

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE



UNIVERSITATEA TEHNICĂ

DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
CATEDRA CALCULATOARE**

DOCUMENTAȚIE

TEMA 1

Teglă Lidia-Sorina

Grupa 30226 | An 2 semestrul 2

CUPRINS

1.	Obiectivul temei.....	3
2.	Analiza problemei, modelare, scenarii, cazuri de utilizare	3
3.	Proiectare	4
4.	Implementare	5
5.	Rezultate	5
6.	Concluzii.....	6
7.	Bibliografie	6

1. Obiectivul temei

Obiectivul principal al temei este de a proiecta și implementa un calculator de polinoame. Obiectivul secundar constă în crearea unei interfețe grafice prin intermediul căreia utilizatorul poate introduce polinoame, selecta operația matematică (adică adunare, scădere, înmulțire, împărțire, derivare, integrare) de efectuat și vizualiza rezultatul.

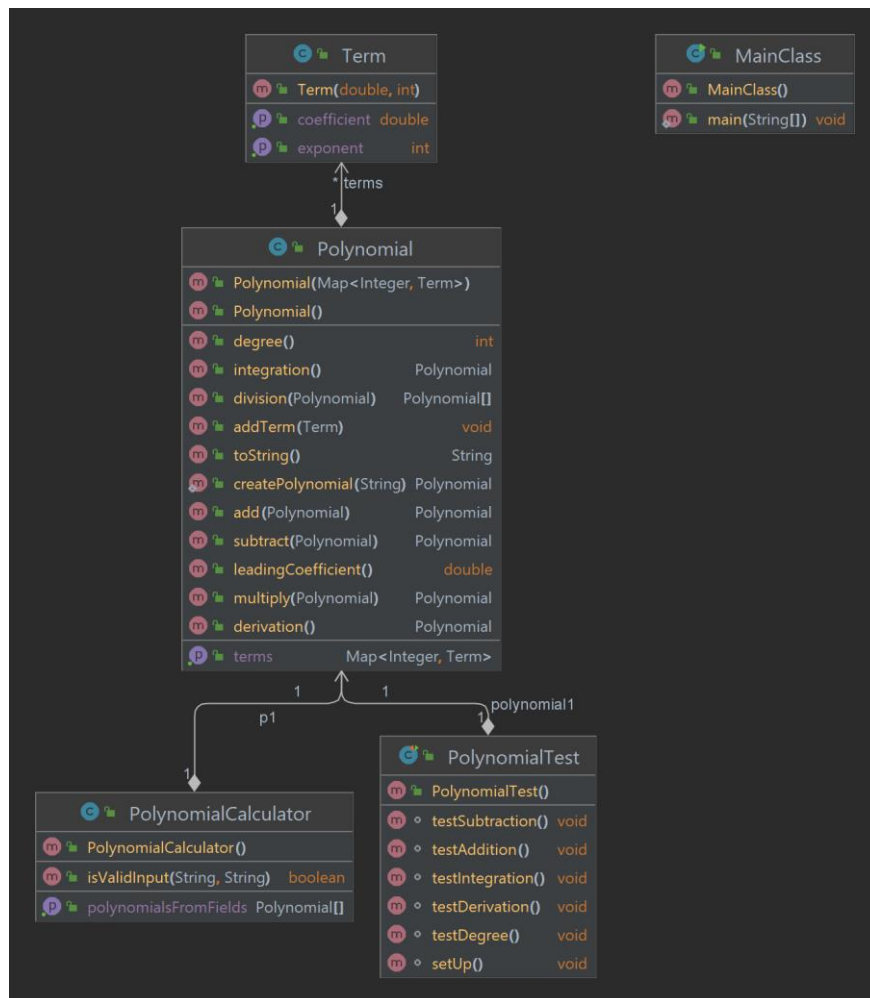
2. Analiza problemei, modelare, scenarii, cazuri de utilizare

The screenshot shows a window titled "Polynomial Calculator". It contains three input fields labeled "Polynomial 1:", "Polynomial 2:", and "Result:". Below these fields is a grid of buttons for mathematical operations: "Add", "Subtract", "Multiply", "Derivative", "Integrate", "Divide", and "Clear". The "Clear" button is located at the bottom left of the button grid.

