DOCUMENTATIE

TEMA NUMARUL 1

NUME STUDENT: Turcu Cezar Simeon

GRUPA: 30229

# CUPRINS

[1. Obiectivul temei 3](#_Toc97470163)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](#_Toc97470164)

[3. Proiectare 5](#_Toc97470165)

[4. Implementare 9](#_Toc97470166)

[5. Rezultate 12](#_Toc97470167)

[6. Concluzii 13](#_Toc97470168)

[7. Bibliografie 14](#_Toc97470169)

# Obiectivul temei

* **Principalul obiectiv**

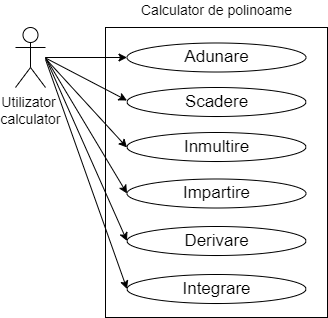
Realizarea design-ului si implementarii unui calculator de polinoame cu o interfata grafica cu utilizatorul dedicata, prin intermediul careia acesta poate insera polinoame, alege operatia matematica ce va fi realizata si viziona rezultatul.

* **Obiective secundare**

1. Modelarea unui polinom ca fiind o lista de monoame
2. Posibilitatea de a introduce un polinom ca si un sir, in orice ordine a exponentilor monoamelor, acesta urmand sa fie automat impartit in monoamele componente
3. Modelarea operatiilor de adunare, scadere, inmultire, impartire, derivare si integrare
4. Afisarea rezultatului unui calcul sub o forma lizibila, usor de inteles
5. Realizarea design-ului interfetei grafice cu utilizatorul
6. Testarea calculatorului de polinoame folosind JUnit

# Analiza problemei, modelare, scenarii, cazuri de utilizare

* **Diagrama use case**



**Use case:** Adunare, Scaderea sau Inmultirea a a doua polinoame

**Actor principal:** Utilizator

**Pasi in caz de succes:**

1. Utilizatorul introduce 2 polinoame in interfata grafica, in cazul in care un polinom ramane neintrodus, acesta va fi considerat ca fiind egal cu 0.
2. Calculatorul se asigura ca polinoamele introduse au un format corect, folosind un ‘x’ litera mica ca si variabila.
3. Utilizatorul alege butonul etichetat “Addition”, “Subtraction” sau “Multiplication” din interfata grafica.
4. Calculatorul realizeaza operatia aleasa de utilizator pentru cele doua polinoame si afiseaza rezultatul acesteia in interfata grafica.

**Secventa alternativa:** Polinoamele introduse nu respecta formatul

1. Calculatorul afiseaza un mesaj de avertizare in care il informeaza pe utilizator sa foloseasca doar un ‘x’ litera mica ca si variabila
2. Scenariul se intoarce la pasul 1

**Use case:** Impartirea a a doua polinoame

**Actor principal:** Utilizator

**Pasi in caz de succes:**

1. Utilizatorul introduce 2 polinoame in interfata grafica, in acest caz ambele fiind obligatorii pentru realizarea calculului.
2. Calculatorul se asigura ca polinoamele introduse au un format corect, folosind un ‘x’ litera mica ca si variabila si ca niciunul dintre acestea nu are valoarea ‘0’.
3. Utilizatorul alege butonul etichetat “Division” din interfata grafica.
4. Calculatorul realizeaza impartirea celor 2 polinoame, considerand deimpartitul ca fiind polinomul cu gradul cel mai mare al exponentului si afiseaza rezultatul acesteia in interfata grafica, separat in cat si rest.

