

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

## **Tehnici de programare**

### **Documentatie: TEMA 1**

**Ardeleanu Madalin-Florin**

**An academic : 2018 – 2019**



## 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
5. Cazuri de testare
6. Concluzii, ce s-a invatat din tema, dezvoltari ulterioare



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

## 1. Obiectivul temei

*Enunt:* Propuneti, proiectati si implementati un sistem de procesare a polinoamelor de o singura variabila cu coeficienti intregi.

Aceasta tema are ca obiectiv crearea unui calculator de polinoame. La fel ca si calculatorul de buzunar, calculatorul sistemului de operare pe care il folositi sau orice alt tip de calculator existent, acesta trebuie sa implementeze operatii precum, adunarea, scaderea, inmultirea, impartirea a doua polinoame, precum si derivarea unuia si integrarea lui. La fel ca si orice sistem de calcul si acesta are date de intrare si data de iesire date de rezultatul calculelor.

Imi propun sa implementez un calculator cat mai functional, usor de inteles si util pentru orice tip de utilizator.

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

Principile de baza sunt:

**Abstractizarea** – Este posibilitatea ca un program să separe unele aspecte ale informației pe care o manipulează, adică posibilitatea de a se concentra asupra esențialului. Fiecare obiect în sistem are rolul unui “actor” abstract, care poate executa acțiuni, își poate modifica și comunica starea și poate comunica cu alte obiecte din sistem fără a dezvălui cum au fost implementate acele facilități. Procesele, funcțiile sau metodele pot fi de asemenea abstracte, și în acest caz sunt necesare o varietate de tehnici pentru a extinde abstractizarea:

**Încapsularea** – numită și ascunderea de informații: Asigură faptul că obiectele nu pot schimba starea internă a altor obiecte în mod direct (ci doar prin metode puse la dispoziție de obiectul respectiv); doar metodele proprii ale obiectului pot accesa starea acestuia. Fiecare tip de obiect expune o interfață pentru celelalte obiecte care specifică modul cum acele obiecte pot interacționa cu el. Polimorfismul – Este abilitatea de a procesa obiectele în mod diferit, în funcție de tipul sau de clasa lor. Mai exact, este abilitatea de a redefini metode pentru clasele derivate. De exemplu pentru o clasă Figura putem defini o metodă arie. Dacă Cerc, Dreptunghi, etc. ce vor extinde clasa Figura, acestea pot redefini metoda arie.

**Moștenirea** – Organizează și facilitează polimorfismul și încapsularea, permițând definirea și crearea unor clase specializate plecând de la clase (generale) deja definite - acestea pot împărtăși (și extinde) comportamentul lor, fără a fi nevoie de a-l redefini. Aceasta se face de obicei prin gruparea obiectelor în clase și prin definirea de clase ca extinderi ale unor clase existente. Conceptul de moștenire permite construirea unor clase noi, care păstrează caracteristicile și comportarea, deci datele și funcțiile membru, de la una sau mai multe clase definite anterior, numite clase de bază, fiind posibilă redefinirea sau adăugarea unor date și funcții noi. Se utilizează ideea: ”Anumite obiecte sunt similare, dar în același timp



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

diferite”. O clasă moștenitoare a uneia sau mai multor clase de bază se numește clasă derivată. Esența moștenirii constă în posibilitatea refolosirii lucrurilor care funcționează.

Pentru realizarea acestei teme avem nevoie de cumulara cunostintelor anterioare de java dar si de cunostiinte din matematica.

Calculatorul care trebuie implementat trebuie sa aiba coeficienti intregi. Aceasta constrangere ne poate pune problema la operatiile de impartire si integrare. La impartire, pentru calcularea catului avem nevoie sa impartim doi cate doi intregi care de cele mai multe ori ne vor da un numar real. La integrare coeficientul va fi dat de vechiul coeficient care e un numar intreg, si vechea putere care e tot un intreg. La fel ca la impartire este probabil sa obtinem un numar real pentru noul coeficient. Din aceste cauze la cele doua operatii de mai sus mentionate putem avea erori deoarece numarul real rezultat va fi “salvat” ca si partea lui intreaga. Pentru celelalte operatii constrangerea nu ne pune nici un fel de problema.

