**DOCUMENTATIE TEMA 1**

**CALCULATOR POLINOAME**

**Brincoveanu Vasile Vlad**

**30223**

**Profesor Laborator Assist:Antal Marcel**

Contents

[1. Cerinte Functionale 3](#_Toc476131445)

[2. Obiective 3](#_Toc476131446)

[2.1. Obiectiv Principal: 3](#_Toc476131447)

[2.2. Obective Secundare: 3](#_Toc476131448)

[3. Analiza Problemei 3](#_Toc476131449)

[4. Proiectare 3](#_Toc476131450)

[4.1. Structuri de date 3](#_Toc476131451)

[4.2. Diagrama de clase 3](#_Toc476131452)

[4.3. Algoritmi 3](#_Toc476131453)

[5. Implementare 4](#_Toc476131454)

[6. Testare 4](#_Toc476131455)

[7. Concluzii si Dezvoltari Ulterioare 4](#_Toc476131456)

[8. Bibliografie 4](#_Toc476131457)

# Cerinte Functionale

Dezvoltati un calculator de polinoame care sa perimita urmatoarele operatii:

* Citirea unui polinom de la tastatura sub forma 3X^2+2X^3-X+2
* Adunare, Scadere, Inmultire, Impartire, Derivare, Integrare

# Obiective

## Obiectiv Principal:

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

## Obective Secundare:

|  |  |  |
| --- | --- | --- |
| **Obiectiv Secundar** | **Descriere** | **Capitol** |
| Dezvoltarea de use case-uri si scenarii |  | 3 |
| Alegerea structurilor de date |  | 4 |
| Impartirea pe clase |  | 4 |
| Dezvoltarea algoritmilor |  | 4 |
| Implementarea solutiei |  | 5 |
| Testare |  | 6 |

# Analiza Problemei

Un polinom, se defineste ca o functie ce contine una sau mai multe variabile si constante, legate prin operatii de adunare, scadere, inmultire si ridcare la putere constanta pozitiva intreaga.

Polinoamele sunt construite din termeni numiți monoame, care sunt alcătuite dintr-o constantă (numită coeficient) înmulțită cu una sau mai multe variabile. Fiecare variabilă poate avea un exponent constant întreg pozitiv. Exponentul unei variabile dintr-un monom este egal cu gradul acelei variabile în acel monom. Un monom fără variabile se numește monom constant, sau doar constantă. Gradul unui termen constant este 0. Coeficientul unui monom poate fi orice număr, inclusiv fracții, numere iraționale sau negative.

Suma, scaderea, inmultirea, impartirea a două polinoame este un polinom.

Derivata si primitva unui polinom este un polinom..

Problema se refera la polinoame cu o singura variabila si cu coeficienti intregi(dubli pentru imparitire,integrare).

Această forma este considerată forma generală a polinoamelor de o singură variabilă.

C:\Users\Vlad\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Untitled.png

Vom transforma toate informatiile teoretice de mai sus si vom abstractiza clasele de obiecte necesare.

Acestea vor fi:

* Monom
* Polinom
* Operatii
* GUI

Programul a fost conceput, folosind concepte din programarea orientate obiect, aplicate prin limbajul JAVA, folosit in mediul de dezvoltare ECLIPSE. Utilizatorul poate fi un mathematician, fizician, un om de stiinta , dar si un om de rand, deoarece interfata grafica este instinctual facuta pentru oricine.Ea este implementata folosind libraria SWING.

**Scenarii si user case-uri**

Programul prevede urmatorii actori: utilizatorul, care introduce polinoamele, dupa un anumit format dat, alege operatia si obtine rezultatul pe ecran in acelasi format;controlerul,care citeste polinoamele scrise de utilizatorul,si le paseaza unui alt actor,modelul(clasa Operatii) care parseaza string-ul dat de utilizator, instanteaza clasele polinom,si ,prin ajutorul controlerului aplica operatia corecta si o returneaza controlerului,daca sunt erori le va pasa controlerului sa se ocupe de ele. Cel din urma, se asigura ca daca a primit erori de la model sa le afiseze utilizatorului, sa elibereze ecranul, iar daca a primit rezultatul operatiei,ales de utilizator, corect si fara erori, atunci sa il afiseze pe ecran in acelasi format dat de program.