**Secventa alternativa:** Polinoamele introduse nu respecta formatul sau unul dintre ele ramane neintrodus

1. Calculatorul afiseaza un mesaj de avertizare in care il informeaza pe utilizator sa foloseasca doar un ‘x’ litera mica ca si variabila sau, in cazul in care unul dintre polinoame ramane neintrodus utilizatorul este avertizat sa introduca un polinom valid.
2. Scenariul se intoarce la pasul 1

**Use case:** Derivarea sau Integrarea a a doua polinoame

**Actor principal:** Utilizator

**Pasi in caz de succes:**

1. Utilizatorul poate introduce niciunul, unul sau doua polinoame in interfata grafica, polinoamele neintroduse fiind considerate ca fiind egale cu ‘0’.
2. Calculatorul se asigura ca polinoamele introduse au un format corect, folosind un ‘x’ litera mica ca si variabila.
3. Utilizatorul alege butonul etichetat “Derivation” sau “Integation” din interfata grafica.
4. Calculatorul realizeaza calculul derivatei sau a integratei pentru ambele polinoame si afiseaza rezultatele in interfata grafica, rezultatul pentru polinoamele neintroduse fiind cel al derivarii sau integrarii lui ‘0’.

**Secventa alternativa:** Polinoamele introduse nu respecta formatul

1. Calculatorul afiseaza un mesaj de avertizare in care il informeaza pe utilizator sa foloseasca doar un ‘x’ litera mica ca si variabila.
2. Scenariul se intoarce la pasul 1

* **Cerinte functionale**

**-**Calculatorul ar trebui sa ii permita utilizatorului sa insereze polinoame

**-**Calculatorul ar trebui sa ii permita utilizatorului sa stearga si sa reintroduca polinoamele inainte si dupa selectarea operatiei ce va fi realizata

**-**Calculatorul ar trebui sa ii permita utilizatorului sa aleaga operatia pe care acesta doreste sa o realizeze

**-**Calculatorul ar trebui sa avertizeze utilizatorul in cazul in care input-ul nu este unul corect si sa nu permita continuarea pana cand nu este realizata corectia

**-**Calculatorul ar trebui sa realizeze operatia aleasa odata ce input-ul este corect

**-**Calculatorul ar trebui sa afiseze rezultatul obtinut in urma realizarii operatiei aleasa de catre utilizator

* **Cerinte non-functionale**

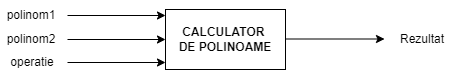
**-**Calculatorul ar trebui sa fie intuitiv si usor de folosit de catre utilizator

**-**Calculatorul at trebui sa permita realizarea succesiva a mai multor calcule intr-un mod usor, fara a fi necesata inchiderea si repornirea interfetei

**-**Calculatorul ar trebui sa poata permite introducerea polinoamelor fara a necesita ca un polinom sa fie ordonat dupa puterile monoamelor din care este alcatuit

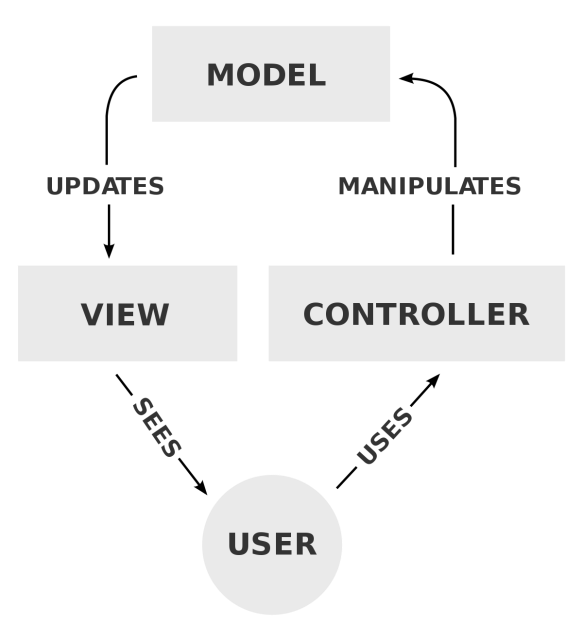
# 3. Proiectare

* **Design-ul general al sistemului**



Utilizatorul introduce in interfata grafica cu utilizatorul cele doua polinoame si alege din aceeasi interfata operatia care va dori sa fie realizata cu input-urile, urmand ca rezultatul sa fie afisat in GUI.