Pentru a se putea implementa problema trebuie sa vedem de ce date avem nevoie, cum vrem sa primim datele de intrare, cum salvam polinoamele, cum le creem o structura in interiorul softului cu care putem sa le manevram usor pentru operatiile cerute, cum efectuam operatiile pentru a utiliza functii cat mai usoare si mai putin spatiu pentru stocare. Totodata trebuie sa cream o interfata cat mai practica si usor de folosit.

Pentru a efectua oricare din operatiile dorite, avem nevoie de date precum coeficientul si gradul fiecarui monom din polinom. Pentru aceasta am ales sa folosim introducerea polinoamelor ca si stringuri de tipul (“ $2x^3-5x^1-3x^0$ ”). Am ales aceasta metoda pentru ca utilizatorul vede exact coeficientul si gradul pentru fiecare monom si pentru ca e metoda clasica de reprezentare a oricarui polinom. Gradul polinomului este calculat imediat dupa aceea, deoarece este necesar pentru alocarea acestuia.

Pentru rezolvarea acestei probleme trebuie sa luam in considerare procesarea polinoamelor(parsarea stringului), modelarea fiecarei operatii si afisarea rezultatului dorit.



### 3. Proiectare (diagrame UML, structuri de date, proiectare clase, interfețe, relații, pachete, algoritmi, interfața utilizator)

În realizarea problemei am decis că am nevoie de 7 clase: Polinom, Monom, MonomIntreg, MonomReal, Test, IllegalArgumentException și JUnit.

În clasa Polinom folosesc o listă de monoame care sunt alcătuite dintr-un coeficient și o putere. Folosesc un constructor cu un parametru (int grad), pentru că să aloc spațiu maxim necesar fiecărui polinom. În această clasă am metode precum adaugareMonom(Monom m) ce adaugă monomul m în polinom pe poziția puterii stocate în monom, afisPolinomIntreg() ce afișează polinomul rezultat ce are coeficienții întregi, afisPolinomReal() ce afișează polinomul rezultat ce are coeficienții reali, invers(int nr) ce ne ajută pentru calcularea unor puteri din stringul pe care îl introducem: de exemplu dacă introduce  $x^{20}$ , trebuie să salvăm puterea ca vin 20, nu doar prima cifră, crearePolinom(String polinom, int grad), ce primește polinomul sub formă de string și gradul maxim al acestuia și memorează în memorie un polinom format din monoame, adunare(Polinom a, int grad) ce adună 2 polinoame, pe cel ce apelează și polinomul a, gradul alocat pentru polinomul rezultat trebuie să fie egal cu maximumul dintre gradele celor 2 polinoame, scadere(Polinom a, int grad) ce scade 2 polinoame, pe cel ce apelează și polinomul a, gradul alocat pentru polinomul rezultat trebuie să fie egal cu maximumul dintre gradele celor 2 polinoame, coefMaxim(int putere) returnează coeficientul corespunzător puterii putere sub formă unui double, inmultirePolinom(Polinom a, int gradMaxProdus) ce returnează un polinom ce este calculat ca produs dintre polinomul ce apelează funcția și polinomul a, gradul polinomului rezultat trebuie să fie egal cu gradul primului polinom + gradul celui de-al doilea polinom + 1, derivarePolinom(int grad) ce returnează polinomul derivat corespunzător polinomului ce apelează funcția, derivarea are loc astfel: noul coeficient este format din coeficientul respectiv \* puterea, iar puterea scade cu o unitate, impartireaPolinom(Polinom a, gradPolinom1, gradPolinom2) nu returnează nimic, în interiorul funcției sunt calculate polinoamele corespunzătoare catului și restului urmând să fie afișate, se dau ca parametrii și gradele corespunzătoare fiecărui polinom pentru o bună alocare a spațiului pentru monoame și nu în cele din urmă avem metoda integrarePolinom(grad+1) ce returnează polinomul integrat corespunzător polinomului ce apelează metoda, se primește ca parametru un grad folosit pentru alocare, gradul trebuie să crească cu o unitate la operația de integrare.

În Clasa Test ne creăm interfața, și transmitem fiecare cerință a utilizatorului către soft. Utilizatorului i se prezintă o interfață ușor de utilizat și în care își poate implementa operațiile dorite. Interfața este formată din 5 TextField-uri, 6 butoane pentru fiecare operație, și câteva JLabel-uri care îl ajută să introducă date. Rezultatul operațiilor este afișat în un JLabel.



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

Aproape la fiecare operatie am ales sa salvam rezultatul in un nou polinom rezultat pentru a mai putea efectua si alte operatii cu polinoamele de intrare.

