DOCUMENTAȚIE

NUME: IONAȘ ALEX-RĂZVAN

# CUPRINS

[1. Obiectivul temei 3](file:///C:\Users\Alex\Downloads\PT2021-2022_Documentatie.doc#_Toc95297885)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](file:///C:\Users\Alex\Downloads\PT2021-2022_Documentatie.doc#_Toc95297886)

[3. Proiectare 4](file:///C:\Users\Alex\Downloads\PT2021-2022_Documentatie.doc#_Toc95297887)

[4. Implementare 4](file:///C:\Users\Alex\Downloads\PT2021-2022_Documentatie.doc#_Toc95297888)

[5. Rezultate 6](file:///C:\Users\Alex\Downloads\PT2021-2022_Documentatie.doc#_Toc95297889)

[6. Concluzii 6](file:///C:\Users\Alex\Downloads\PT2021-2022_Documentatie.doc#_Toc95297890)

[7. Bibliografie 6](file:///C:\Users\Alex\Downloads\PT2021-2022_Documentatie.doc#_Toc95297891)

# Obiectivul temei

Obiectivul principal presupune dezvoltarea unei aplicații Java capabilă de a efectua operații de adunare, scădere, înmulțire, împărțire, derivare și integrare asupra polinoamelor.

De asemenea, trebuie creată o interfață prin intermediul căreia utilizatorul aplicației să introducă polinoamele și operatorii menționați.

Obiectivele secundare care stau la baza îndeplinirii obiectivului principal sunt următoarele:

1. abstractizarea informațiilor preluate din cerință
2. determinarea claselor și atributelor, aceasta reprezentând partea de structură a proiectului
3. determinarea metodelor specifice pentru a îndeplini funcționalitatea cerută

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

Cum am precizat, aplicația trebuie să implementeze următoarele cerințe:

1. calculatorul de polinoame trebuie sa permită utilizatorilor să introducă polinoame
2. utilizatorul trebuie sa poată selecta operația care dorește sa fie aplicată polinoamelor
3. calculatorul trebuie să poată efectua operația de adunare și scădere asupra polinoamelor
4. calculatorul trebuie să poată efectua operația de înmulțire și împarțire a polinoamelor
5. calculatorul trebuie să poată efectua operația de derivare și integrare a unui polinom

De asemenea, calculatorul trebuie să

1. fie intuitiv și ușor de folosit
2. să furnizeze rezultate corecte
3. să furnizeze rapid rezultatele

Scenariile posibile sunt legate de felul în care utilizatorii folosesc aplicația. Pentru a avea un scenariu de succes, utilizatorul trebuie să introducă date valide în câmpurile “Polinom A” și “Polinom B”, după care, în cazul adunării și a scăderii, trebuie să apese pe câmpul “Rezultat” și să selecteze operatorul dorit. În cazul înmulțirii și a împărțirii, nu este necesară selectarea campului specific polinomului rezultat. Pentru derivare și integrare, nu este necesară intdorucerea unui polinom în cel de-al doilea câmp.

Dacă datele de intrare nu sunt valide, atunci rezultatul va fi de asemenea invalid.

# 3. Proiectare

Pentru proiectare am folosit două clase principale de obiecte: Monom și Polinom. Prima are două atribute: “coef” reprezentând coeficientul monomului și „pow” reprezentând puterea monomului care e de forma: mX^n. Polinom are drept atribute o listă de monoame. Pentru modularizare și cod ușor de citit, am folosit o arhitectură de tipul MVC. Astfel clasa Model reprezintă clasa în care am implementat operațiile cerute. Clasa GUI(View) reprezintă interfața aplicației cu butoane, câmpuri de text și altele, iar clasa Controller leagă cele două clase prezentate, implementând în ea “listenerii” pentru butoane.

Pe partea de algoritmi, atât adunarea, cât și scăderea, se fac termen cu termen, unde termenii au același exponent. La înmulțire, am înmulțit fiecare monom din polinomul A cu fiecare monom din polinomul B, iar la final am adunat tot.