Interfata este de asemenea importanta, dar in aceste randuri am “integrat-o controlerului” astfel explicatiile despre mersul programului sunt mult mai clare.

*Scenariul 1:Impartirea*

1. Utilizatorul introduce in casuta de scris primul polinom in formatul dat, apoi apasa butonul adauga.
2. Controlerul primeste string-ul si il va comunica cu modelul
3. Modelul, daca nu va depista anumite erori, va instantia polinomul respectiv(il va metine in memoria sa), si il va returna controlerului,
4. Acesta il va afisa pe ecran.
5. Utilizatorul poate vedea polinomul si va introduce in aceeasi casuta urmatorul polinom,apoi apasa butonul adauga.
6. Se vor repeta pasii 2-4.
7. Utilizatorul va selecta operatia dorita(impartirea) si va apasa butonul start.
8. Controlerul depisteaza operatia selectata, si comunica cu modelul sa aplice acea operatie pe cele doua polinoame care le are in memorie.
9. Modelul va aplica algoritmul pe cele doua polinoame si va returna rezultatul controlerului.
10. El va afisa rezultatul pe ecran, impreuna cu operatia selectata si eventualele erori primite de la model.

*Scenariul 2:Derivarea*

1. Utilizatorul introduce in casuta de scris polinom-ul in formatul dat, apoi apasa butonul adauga.
2. Controlerul primeste string-ul si il va comunica cu modelul.
3. Modelul, daca nu va depista anumite erori, va instantia polinomul respectiv(il va metine in memoria sa), si il va returna controlerului.
4. Acesta il va afisa pe ecran.
5. Utilizatorul va selecta operatia respectiva(derivare) si va apasa buntoul start.
6. Controlerul depisteaza operatia selectata, si comunica cu modelul sa aplice acea operatie pe polinoamul care il are in memorie.
7. Modelul va aplica algoritmul pe acel polinom si va returna rezultatul controlerului.
8. El va afisa rezultatul pe ecran, impreuna cu operatia selectata si eventualele erori primite de la model.

*Scenariul 3:introducere gresita a polinomului(sau neintroducerea niciunui polinom)*

1. Utilizatorul va apasa butonul adauga fara a scrie un polinom in casuta de scris.
2. Controlerul va depista acest lucru si va afisa o eroare utilizatorului.
3. Utilizatorul va apasa butonul start fara sa adauge un polinom(daca operatia nu este selectata, se va efectua implicit adunarea).
4. Controlerul va afisa o eroare utilizatorului.
5. Dupa efectuarea unei operatii, utilizatorul va apasa butonul start dinou, fara a adauga un nou polinom.
6. Controlerul va afisa o eroare utilizatorului