Algoritmii folositi la implementarea operatiilor:

a) Pentru operatia de adunare se alocă spațiu Polinomului rezultat, în funcție de gradul polinomului de intrare mai mare. Pe pozițiile polinomului rezultat până la gradul polinomului mai mic efectuăm suma între coeficienții de același grad, iar pe restul pozițiilor copiem în rezultat coeficienții polinomului care nu a fost parcurs.

b) Pentru operatia de scadere se alocă spațiu Polinomului rezultat, în funcție de gradul polinomului de intrare mai mare. Pe pozițiile polinomului rezultat până la gradul polinomului mai mic efectuăm diferența între coeficienții de același grad, iar pe restul pozițiilor copiem în rezultat coeficienții polinomului care nu a fost parcurs.

c) Pentru operatia de înmulțire se alocă spațiu Polinomului rezultat, grad dat de suma gradelor polinoamelor de intrare. Fiecare coeficient este dat de produsul coeficienților de același grad adunat la vechiul coeficient de pe aceeași poziție.

d) Pentru operatia de împărțire utilizăm un polinom auxiliar. Operatia de împărțire se efectuează în x pași unde  $x = \text{gradA} - \text{gradB}$ . În polinomul rest salvăm polinomul rezultat după fiecare scădere din deîmpărțit a polinomul împărțitor cu polinomul cât.

e) La operatie de derivare mutăm toți coeficienții cu o poziție în urmă și îi înmulțim cu vechea poziție(grad).

f) La operatie de integrare mutăm toți coeficienții cu o poziție înainte de la gradul cel mai mare la cel mai mic și îi împărțim la noua poziție(noul grad).

În clasa JUnit ne testăm programul dacă funcționează prin folosirea unor asertiuni.

Alte clase care sunt utilizate în rezolvarea acestei probleme sunt: JButton: care este o punere în aplicare a în cazul în care este apăsat buton. JLabel: cu această clasă aveți posibilitatea să afișați textul sub formă neselectabilă și imagini JTextField: este o componentă ușor de folosit care permite editarea unei singure linii de text. JPanel: reprezintă fereastra principală în care se adaugă celelalte elemente precum JButton, JLabel, JTextField.