* **Arhitectura**

****

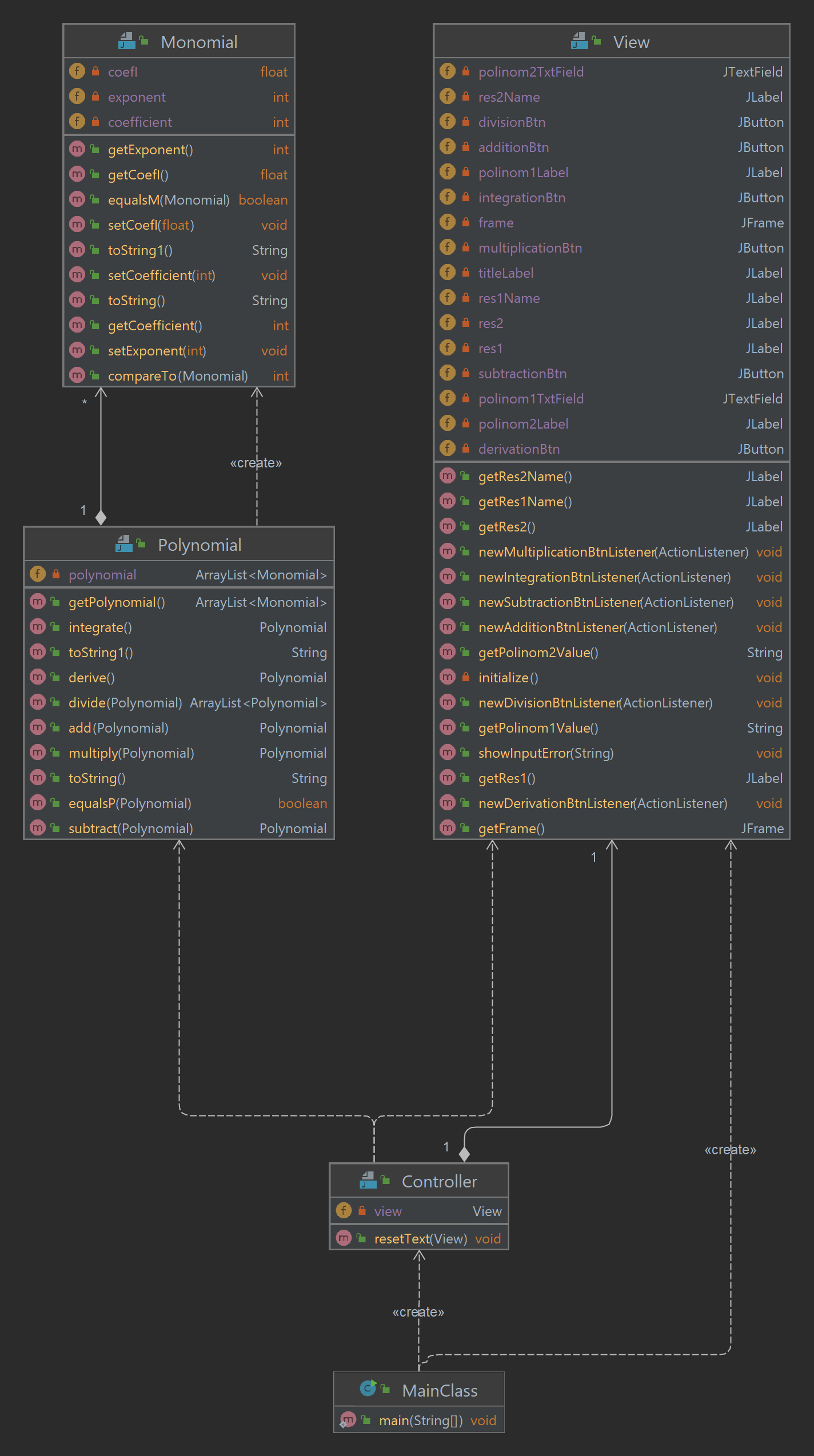
Am ales sa folosesc un model arhitectural MVC(Model, View, Controller) pentru implementarea Calculatorului de polinoame intrucat acesta izoleaza logica de calcul fata de considerentele interfetei cu utilizatorul, utilizand o aplicatie in care aspectul vizual este mai usor de modificat, fara a afecta alte nivele, permitand astfel dezvoltari ulterioare.

-Model: Manageriaza datele si functionalitatea aplicatiei, in acest caz implementeaza clasele pentru modelarea monoamelor, polinoamelor, a operatiilor cu polinoame si afisarea intuitiva a polinoamelor.