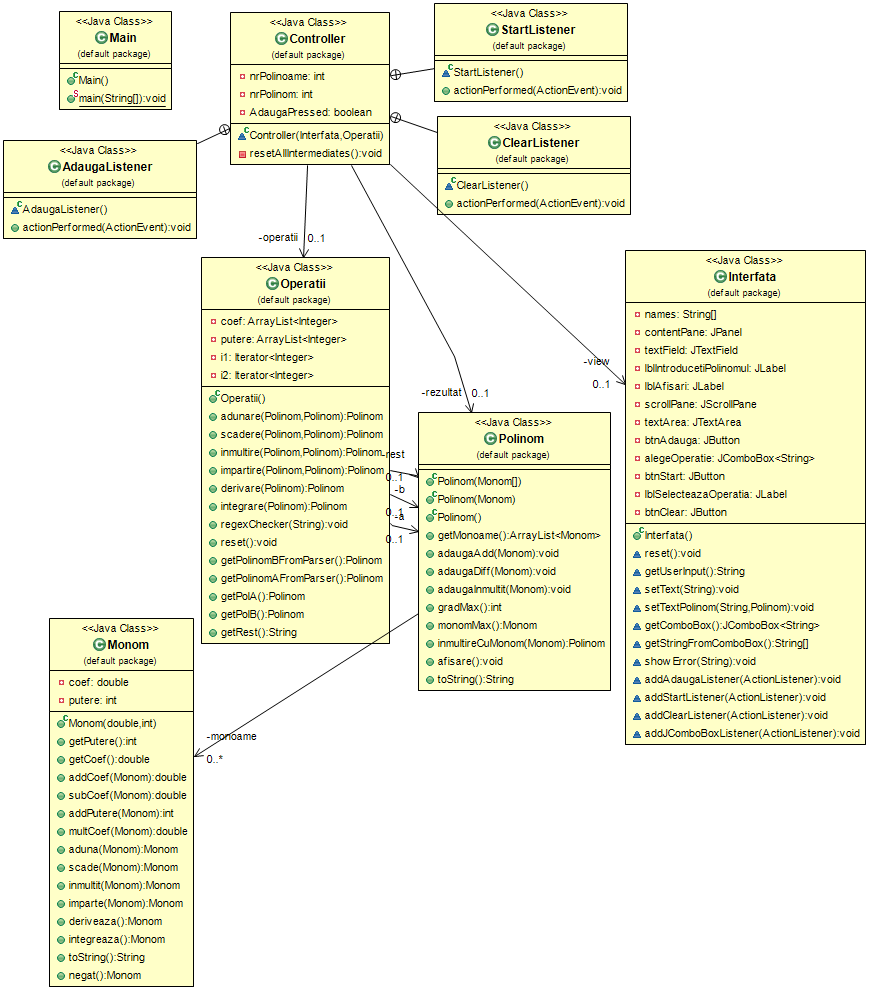
# Proiectare

## Structuri de date

Sunt folosite date simple precum integer, double, string etc pentru memorarea puterii, a coeficientului, respectiv string-ul introdus de utilizator.

Pentru a stoca intr-un polinom fiecare monom am folosit un ArrayList, deoarece este alocat dinamic si este simplu de utilizat si inteles(se putea lucra mai eficient folosind HashTable pentru a stoca monoame)

## Diagrama de clase



## Algoritmi

In acest program putem vedea algoritmi de adunare, scadere,impartire,inmultire a monoamelor.

Cel mai evident algoritm, cel de impartire, este efectuat de metoda impartire() din clasa Operatii.El primeste doua polinoame la intrare(deimpartitul si imparitorul) si va returna un polinom(catul),cat si restul(deimpartitul dupa efectuarea algorimtului).

1. Se va imparti monomul de grad maxim din impartitor la monomul de grad maxim din deimpartit.
2. Monomul rezultat se va adauga la cat. Acelasi monom rezultat se va inmulti la impartitor si se va salva intr-un polinom auxiliar.
3. Din deimpartit vom scadea polinomul auxiliar.
4. Vom repeta pasii 1-3 cat timp gradul maxim din deimpartit va fi mai mare strict decat gradul maxim din impartitor si coeficientul monomului maxim din deimpartit / coeficientul monomului maxim din impartitor sa fie diferit de 0.

Un alt algoritm implementat ar fi, adaugarea fiecarui monom intr-un arraylist care formeaza un polinom. De aici putem explica adunarea a doua polinoame, ori scaderea, ori inmultirea, care sunt efectuate in anumite metode din clasa Operatii. De exemplu, la adunare:

1. Se parcurge primul polinom si fiecare monom din el este adaugat(se va verifica gradul,daca se adauga un monom de acelasi grad cu alt monom din lista, se va scoate din lista primul si se va adauga monomul cu coeficienti adunati si de aceeasi putere) intr-un nou monom;
2. Apoi, se va parcurge fiecare monom din polinomul doi si se va adauga la noul polinom urmand acelasi principiu explicat mai in sus, in paranteze.

Acelasi algoritm il efectueaza si scaderea, doar ca coeficientii monoamelor de acelasi grad se scad.

