilei instanţă coef care reprezină un tabel de lungimea gradului introdus ca parametru. Aceşti 2 constructori sunt folosiţi in diferite cazuri, cel fară parametru este folosit la iniţierea polinoamelor introduse de la tastatură iar celălalt pentru formarea rezultatelor de la operaţii.

În continuare voi descrie în linii generale metodele care sunt implementate în această clasă.

Metoda adCoef o folosesc pentru adunarea unui coeficient specificat prin parametru de intrare, care reprezintă puterea lui x, la un intreg. Această metodă o voi volosi la operaşia de îmulţire unde am nevoie de adunare a coeficienşilor lui x de acelaşi grad.

În această clasă am definit mai multe metode de tipul set-, get- care sunt folosite pentru setarea coeficienţilor şi gradului polinomului sau pentru returnarea unui coeficient sau gradului.

Metodele setCoef() şi setGrad() fără parametri de intrare setează cu ajutorul ferestrelor de mesaje aceşti aprametri, dar aceste metode vor fi descrise ulterior înmpreună cu clasa interfaţă deoarece aparţin mai mult acestei clase.

Metoda toString este folosită pentru transforrmarea obiectului Polinom într-un sir de caracter după nişte regului:

* Se afişează polinomul în forma canonică;
* Pentru coeficienţii egali cu zero sau grad egal cu zero nu se afişează x;
* Puterea este reprezentată de simbolul ^ (ex: 3x^2 = 3x2);

Ex: P(x)= -3x^2 + 2x – 4

În rezultat vom obţine şirul anterior.

* **Clasa Operatii**

Toate metodele din această clasă sunt declarate statice.

Toate metodele au ca parametri de intrare unul sau două Polinoame în dependenţă de operaţia efectuata şi majoritatea returnează tot un polinom ce reprezintă rezultatul operaţiei făcute asupra polinoamelor de la intrare, excepţie facând urmaătoarele metode:

* metoda egalitate returnează un boolean care este true dacă cele 2 polinoame sunt egale şi false daca nu;
* metoda rezolvare returnează un intreg ce reprezintă valoare polinomului în punctul x specificat de către utilizator
* metoda radacini returnează un string care va fi descris ulterior;

Metodele adunare, diferenta, inmultirea, derivata reprezintă efectuarea acestor operaţii rand pe rand pentru fiecare coeficient al polinomului.

Metoda radacini formează un sir de caractere care va fi detaliat împreună cu clasa Interfaţă deoarece ţine de partea de interfaţă grafică.

* **Clasa Interfata**

Această clasă extinde clasa Jframe ţi impelemntează interfaţa ActionListener. Ea formează un obiect de tip fereastră cu nume Operaţii cu polinoame care conţine o coloană cu 8 linii de butoane etichetate.

Variabilele instanţă sunt pane care reprezintă un JPanel la care adăugam butoanele şi cele 8 butoane care reprezintă fiecare operaţie respective etichetei de pe ele. În constructor definim setările ferestrii cum ar fi dimensiune, aliniere şi altele, în constructor se apelează si funcţia setPane() care face mai multe operaţii

Metoda setPane face urmatoarele operaţii:

* inişializează butoanele;
* adauga ascultători si comandă la fiecare buton
* adaugă butoanele la pane în forma de matrice cu o coloană si opt linii;

Deasemenea în clasa Interfaţă este implementată metoda actionPerformed din Interfaţa ActionListener care primeşte ca marametru un obiect de tip ActionEvent. Această metodă este apelată atunci cand se apasă un buton din lista de butoane din fereastra Operaţii cu Polinoame. Această metodă conţine in ea o instrucţiune swich care verifica comanda aparută prin apăsarea butonului respectiv şi executa blocul de instricţiuni care apare intre case si break a comandei apărute.

În continuare voi prezenta cum se execută fiecare operaţie din blocul swich();

Mai intai se iniţializează fiecare polinom cu ajutorul unor ferestre de mesaje şi cu ajutorul constructorului fara parametri din clasa Polinom care apelează la randul lui metoda setCoef care mai intai aplelează metoda setGrad care afişează un mesaj in care introduci valoarea gradului Polinomului, apoi apar n ferestre in care introduci coeficientul elementului respectiv din polinom.