-View: Ii corespunde interfetei grafice cu utilizatorul. Aceasta ii afiseaza informatie utilizatorului si obtine datele pe care le afiseaza de la clasele de model. In calculatorul de polinoame reprezinta interfata grafica in care utilizatorul introduce polinoamele, alege butonul pentru operatia pe care vrea sa o realizeze si vede rezultatul operatiei.

-Controller: Fiecare view are o componenta de controller asociata. Controllerele primesc un input, de obicei ca si evenimente asociate miscarii mouse-ului, activarii unui buton sau ca si input-uri de la tastatura. Aceste evenimente sunt traduse in cereri de resurse ce sunt transmise modelului sau vederii. In acest caz, controller-ul primeste polinoamele din interfafta grafica pe care le transmite clasei de polinoame unde sunt transformate intr-o lista de monoame. Apoi controller-ul interpreteaza ce buton a apasat utilizatorul in view si apeleaza operatia respectiva din clasa Polynomial si transmite rezultatul vederii pentru a fi afisat.

* **Diagrama UML**

****

# 4. Implementare

* **Clase folosite, campuri si metode importante**

**Pachetul Model:**

**Clasa Monomial:**

-Contine campurile: coefficient, coefI, exponent ce corespund coeficientului(valoare intreaga in toate cazurile mai putin pentru rezultatul integrarii unui monom) si exponentului unui monom.

-Cele mai importante metode sunt

* 1. Metoda suprascrisa CompareTo ce permite compararea monoamelor in functie de exponentul acestora, intr-un polinom monoamele fiind ordonate in mod descrescator dupa exponentele monoamelor.
  2. Metoda suprascrisa toString ce permite afisarea intuitiva a monoamelor, implementand reguli precum: daca exponentul unui monom este 1 , nu se va afisa niciun exponent, daca exponentul este 0, nu se va afisa nici variabila ‘x’, etc.
  3. Metoda equalsM ce defineste doua monoame ca fiind egale atunci cand coeficientii si exponentii acestora sunt egale intre ele.

**Clasa Polynomial:**

**-**Contine campul: polynomial ce reprezinta un ArrayList de Monomial.

**-**Exista doi constructori

1. Primeste ca si parametru un ArrayList de Monomial si creeaza un obiect de tip Polynomial al carui camp polynomial este reprezentat de parametrul primit
2. Primeste ca si parametru un String reprezentat de input-ul introdus de catre utilizator in interfata grafica si foloseste urmatoarea expresie regulata pentru a imparti Stringul introdus in monoamele sale componente."(?=.+)([+-]?[0-9]\*(?:\\\*?x(?:\\^[0-9]+)?)?)"

Cat timp in Stringul introdus se mai pot regasi portiuni ce respecta formatul cerut de acest regEx, monoamele regasite sunt impartite in campurile clasei monomial, adica in coeficient si exponent, conform expresiiei regulate: "(?=.+)([+-]?[0-9]\*)(x?)(?:\\^?)([0-9]\*?)"

Aceasta imparte monomul intr-un prim grup ce contine semnul si o cifra, reprezentand coeficientul, un al doilea ce reprezinta variabila x, si un ultim grup reprezentat de exponent.

In urma tratarii cazurilor particulare, in care cifra coeficientului lipseste fiind considerata egala cu ‘1’, si a altor posibilitati ce pot aparea, lista de monoame rezultata este ordonata descrescator in functie de exponentii monoamelor, pentru a usura calculele viitoare, folosind metoda Collections.sort, ce se foloseste de metoda compareTo din clasa Monomial.

-Cele mai importante metode:

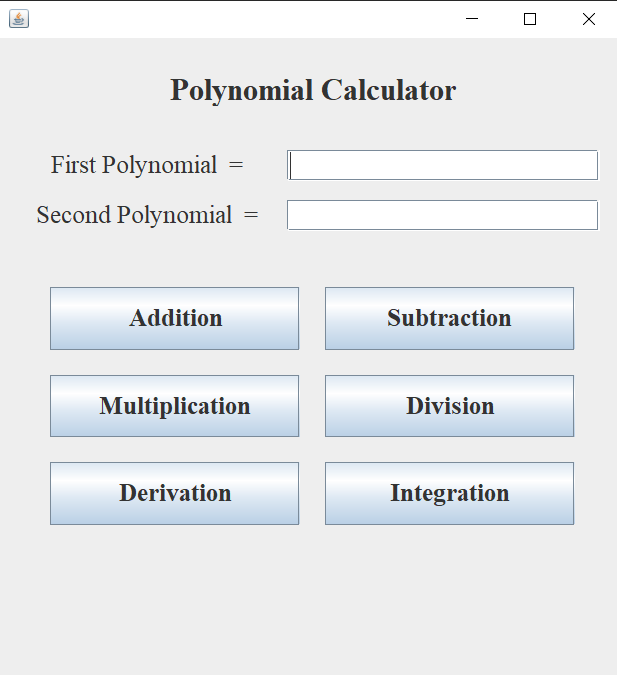
1. add - Primeste ca parametru un alt polinom, creeaza un nou ArrayList, dupa care incepe parcurgerea celor doua polinoame. Daca primul monom al primului polinom are exponentul mai mare decat primul monom al celui de-al doilea polinom, coeficientul si exponentul monomului nou creat sunt cele ale primului polinom, acest nou monom fiind adaugat la ArrayListul pentru rezultat, iar indexul primul polinom fiind incrementat. Analog in cazul in care cel de-al doilea polinom are exponentul mai mare. Daca, inca cele doua sunt egale, coeficientul noului monom este suma celor doi coeficienti iar ambii indecsi sunt incrementati. La final, daca exista polinoame ce nu au fost parcurse pana la final, restul monoamelor sunt adaugate secvential la rezultat.
2. subtract - Analog operatiei de add insa coeficientul noului sum este egal cu diferenta coeficientilor celor doua monoame.
3. multiply- Algoritmul foloseste doua bucle for imbricate pentru a inmutli fiecare monom al unui polinom cu fiecare monom al celuilalt polinom. Coeficientul monomului rezultat va fi produsul celor doi coeficienti iar exponentul suma exponentilor. Din ArrayListul rezultat, daca exista monoame cu acelasi exponent, acestea vor fi insumate iar rezultatul final va fi sortat.
4. derive- Se parcurge polinomul cu care se apeleaza metoda folosind o bucla foreach, noul coeficient fiind exponentul monomului inmultit cu coeficientul acestuia iar noul exponent fiind exponentul monomului din care se scade ‘1’.
5. divide- Se decide care polinom va fi deimpartit si care impartitor prin compararea exponentului primul monom din cele doua polinoame. Cat timp exponentul restului este mai mare sau egal cu exponentul impartitorului, se imparte resul la impartitor, iar rezultatul se adauga la cat. Noul rest va fi fostul rest din care se va scade produsul dintre impartiror si ultimul termen al catului.
6. integrate- - Se parcurge polinomul cu care se apeleaza metoda folosind o bucla foreach, noul coeficient fiind raportul dintre coeficientul monomului si exponentul acestuia la care se aduna 1, iar noul exponent fiind exponentul monomului + ‘1’.