# 

# Implementare

**Clasa Monom**

Clasa monom este clasa de baza a acestui program. Precum si numele, clasa va instantia obiect de tipul monom, care va avea un coeficient si o putere,si care va putea sa adune sau sa scada un alt monom de acelasi grad si sa ,imparta ori,sa inmulteasca un alt monom la ea, sau sa se deriveze,integreze ori nege. Pe langa acestea mai are getteri pentru putere si coeficienti, dar NU si setter, astfel creand o mai mare integrare si siguranta, neputand fi modificata oricum din exterior !!.

Se **defineste** astfel:

public class **Monom**

extends java.lang.Object

Are urmatoarele **campuri(variabile instanta)**:

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private double | [**coef**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#coef) |
| private int | [**putere**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#putere) |

(coeficientul este double, pentru o mai buna precizie la impartirea a doua polinoame cat si la integrarea unui polinom).

**Constructorul** este de forma:

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#Monom-double-int-)(double coef, int putere) |

(constructorul va instantia cate un obiect din aceasta clasa cu un anumit coeficient si o anumita putere)

Contine urmatoarele **metode**:

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| double | [**addCoef**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#addCoef-Monom-)([**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) b)  returneaza suma coeficientilor acestui monom cu monomul dat prin parametru |
| int | [**addPutere**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#addPutere-Monom-)([**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) b)  returneaza suma puterilor acestui monom cu monomul dat prin parametru |
| [**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) | [**aduna**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#aduna-Monom-)([**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) b)  returneaza un nou monom,care are ca si coeficient, suma coeficientilor acestui monom  cu monomul dat si aceeasi putere,doar daca suma coeficientilor e diferita de 0,altfel  returneaza null |
| [**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) | [**deriveaza**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#deriveaza--)()  returneaza un nou monom, care are ca si coeficient, produsul dintre coeficientul si  puterea acestui monom, si ca si putere, puterea monomului-1 |
| double | [**getCoef**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#getCoef--)()  returneaza variabila instanta coeficient a unui obiect instantiat |
| int | [**getPutere**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#getPutere--)()  returneaza variabila instanta putere a unui obiect instantiat |
| [**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) | [**imparte**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#imparte-Monom-)([**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) b)  returneaza un nou monom,care are ca si coeficient, impartirea coeficientilor acestui  monom cu monomul dat si ca si putere,diferenta coeficientilor dintre cele doua monoame |
| [**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) | [**inmultit**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#inmultit-Monom-)([**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) b)  returneaza un nou monom, care are ca si coeficient, produsul coeficientilor acestui  monom cu monomul dat si ca si putere, suma coeficientilor dintre cele doua monoae |
| [**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) | [**integreaza**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#integreaza--)()  returneaza un nou monom, care are ca si coeficient, impartirea dintre coeficientul  si puterea acestui monom, si ca si putere, puterea monomului+1 |
| double | [**multCoef**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#multCoef-Monom-)([**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) b)  returneaza produsul coeficientilor acestui monom cu monomul dat prin parametru |
| [**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) | [**negat**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#negat--)()  returneaza un nou monom care va avea coeficientul acestui monom negat si aceeasi putere |
| [**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) | [**scade**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#scade-Monom-)([**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) b)  returneaza un nou monom,care are ca si coeficient, diferenta coeficientilor acestui  monom cu monomul dat si aceeasi putere,doar daca diferenta coeficientilor e diferita  de 0,altfel returneaza null |
| double | [**subCoef**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#subCoef-Monom-)([**Monom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html) b)  returneaza diferenta coeficientilor acestui monom cu monomul dat prin parametru |
| java.lang.String | [**toString**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Monom.html#toString--)()  S-a suprascris metoda toString() din clasa Object pentru o mai buna formatare la  aparitia pe ecran,astfel s-au luat in considerare urmatoarele lucruri: -daca coeficientul  este 0 se va returna blank; -daca coeficientul este 1 se vor returna caracterele 'x'.'^',  urmate de valoarea puterii; -daca puterea este 0 se va returna o aproximare cu 2  zecimale a coeficientului; -daca puterea este 1 se va returna o aproximare cu 2 zecimale  a coeficientului + caracterul 'x'; |

**Clasa Polinom**

Clasa Polinom stocheaza intr-un ArrayList, monoamele care vor forma un polinom. Ea contine metode de adaugare pentru diferite operatii, care se bazeaza pe stergerea unui monom de acelasi grad si adaugarea, impreuna cu celalat monom, a unui nou monom format dintr-o oepratie pe cele 2 monoame. Avem si o suprascriere a metodei toString() sporind eficienta si frumusete afisarii pe ecran a polinomului.