* **Clasa Principal**

Această clasă conţine doar metoda main care apelează constructorul interfeţei prin crearea unui obiect de tip Interfaţă astfel compiland această clasa se va porni aplicaţia.

In continuare va fi prezentat toate mesajele care apar in aplicaţie în diferite cazuri.

**Start aplicaţie**

(apare o fereastră cu butoane în care alegeţi opţiunea dorită de executat)

Egalitate Adunare Diferenţă Inmultire Derivata Aflarea Afisarea Rezolvarea

radacinilor polinomului in x

Doar pentru operaţia

Mesaj informativ Mesaj informativ de rezolvare a

„Setarea polinomului 1” „Setarea polinomului 2 polinomului in

Nota: Se seteaza şi polinomul punctul x

2 doar pentru

Setarea gradului operaţiile:

polinomului respectiv adunare

diferenţă

egalitate

Setarea coeficienţilor inmultire

Nota: Acest mesaj va apărea

pentru fiecare coeficient

al polinomului

* **Testarea**

Testarea este o parte foarte importantă pentru că ne ajută sa ne asiguram că aplicaţia merge bine. Din start vreau sa atenţionez ca aplicaţia nu tratează nici o eraoare care poate apărea la introducerea datelor. Astfel va prezint situaţiile în care aplicaţia nu va funcţiona:

* se introduce altceva decât un întreg pozitiv la gradul polinomului
* se introduce altceva decât un întreg pozitiv sau negativ la coeficienţii polinomului
* nu se introduce nimic
* la introducerea datelor se apasă butonul Cancel în loc de OK
* **Egalitate:**

Intarări 1:

Grad polinom1: 6

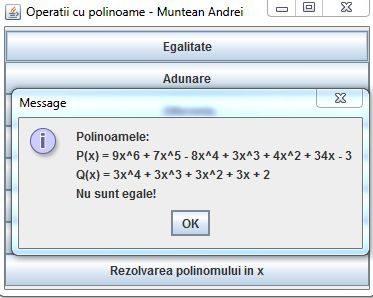
Coeficienţi:9 7 -8 3 4 34 -3

,

Grad polinom2:4

Coeficienţi:3 3 3 3 2

Rezultat:



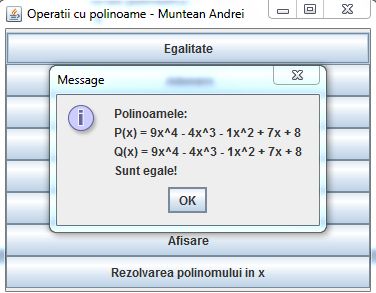
Intarări 1:

Grad polinom1: 4

Coeficienţi:9 -4 -1 7 8

Grad polinom2: 4

Coeficienţi: 9 -4 -1 7 8

Rezultat:

* **Adunare:** Intarări:

Grad polinom1: 5

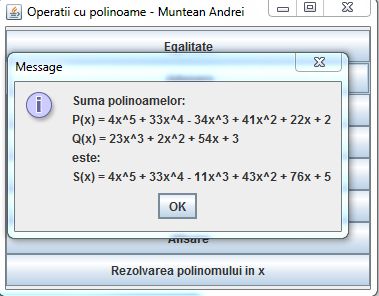
Coeficienţi:4 33 -34 41 22 2

,

Grad polinom2:3

Coeficienţi:23 2 54 3

Rezultat:



* **Diferenta:**Intarări:

Grad polinom1: 5

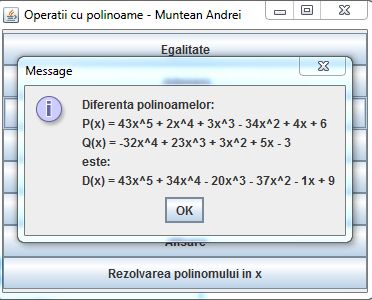
Coeficienţi:43 2 3 -34 4 6

,

Grad polinom2:4

Coeficienţi:-32 23 3 5 -3

Rezultat:



* **Inmultire:**

Intarări:

Grad polinom1: 4

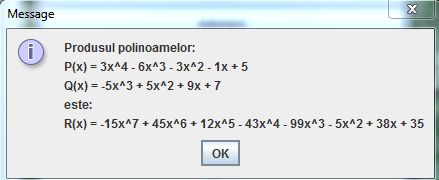
Coeficienţi: 3 -6 -3 -1 5

,

Grad polinom2:3

Coeficienţi: -5 5 9 7

Rezultat:



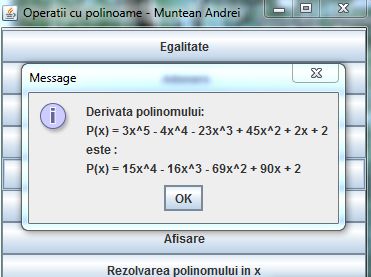
* **Derivata:**

Intarări:

Grad polinom1: 5

Coeficienţi:3 -4 -23 45 2 2

Rezultat



* **Rezolvarea polinomului in x:**

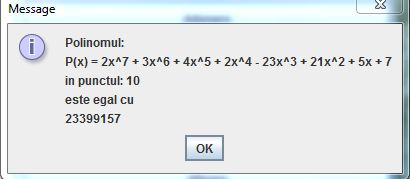
Intarări 1:

Grad polinom1: 7

Coeficienţi:2 3 4 2 -23 21 5 5

X: 10

Rezultat:



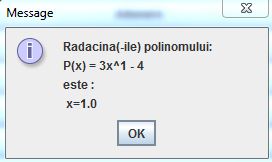
* **Aflarea rădăcinilor:**

Intarări 1:

Grad polinom1: 1

Coeficienţi: 3 -4

Rezultat:

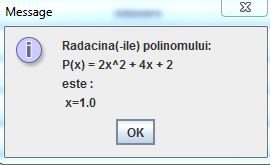


Intarări 2:

Grad polinom1: 2

Coeficienţi: 2 4 2

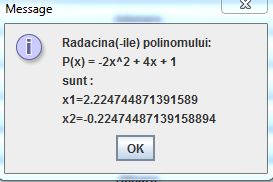
Rezultat:



Intarări 3:

Grad polinom1: 2 Coeficienţi: -2 4 1

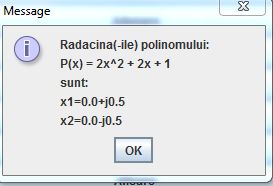
Rezultat:



Intarări 4:

Grad polinom1: 2 ;Coeficienţi:2 2 1

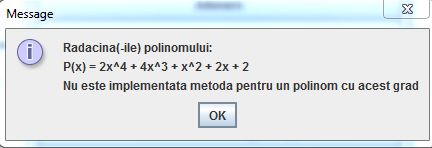
Rezultat:



Intarări 5:

Grad polinom1: 4 Coeficienţi:2 4 1 2 2

Rezultat:



**5.Concluzii**

În final am obţinut o aplicaţie care o poţi rula pe orice masina, deasemenea are clase si metode care pot fi folosite ulterior in alte aplicaţii, cum ar fi clasa polinom si clasa operatii care poate fi folosita la calcule mai complexe a unor polinoame.

Dezvoltări ulterioare.

Prima problemă care ar trebui rezolvată şi dezvoltată este memorarea unui sau mai multor polinoame şi efectuarea mai multor operaţii fără necesitatea de a introduce iar datele polinoamelor. Alta problemă ar fi tratarea erorilor datelor de intrare pentru a uşura lucru cu aplicaţia. De asemenea partea de interfaţă ar trebui îmbunătăţită prin folosirea unei ferestre cu zone de introducere si afisare pentru înlocuirea ferestrelor de mesaje. Ar fi bine de implementat o metoda de aproximare a radăcinilor polinoamelor de grad mai mare şi implementarea unei funcţii de desenarea graficului polinomului întrţun interval definit.