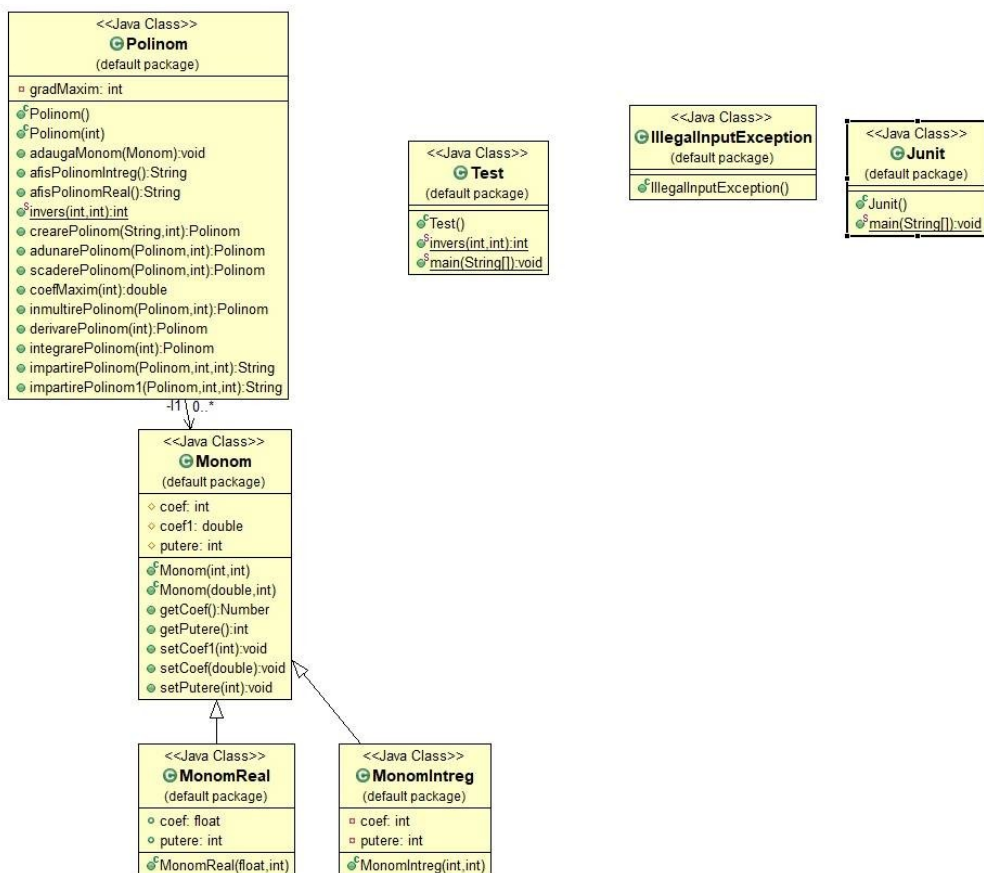


**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

**Digrama UML:**





## 4.Implementare

### Metodele din Clasa Operatii:

#### - adunarePolinom

⌚ public Polinom adunarePolinom(Polinom a, int gradMaxim)

Polinomul rezultat va avea gradul polinomului mai mare. Pana la gradul polinomului mai mic facem adunarea. Dupa ce am trecut de gradului mai mic in polinomul rezultat punem pe urmatoarele grade coeficientii corespunzatori polinomului de grad mai mare.

#### - scaderePolinom

⌚ public Polinom scaderePolinom(Polinom a, int gradMaxim)

#### - afisPolinomIntreg() sau -afisPolinomReal()

⌚ public void afisPolinomIntreg() sau public void afisPolinomReal()

Functia de afisare a polinomului sub forma -coef-X<sup>-exp</sup>. In cazul in care toti coeficientii rezultatului sunt 0 nu afisam nimic(corespunzator valorii 0).

#### - derivarePolinom

⌚ public Polinom deriveazaPolinom(grad)

La operatie de derivare mutam toti coeficientii cu o pozitie in urma si ii inmultim cu vechea pozitie(grad).

#### - impartirePolinom

⌚ public Polinom impartirePolinom(Polinom a, gradPolinom1,gradPolinom2)

Pentru operatia de impartire utilizam trei polinoame auxiliare.

In 2 dintre ele se vor retine catul si restul ce vor fi afisate la final.

Operatia de impartire se efectueaza in x pasii unde x= gradA-gradB. In polinomul rest salvam polinomul deimpartit. La fiecare pas in polinomul rezultat punem gradul catului si coeficientul dat de





**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

catul coeficientului rest pe coeficientul de grad maxim al impartitorului. Polinomul rezultat va fi salvat in un nou polinom cat care va fi inmultit cu impartitorul si adunat la rest. Astfel obtinem noul rest si continuam impartirea. Restul impartiri il calculam ca fiind impartitorul inmultit cu catul, adunat la polinomul deimpartit.

**- inmultirePolinom**

🕒 public Polinom inmultirePolinom(Polinom a, gradProduxMax)

Pe pozitia  $i+j$  care e gradul lui A + gradul lui B punem rezultaul inmultirii celor 2 coeficienti + vechiul rezultat de pe acea pozitie.

**-integrarePolinom**

🕒 public Polinom integreaza(Polinom a)

Mutam toti coeficientii pe un grad mai mare si noi coeficienti vor fi catul lor cu gradul nou.

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

## 5.Cazuri de testare

În continuare vom prezenta câteva cazuri pentru a ne asigura că softul funcționează corect.

Pentru operația de **adunare** cu datele următoare:

$$\text{pol1} = 3x^3 - 5x^2 - 7x^1 + 12x^0$$

$$\text{pol2} = 2x^2 - 2x^1 + 4x^0$$

 <b>Calculator polinoame</b> <span style="float: right;">— □ ×</span>	
<b>Adunare</b>	<b>Scadere</b>
<b>Inmultire</b>	<b>Impartire</b>
<b>Derivare</b>	<b>Integrare</b>
Polinom 1:	$3x^3 - 5x^2 - 7x^1 + 12x^0$
Polinom 2:	$2x^2 - 2x^1 + 4x^0$
Rezultat:	$+3x^3 - 3x^2 - 9x^1 + 16x^0$
Catul:	Afisare rezultat operatie
Restul:	Afisare rezultat operatie


**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

 Pentru operația de **scadere** cu datele următoare:

 $\text{pol2} = 7x^5 + 3x^3 - 4x^2 - 7x^1 + 2x^0$ 
 $\text{pol2} = 2x^3 - 8x^1 + 1x^0$ 