Se **defineste** astfel:

* public class **Polinom**

extends java.lang.Object

Are urmatoarele **campuri(variabile instanta)**:

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private java.util.ArrayList<Monom> | [**monoame**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#monoame)  strucutra de date in care se pastreaza monoamele |

**Constructorul** este de forma:

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Polinom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#Polinom--)()  contructorul deafult |
| [**Polinom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#Polinom-Monom-)(Monom a)  contructorul care primeste ca si parametru un monom |
| [**Polinom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#Polinom-Monom:A-)(Monom[] a)  contructorul care primeste ca si parametru un array de monoame |

Contine urmatoarele **metode**:

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| void | [**adaugaAdd**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#adaugaAdd-Monom-)(Monom b)  se va adauga prin adunare monomul primit ca si parametru acestui polinom |
| void | [**adaugaDiff**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#adaugaDiff-Monom-)(Monom b)  se va adauga prin scadere monomul primit ca si parametru acestui polinom |
| void | [**adaugaInmultit**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#adaugaInmultit-Monom-)(Monom b)  se va adauga prin inmultire monomul primit ca si parametru acestui polinom |
| void | [**afisare**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#afisare--)()  afisare polinom(doar pentru debugg) |
| java.util.ArrayList<Monom> | [**getMonoame**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#getMonoame--)()  getter pentru lsita de monoame |
| int | [**gradMax**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#gradMax--)()  returneaza valoarea gradului maxim din acest polinom |
| [**Polinom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html) | [**inmultireCuMonom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#inmultireCuMonom-Monom-)(Monom b)  returneaza acest polinomul inmultit cu monomul dat prin parametru |
| Monom | [**monomMax**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#monomMax--)()  returneaza monomul cu grad maxim din acest polinom |
| java.lang.String | [**toString**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Polinom.html#toString--)()  suprascriere toString() din clasa Object,astfel: -se returneaza 0,daca monomul e null -se concateneaza toate monoamele.toString() -daca la primul monom avem caracterul'+',el se sterge |

**Clasa Operatii**

Daca pana acum am prezentat clase care stocau informatia primita, acum urmeaza clasa Operatii care este creierul din spatele programului. Ea face toate operatiile pe polinoamele date, face parsarea din string in polinom, face reinitializari etc. Clasa se mai poate numi si model, in metoda MVC.

Se **defineste** astfel:

* public class **Operatii**

extends java.lang.Object

Are urmatoarele **campuri(variabile instanta)**:

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private Polinom | [**a**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#a)  primul polinom care va fi populat cu monoame formate din coeficienti si puteri luate din ArrayListurile anterioare |
| private Polinom | [**b**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#b)  al doile polinom care va fi populat cu monoame formate din coeficienti si puteri luate din ArrayListurile anterioare |
| private java.util.ArrayList<java.lang.Integer> | [**coef**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#coef)  contine toti coeficienti din parsarea string-ului dat de utilizator |
| private java.util.Iterator<java.lang.Integer> | [**i1**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#i1)  iteratorii sunt folositi pentru a popula polinoamele cu monoame formate din coeficienti si puteri luate din ArrayListurile anterioare |
| private java.util.Iterator<java.lang.Integer> | [**i2**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#i2)  iteratorii sunt folositi pentru a popula polinoamele cu monoame formate din coeficienti si puteri luate din ArrayListurile anterioare |
| private java.util.ArrayList<java.lang.Integer> | [**putere**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#putere)  contine toate puteriile din parsarea string-ului dat de utilizator |
| private Polinom | [**rest**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#rest)  polinomul rest din impartirea a doua polinoame |