**Pachetul View:**

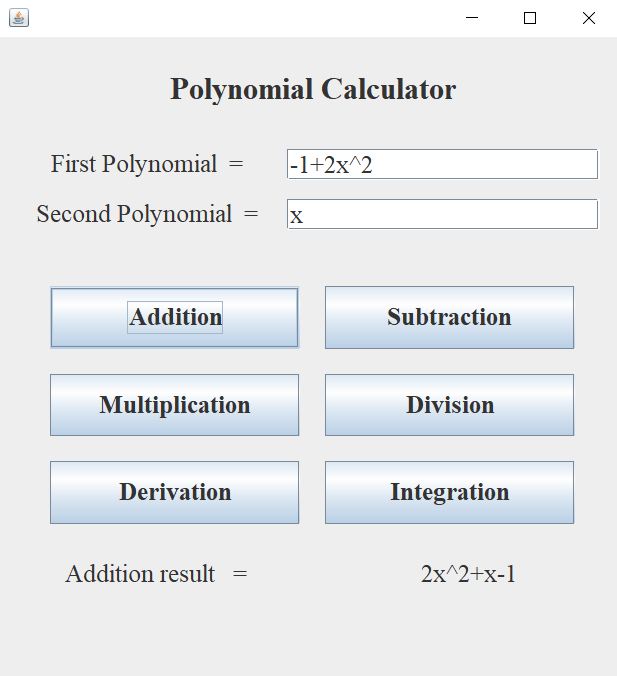
**Clasa View:**

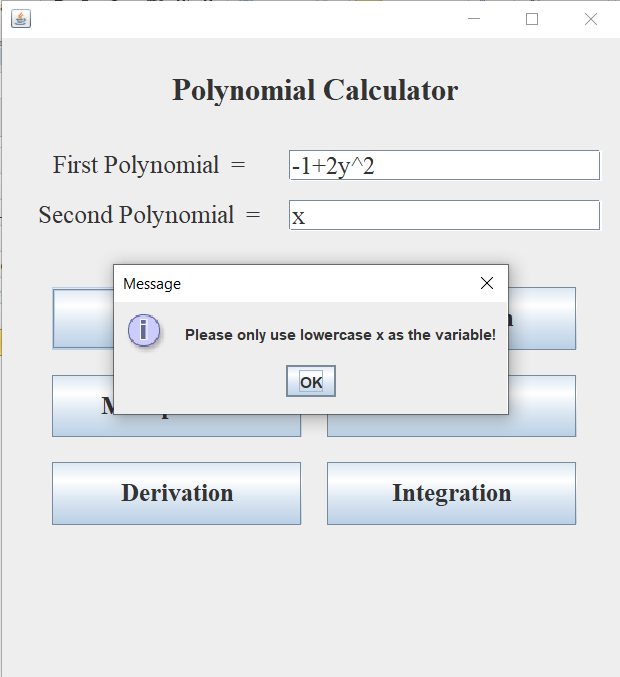
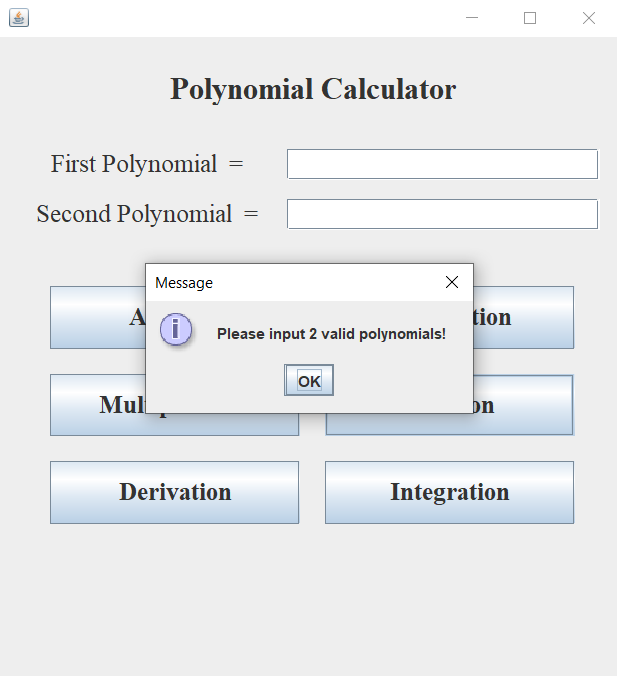
-Contine instante ale claselor Jlabel pentru etichetarea campurilor unde se introduc cele doua polinoame si a afisarii rezultatelor calculelor, instante ale JtextField in care se introduc Stringurile reprezentative pentru polinoame si instante ale Jbutton pentru crearea butoanelor de pe care se alege operatia ce va fi realizata.

-Contine metode pentru adaugarea de buttonListeners pentru butoanele de realizarea a operatiilor, metode de obtinere a textului din textField pentru a trasmite Stringurile la clasa de Polynomial si metode de setare a textului in Jlabels pentru afisarea rezultatului si a etichetarii acestuia conform operatiei realizate.