<div>            Calculator polinoame           <span>— □ ×</span> </div>	
Adunare	Scadere
Inmultire	Impartire
Derivare	Integrare
Polinom 1:	$7x^5 + 3x^3 - 4x^2 - 7x^1 + 2x^0$
Polinom 2:	$2x^3 - 8x^1 + 1x^0$
Rezultat:	$+7x^5 + 1x^3 - 4x^2 + 1x^1 + 1x^0$
Catul:	Afisare rezultat operatie
Restul:	Afisare rezultat operatie

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

Pentru operația de **înmulțire** cu datele următoare:

$$\text{pol1} = 7x^1 + 2x^0$$

$$\text{pol2} = 4x^4 - 2x^3 - 8x^1 + 1x^0$$

<div> <span>Calculator polinoame</span> <span>— □ ×</span> </div>	
Adunare	Scadere
Inmulțire	Impartire
Derivare	Integrare
Polinom 1:	$7x^1 + 2x^0$
Polinom 2:	$4x^4 - 2x^3 - 8x^1 + 1x^0$
Rezultat:	$+28x^5 - 6x^4 - 4x^3 - 56x^2 - 9x^1 + 2x^0$
Catul:	Afisare rezultat operatie
Restul:	Afisare rezultat operatie

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

Pentru operația de **împartire** cu datele următoare:

pol1=  $1x^3+1x^0$

pol2=  $1x^1+1x^0$

Calculator polinoame	
Adunare	Scadere
Inmultire	Impartire
Derivare	Integrare
Polinom 1:	$1x^3+1x^0$
Polinom 2:	$1x^1-1x^0$
Rezultat:	Afisare rezultat operatie
Catul:	$+1x^2+1x^1+1x^0$
Restul:	$+2x^0$


**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

Pentru operația de **derivare** cu datele următoare:

pol1=  $6x^4-3x^3+9x^2-1x^1+1x^0$

Calculator polinoame	
Adunare	Scadere
Inmultire	Impartire
Derivare	Integrare
Polinom 1:	$6x^4-3x^3+9x^2-1x^1+1x^0$
Polinom 2:	Introdu al doilea polinom
Rezultat:	$+24x^3-9x^2+18x^1-1x^0$
Catul:	Afisare rezultat operatie
Restul:	Afisare rezultat operatie

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

Pentru operația de **integrare** cu datele următoare:

$$pol1 = 24x^{11} - 27x^8 + 7x^6 - 8x^3 + 1x^0$$

Adunare	Scadere
Inmultire	Impartire
Derivare	Integrare
Polinom 1:	$24x^{11} - 27x^8 + 7x^6 - 8x^3 + 1x^0$
Polinom 2:	Introdu al doilea polinom
Rezultat:	$+2x^{12} - 3x^9 + 1x^7 - 2x^4 + 1x^1 + C$
Catul:	Afisare rezultat operatie
Restul:	Afisare rezultat operatie

**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

Pentru introducerea de polinoame **invalide** :

Calculator polinoame	
Adunare	Scadere
Inmultire	Impartire
Derivare	Integrare
Polinom 1:	$x^42$
Polinom 2:	Introdu al doilea polinom
Rezultat:	Ai introdus polinomul gresit!
Catul:	Afisare rezultat operatie
Restul:	Afisare rezultat operatie





**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

Pentru incercare impartirii unui polinom **de grad mai mic** la un polinom de grad mai mare:


 Calculator polinoame
 —
□
×

Adunare	Scadere
Inmultire	Impartire
Derivare	Integrare
Polinom 1:	$2x^1$
Polinom 2:	$3x^4+2x^2$
Rezultat:	Nu se pot impartii!
Catul:	Afisare rezultat operatie
Restul:	Afisare rezultat operatie



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE**

## **6. Concluzii, ce s-a invatat din tema, dezvoltari ulterioare**

Prin aceasta tema am reusit sa reiau conceptele programarii orientate pe obiect. Am invatat noi metode de a implementa o interfata mai usor.

Ca si dezvoltari ulterioare putem aminti:

- a) Stocarea polinoamelor ca cu coeficienti reali si aplicarea operatiilor pe acestea.
- b) Se mai pot adauga grafice. Pe grafice se vor putea reprezenta polinoamele de intrare dar si polinoamele rezultat.
- c) Se mai pot adauga noi operatii precum gasirea radacinilor unui polinom.
- d) Se poate schimba interfata

## **7. Bibliografie**