**Constructorul** este de forma:

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Operatii**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#Operatii--)() |

Contine urmatoarele **metode**:

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| Polinom | [**adunare**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#adunare-Polinom-Polinom-)(Polinom a, Polinom b)  se returneaza un nou polinom format de pe urma adunarii celor doua polinoame(se folosesc metodele de adauga din clasa Polinom) |
| Polinom | [**derivare**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#derivare-Polinom-)(Polinom a)  se returneaza un nou polinom format de pe urma derivarii polinomului (se folosesc metodele de adauga din clasa Polinom,precum si metode de derivare din calsa Monom) |
| Polinom | [**getPolA**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#getPolA--)()  getter pentru primul polinom |
| Polinom | [**getPolB**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#getPolB--)()  getter pentru al doilea polinom |
| Polinom | [**getPolinomAFromParser**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#getPolinomAFromParser--)()  getter pentru primu polinom+crearea lui |
| Polinom | [**getPolinomBFromParser**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#getPolinomBFromParser--)()  getter pentru al doilea polinom+crearea lui |
| java.lang.String | [**getRest**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#getRest--)()  getter pentru polinomul rest al impartirii |
| Polinom | [**impartire**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#impartire-Polinom-Polinom-)(Polinom d, Polinom i)  se returneaza un nou polinom(catul) format de pe urma impartirii celor doua polinoame(se folosesc metodele de adauga,inmultire,grad max din clasa Polinom, precum si getCoef din clasa Monom) |
| Polinom | [**inmultire**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#inmultire-Polinom-Polinom-)(Polinom a, Polinom b)  se returneaza un nou polinom format de pe urma inmultirii celor doua polinoame(se folosesc metodele de adauga din clasa Polinom, precum si metoda de inmultit din clasa Monom) |
| Polinom | [**integrare**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#integrare-Polinom-)(Polinom a)  se returneaza un nou polinom format de pe urma derivarii polinomului (se folosesc metodele de adauga din clasa Polinom,precum si metode de integrare din calsa Monom) |
| void | [**regexChecker**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#regexChecker-java.lang.String-)(java.lang.String check)  se parseaza string-ul dat de utilizator si se populeaza arraylist-urile de coeficienti si puteri + multe verificari spre a corecta si simplifica scrierea polinoamelor |
| void | [**reset**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#reset--)()  reseteaza cele doua polinoamele |
| Polinom | [**scadere**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Operatii.html#scadere-Polinom-Polinom-)(Polinom a, Polinom b)  se returneaza un nou polinom format de pe urma scaderii celor doua polinoame(se folosesc metodele de adauga din clasa Polinom, precum si metoda de negat din clasa Monom) |

**Clasa Controller**

Clasa Controller precum ii spune si numele, leaga, controleaza si verifica informatiile primite de la interfata si model.

Se **defineste** astfel:

* public class **Controller**

extends java.lang.Object

Are urmatoarele **campuri(variabile instanta)**:

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private boolean | [**AdaugaPressed**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Controller.html#AdaugaPressed)  variabila de conditionare asupra unor apasari gresite a unor butoane |
| private int | [**nrPolinoame**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Controller.html#nrPolinoame)  variabila de numarare a aparitie pe ecran a polinomului |
| private int | [**nrPolinom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Controller.html#nrPolinom)  variabila de conditionare asupra unor apasari gresite a unor butoane |
| private Operatii | [**operatii**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Controller.html#operatii)  variabila instanta a modelului(clasa Operatii) |
| private Polinom | [**rezultat**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Controller.html#rezultat)  polinomul rezultat in urma efectuarii operatiilor din clasa model |
| private Interfata | [**view**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Controller.html#view)  variabila instanta a interfetei |