In cazul introducerii conform formatului a celor doua polinoame, in partea de jos a interfetei grafice va fi afisat rezultatul

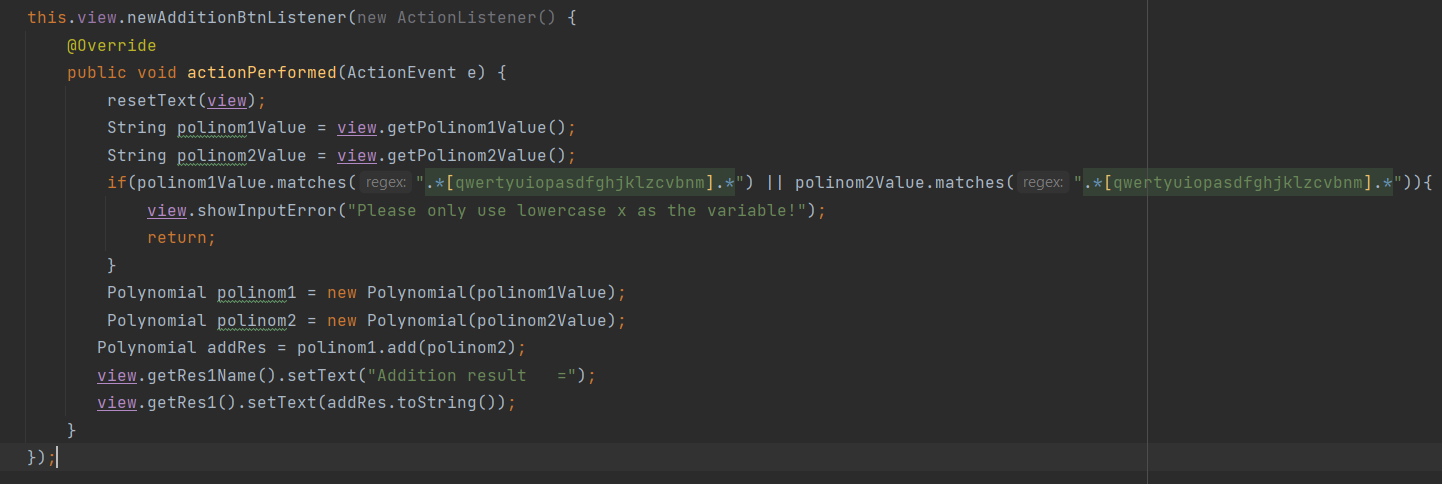


In caz contrar, se va deschide o fereastra in care utilizatorul este informat despre greseala realizata, apoi fiindu-i permis sa reintroduca inputul.