# 4. Implementare

Descrierea clasei Monom:  
 Clasa monom creeaza un obiect de tipul: mX^n, unde m reprezintă coeficientul, iar n reprezintă exponentul. O metoda importantă o reprezintă compareTo, pe care o folosesc pentru a afisa polinomul rezultat în ordinea descrescătoare a coeficienților. De asemenea, metoda toString este folosită pentru a afișa polinomul fără termeni redundanți (de exemplu: 0.0x^5, 5x^0.0).

Descrierea clasei Polinom:

Clasa polinom creeaza o listă de monoame, obiectul final având forma: aX^b + cX^d + … +zX^w. Metoda toString\_2 returneaza polinomul sub formă de text pentru a putea fi afișat în interfață.

Descrierea clasei Model:

Metoda descPolinoame preia două polinoame sub formă de text și le descompune pe fiecare într-un polinom de tipul definit mai sus.

Metoda addPolinoame se ocupă de adunarea polinoamelor. Primește ca parametri două polinoame sub formă de șir de caractere, le descompune folosind metoda descPolinoame, iar adunarea termen cu termen se face cu ajutorul metodei adună. In mod similar se procedează pentru scaderea polinoamelor, implementată în metoda subPolinoame.

Metoda multiplicaPolinoame, de asemenea este similară cu metoda de adunare a polinoamelor, doar că in loc de a folosi metoda “aduna”, se folosește metoda “multiplica” care înmulțește polinoamele termen cu termen, după care se adună.

Metodele integreaza și deriveaza se ocupă cu integrarea polinomului din primul câmp. Acestea vor folosi metoda descPolinoame pentru a descompune polinomul în structura de date descrisă anteterior, după care vor modifica coeficienții și exponenții fiecarui monom pentru a corespunde unui rezultat corect din punct de vedere matematic.

Descrierea clasei GUI

Clasa GUI reprezintă interfața aplicației care constă din mai multe butoane și câteva câmpuri de text. Pentru introducerea datelor se pot folosi butoanele “X”, “^”, “.”, “+”, “-“, 0-9, dar se pot introduce și de la tastatură. Folosirea butoanelor reprezintă un avantaj față de introducerea textului de la tastatură, întrucât se garantează introducerea unui set de date valid, fapt implementat cu ajutorul unor algoritmi. Pentru ștergerea datelor introduse într-un câmp, se poate folosi butonul “AC”. Pentru ștergerea unui singur caracter, se poate folosi butonul “C”.

Descrierea clasei Controller

Clasa Controller leagă ultimele două clase prezentate. Aici se implementează acțiunile fiecărui buton și funcționalitatea sa. Printre cele mai importante clase implementate in Controller, se numără AddXListener, AddPowListener, AddPlusListener, AddMinusListener, AddPointListener. Aceste clase ne garantează că, daca utilizatorul va folosi butoanele, datele de intrare vor fi valide. De exemplu AddXListener se ocupă cu introducerea caracterului “X” în câmpul de introducere a polinomului. Dacă dorim să introducem “X^X”, acest lucru nu va fi posibil, deoarece se verifică să nu fie introdus caracterul “^” inaintea acestuia, până la primul + sau -. Celelalte clase mentionate functionează in mod similar.

Descrierea clasei Main

Clasa Main este clasa în care fiecare obiect al arhitecturii este instanțiat.

# 5. Rezultate

În urma testărilor efectuate cu JUnit, am constatat că operațiile se execută în mod corect

# 6. Concluzii

Pentru a dezvolta și mai mult aplicația, se poate implementa operația de împărțire a polinoamelor, dar și implementarea operațiilor în cazurile în care în loc de coeficienți reali, avem parametri.

# 7. Bibliografie

<https://www.vogella.com/tutorials/JUnit/article.html>

https://docs.oracle.com/javase/tutorial/uiswing/