- Polinoamele vor fi introduse în câmpurile *Polynomial 1* și *Polynomial 2*, utilizatorul va apăsa butonul corespunzător operației pe care acesta dorește să o execute, iar rezultatul va fi proiectat în câmpul *Result*. Pentru a introduce din nou alte polinoame se va apăsa butonul *Clear*.
- Pentru a valida introducerea polinoamelor, acestea trebuie să fie în formatul " $a_n x^n + a_{n-1} x^{n-1} + \dots + a_0 x^0$ " adică " $a_n x^n + a_{n-1} x^{n-1} + \dots + a_0 x^0$ ".
- Pentru operațiile *Add*, *Subtract*, *Multiply* și *Divide* este nevoie ca ambele câmpuri să fie completate, altfel va fi afișat un *DialogBox* cu mesajul *Please enter valid polynomials on both fields*.
- Pentru operațiile cu un singur polinom, adică derivare și integrare, trebuie să fie introdus un singur polinom în oricare dintre câmpurile desemnate, altfel va fi afișat un *DialogBox* cu mesajul *Please enter one single polynomial*.

3. Proiectare

- Pentru o bună organizare a codului sursă am creat pachetele *GUI* și *PolynomialStuff*. În pachetul *GUI* am adăugat clasele legate de interfața grafică, *PolynomialCalculator* și *MainClass*. În pachetul *PolynomialStuff* am adăugat clasele *Polynomial* și *Term*, legate de implementarea efectivă a ceea ce se află în spatele interfeței.
- *Polynomial* este clasa în care am implementat metodele pentru operațiile menționate și alte metode ajutătoare. Clasa *Term* este o clasă ajutătoare pentru a gestiona termenii polinomului.
- Pentru stocarea și modelarea polinoamelor am folosit structuri de tipul *Map*, iar pentru a face posibilă crearea interfeței am folosit *javax.swing*.
- De asemenea, am creat și o parte de testare folosind Junit. Aceasta fiind reprezentată de clasa *PolynomialTest*.



4. Implementare

Descrierea principalelor clase, cu metodele aferente.

Clasa *Polynomial*: metode pentru crearea polinoamelor și implementarea operațiilor.

Metoda *createPolynomial()* constă în crearea polinoamelor bazându-se pe un regex după care string-urile (polinoamele) introduse de utilizator vor fi convertite în Map-uri care vor reține gradul și coeficientul corespunzător, facilitând astfel implementarea operațiilor.

Metodele *add()*, *subtract()*, *multiply()*, *derivation()*, *integration()*, *division()* se bazează pe gestionarea Map-urilor în care sunt reținute polinoamele. Map-urile sunt accesate cu *foreach*, iar puterile și coeficienții aferenți vor fi manipulați în vederea obținerii rezultatelor solicitate.

De asemenea, o metodă esențială care trebuie amintită este *toString()* prin intermediul căreia am făcut posibilă afișarea rezultatelor sub forma de polinom.

Clasa *PolynomialCalculator*

Această clasă implementează interfața grafică. Aici am proiectat label-uri și butoane aferente fiecărei operații folosindu-mă de *javax.swing*. Tot în această clasă am creat *ActionListen*-eri anonimi pentru fiecare buton (operație).

Clasa *PolynomialTest*

Am creat această clasă pentru a testa operațiile folosind *JUnit*. Am implementat câte o metodă pentru testarea creării polinoamelor *setUp()*, respectiv pentru operații *testAddition()*, *testSubtraction()*, *testMultiplication()*, *testDerivation()*, *testIntegration()*.

5. Rezultate

Pentru testarea operațiilor am prezentat scenariul de creare a 2 polinoame " $3x^2 + 2x + 1$ " și " $x^2 + x$ ". Folosindu-mă de *assertEquals* am testat cazul prin ceea ce se așteaptă a fi rezultat:

-adunare- va aștepta " $1+3x+4x^2$ "

-scadere- se va aștepta „ $1+1x+2x^2$ "

-înmulțire- se va aștepta " $1x+3x^2+5x^3+3x^4$ "

-derivare- se va aștepta " $2+6x$ " pentru primul și " $1+2x$ " pentru al doilea

-integrare- se va aștepta " $1x+x^2+x^3$ " pentru primul și " $0.5x^2+0.3x^3$ " pentru al doilea.

6. Concluzii

Această temă mi-a oferit oportunitatea de a-mi accentua competențele OOP prin crearea aplicației interactive cu un posibil utilizator, dar și prin implementarea conceptului de operații cu polinoame din matematică.

O posibilă dezvoltare ulterioară ar fi proiectarea unei interfețe mai prietenoasă și estetică. De asemenea, introducerea mai multor concepte matematice.

7. Bibliografie

1. https://users.utcluj.ro/~igiosan/teaching_poo.html
2. <https://stackoverflow.com/questions/36490757/regex-for-polynomial-expression>
3. <https://www.youtube.com/>