Are urmatoarele **clase interioare**:

|  |  |
| --- | --- |
| **Nested Classes** | |
| **Modifier and Type** | **Class and Description** |
| (package private) class | [**Controller.AdaugaListener**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Controller.AdaugaListener.html)  clasa ce implementeaza ascultatorul pe butonul "Adauga" Se primeste string-ul dat de utilizator, se paseaza catre clasa Operatii unde se face parsarea si se returneaza polinoamele care vor fi date View-ului sa le afiseze pe ecran+cateva conditionari de eroare |
| (package private) class | [**Controller.ClearListener**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Controller.ClearListener.html)  clasa ce implementeaza ascultatorul pe butonul "Clear" Se apeleaza metoda reset de view si metoda reset din Controller |
| (package private) class | [**Controller.StartListener**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Controller.StartListener.html)  clasa ce implementeaza ascultatorul pe butonul "Start" Se verifica ce operatie s-a selectat, comunica cu modelul si cu View-ul pentru a rezolva operatia si a afisa rezultatul pe ecran+multe pop-up-uri de erori daca utilizatorul nu a respectat scenariile scrise in documentatie |

**Constructorul** este de forma:

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Controller**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Controller.html#Controller-Interfata-Operatii-)(Interfata v, Operatii op)  constructorul care primeste si modelul si view-ul ca si parametrii, in care se mai instaneaza si Ascultatorii ce vor fi implementati in aceasta clasa |

Contine urmatoarele **metode**:

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| private void | [**resetAllIntermediates**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Controller.html#resetAllIntermediates--)()  se vor reinitializa toate variabilele instanta |

**Clasa Interfata**

Clasa interfata este partea grafica, vizuala ce va interactiona cu utilizatorul. El va putea introduce polinoame in chenarul de text si va putea apasa butoane corespunzatoare operatiilor vrute, iar mai apoi poate vedea rezultatul pe un ecran,textField.

Se **defineste** astfel:

* public class **Interfata**

extends javax.swing.Jframe

Are urmatoarele **campuri(variabile instanta)**:

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private javax.swing.JComboBox<java.lang.String> | [**alegeOperatie**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#alegeOperatie) |
| private javax.swing.JButton | [**btnAdauga**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#btnAdauga) |
| private javax.swing.JButton | [**btnClear**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#btnClear) |
| private javax.swing.JButton | [**btnStart**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#btnStart) |
| private javax.swing.JPanel | [**contentPane**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#contentPane) |
| private javax.swing.JLabel | [**lblAfisari**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#lblAfisari) |
| private javax.swing.JLabel | [**lblIntroducetiPolinomul**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#lblIntroducetiPolinomul) |
| private javax.swing.JLabel | [**lblSelecteazaOperatia**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#lblSelecteazaOperatia) |
| private java.lang.String[] | [**names**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#names)  nume folosite la JComboBox; |
| private javax.swing.JScrollPane | [**scrollPane**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#scrollPane) |
| private javax.swing.JTextArea | [**textArea**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#textArea) |
| private javax.swing.JTextField | [**textField**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#textField) |

**Constructorul** este de forma:

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Interfata**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#Interfata--)()  contructorul interfetei, foloseste BoxLayout, nu e resizable; |

Contine urmatoarele **metode**:

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| (package private) void | [**addAdaugaListener**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#addAdaugaListener-java.awt.event.ActionListener-)(java.awt.event.ActionListener adauga)  ascultator pentru butonul adauga |
| (package private) void | [**addClearListener**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#addClearListener-java.awt.event.ActionListener-)(java.awt.event.ActionListener clear)  ascultator pentru butonul clear |
| (package private) void | [**addJComboBoxListener**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#addJComboBoxListener-java.awt.event.ActionListener-)(java.awt.event.ActionListener comboBox)  ascultator pentru combo box |
| (package private) void | [**addStartListener**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#addStartListener-java.awt.event.ActionListener-)(java.awt.event.ActionListener start)  ascultator pentru butonul start |
| (package private) javax.swing.JComboBox<java.lang.String> | [**getComboBox**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#getComboBox--)()  getter pentru textField |
| (package private) java.lang.String[] | [**getStringFromComboBox**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#getStringFromComboBox--)()  getter pentru names |
| (package private) java.lang.String | [**getUserInput**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#getUserInput--)()  getter pentru textField |
| (package private) void | [**reset**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#reset--)()  reseteaza campul textArea; |
| (package private) void | [**setText**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#setText-java.lang.String-)(java.lang.String text)  setter pentru textArea de string |
| (package private) void | [**setTextPolinom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#setTextPolinom-java.lang.String-Polinom-)(java.lang.String s, Polinom a)  setter pentru textArea de Polinom |
| (package private) void | [**showError**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Interfata.html#showError-java.lang.String-)(java.lang.String errMessage)  pop-up error panel, foloseste JOptionPane |