**Pachetul Controller:**

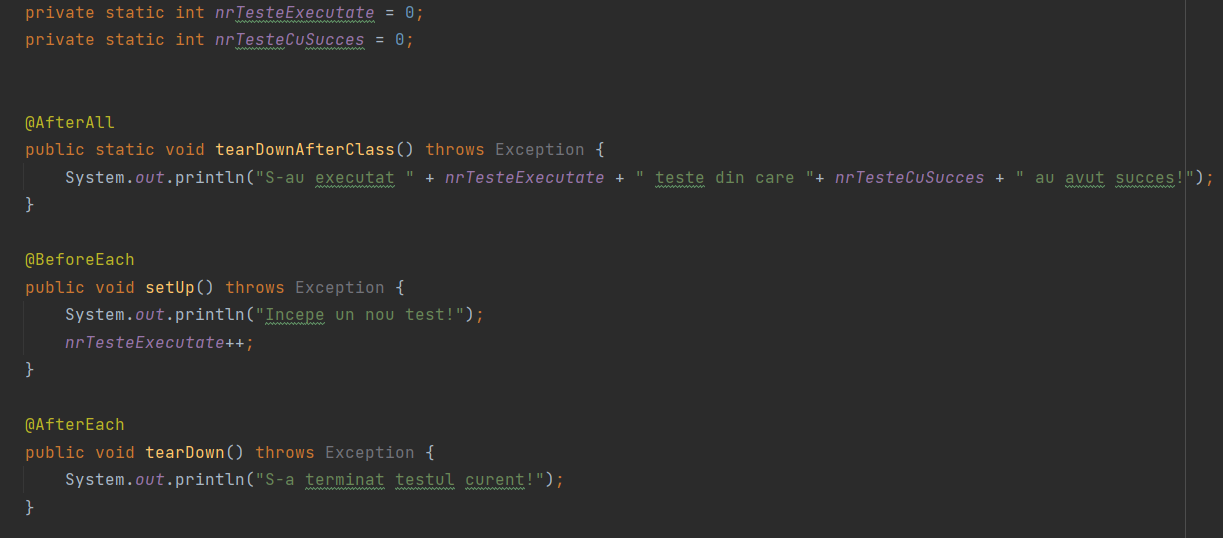
**Clasa Controller:**

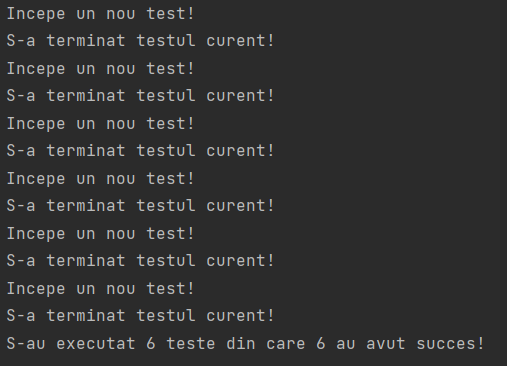
**-**Implementeaza buttonListeners pentru butoanele corespunzatoare celor 6 operatii prin intermediul unor clase anonime, in cadrul carora se obtin Stringurile pentru polinoame prin metodele de getText ale JtextField, se verifica ca polinomul sa nu contina alte variabile, caz in care se afiseaza un mesaj de eroare, se apeleaza metoda de calcul corespunzatoare butonului si se afiseaza rezultatul calculului.



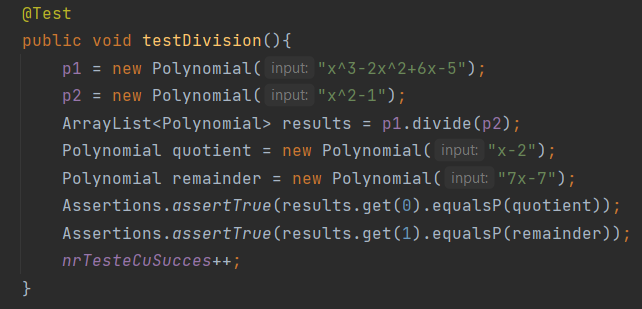
# 5. Rezultate

Am ales sa realizez testarea unitarea cu Junit, declarand variabile pentru numarul total de teste realizate si cele realizate cu succes, ambele fiind initializate cu 0. Inaintea fiecarui test executat se anunta inceperea acestuia si este incrementat numarul de teste realizate. Dupa fiecare test se anunta sfarsitul executiei acestora iar la finalul tuturor testelor sunt afisare numarul total de teste executate si cele executate cu succes.



****

In cadrul fiecarui test, se declara doua polinoame folosite ca si input si un al treilea polinom(eventual un al patrulea pentru operatiile de impartire, derivare si integrare) ca fiind rezultatul asteptat in urma realizarii calculului pentru inptutul introdus de utilizator. Apoi folosim metoda assertTrue si equalsP din clasa Polynomial pentru a verifica daca polinomul rezultat in urma calculului este egal cu polinomul reprezentand rezultatul asteptat. In caz afirmativ, testul este executat cu succes.



# 6. Concluzii

In concluzie, Calculatorul de polinoame reprezinta o unealta folositoare oricarei persoane care necesita sa realizeze adesea operatii de baza cu polinoamele, utilizarea acestuia fiind mult simplificata de interfata grafica cu utilizatorul.  
Prin intermediul acestui proiect am putut sa invat mai multe despre structura de proiectare MVC, am invatat bazele folosirii regEx si am imbogatit cunostiintele legate de interfata grafica cu utilizatorul.  
Consider ca unele dezvoltari ulterioare ce s-ar putea realiza pentru acest proiect ar fi posibilitatea calcululelor si pentru polinoame mai complexe, precum cele ce contin functii exponentiale si trigonometrice.

# 7. Bibliografie

1. Learn Regular Expressions in 20 minutes - <https://youtu.be/rhzKDrUiJVk>
2. Regex101 - <https://regex101.com/r/nI5oA5/6>
3. PT Courses - <https://dsrl.eu/courses/pt/materials/A1_Support_Presentation.pdf>
4. MVC Architecture - <https://ro.wikipedia.org/wiki/Model-view-controller>