(metodele sunt package private, a.k.a default, deoarece nu trebuie sa fie accesate de alte clase din alte pachete, chestier ce se poate schimba in functie de necesitati)

**Clasa Main**

Folosita pentru a initializa MVC si pentru a avea metoda main().

Se **defineste** astfel:

* public class **Main**

extends java.lang.Object

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Main**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Main.html#Main--)() |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| static void | [**main**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\Main.html#main-java.lang.String:A-)(java.lang.String[] args)  se vor initializa interfata operatiile si controllerul |

# Testare

Clasa de test, numita in proiect **TestClass** contine variabile instanta gata pregatite pentru a testa toate metodele din modelul programului, clasa Operatii.Astfel avem testare pe adunare,scadere,inmultire,impartire,derivare,integrare, cat si pe parsare din string in polinom,dar si initialziare de polinom.Toate testele au un rezultat pozitiv, demonstrand corectitudinea programului cat si eficienta sa.

Se **defineste** astfel:

* public class **TestClass**

extends java.lang.Object

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| (package private) Monom[] | [**a**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#a) |
| (package private) Monom[] | [**adunare**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#adunare) |
| (package private) Monom[] | [**b**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#b) |
| (package private) Monom[] | [**derivare**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#derivare) |
| (package private) java.util.Iterator<Monom> | [**i1**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#i1) |
| (package private) java.util.Iterator<Monom> | [**i2**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#i2) |
| (package private) Monom[] | [**impartire**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#impartire) |
| (package private) Monom[] | [**inmultire**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#inmultire) |
| (package private) Monom[] | [**integrare**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#integrare) |
| (package private) Operatii | [**op**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#op) |
| (package private) Polinom | [**p1**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#p1) |
| (package private) Polinom | [**p2**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#p2) |
| (package private) Monom[] | [**scadere**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#scadere) |
| (package private) java.lang.String | [**StringtoTest**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#StringtoTest) |
| (package private) Polinom | [**test**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#test) |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**TestClass**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#TestClass--)() |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| void | [**adunare**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#adunare--)()  testarea adunare doua polinoame |
| void | [**derivare**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#derivare--)()  testarea derivare doua polinoame |
| void | [**impartire**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#impartire--)()  testarea impartire doua polinoame |
| void | [**inmultire**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#inmultire--)()  testarea inmultire doua polinoame |
| void | [**integrare**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#integrare--)()  testarea integrare doua polinoame |
| void | [**regexTester**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#regexTester--)()  testarea parsare corecta din string in polinom |
| void | [**scadere**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#scadere--)()  testarea scadere doua polinoame |
| void | [**verificarePolinom**](file:///C:\Users\Vlad\Desktop\Vlad%20Brincoveanu@An2-Sem1\JAVA%20WORK\Tema1\tema1-tp\doc\TestClass.html#verificarePolinom--)()  testarea initializare polinom |

# Concluzii si Dezvoltari Ulterioare

Prin dezvoltarea acestui program am folosit tehnici de oop, precum mostenire, agregare, etc.Am folosit un semi model MVC, precum si testari, am facut si o documentatie programului destul de riguroasa, rezultand un program simplist de operatii pe polinoame, care poate fi folosit de oricine.

-utilizarea Collections.sort

-hashtable in loc de arraylist

-binary search pentru adaugare

-o interfata mai user friendly si cu mai multe optiuni

-posibilitatea de a face polinoame de n variabile si de a calcula valori din ele

# Bibliografie

* <https://ro.wikipedia.org/wiki/Polinom>
* https://en.wikipedia.org/wiki/Polinom
* <http://coned.utcluj.ro/~marcel99/PT/>
* http://www.oracle.com/