DOCUMENTATIE

TEMA *1*

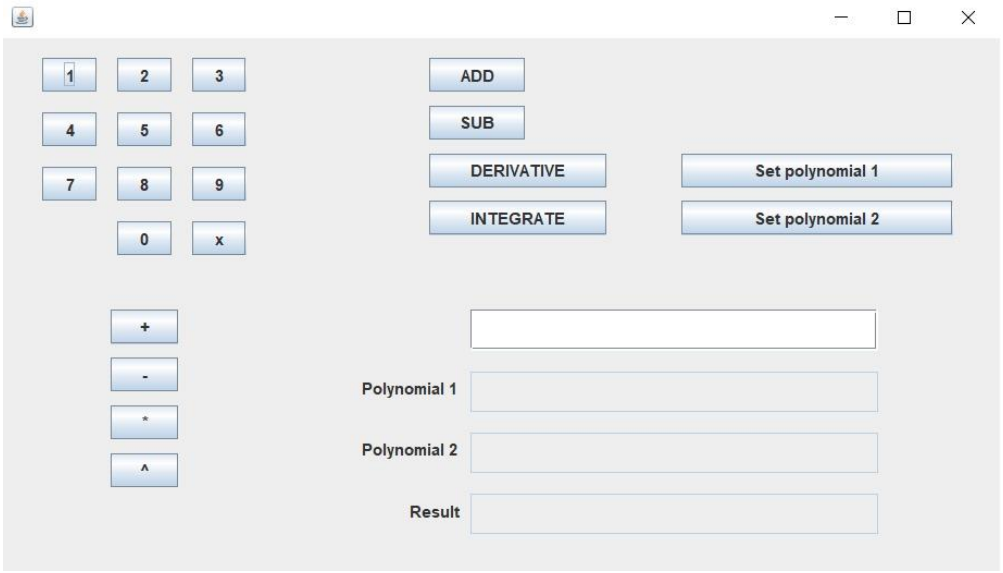
NUME STUDENT: TODOR ANDREI-PETRU
GRUPA: 30227

CUPRINS

1. Obiectivul temei

Obiectivul principal al acestei teme este realizarea unui calculator de polinoame, care să realizeze operații precum adunarea a două polinoame, scăderea lor, integrarea sau derivarea primului polinom.

Calculatorul dispune de o interfață grafică prin care utilizatorul poate să introducă două polinoame și apoi să realizeze operația dorită.



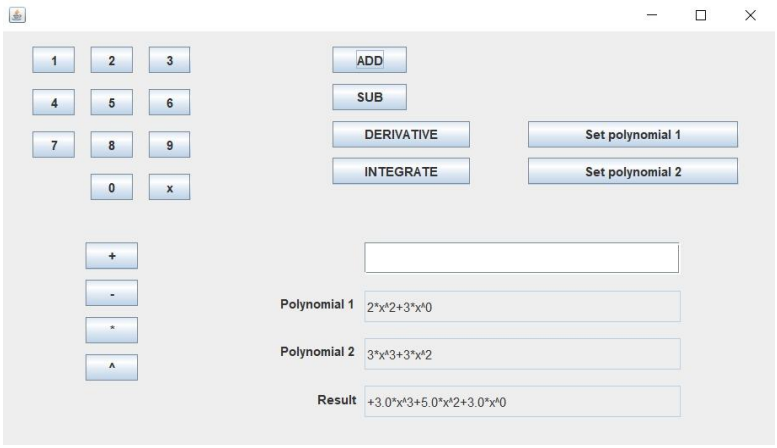
1. Analiza problemei, modelare, scenarii, cazuri de utilizare

*Calculatorul poate să realizeze operațiile de adunare, scădere, integrare și derivare asupra polinoamelor. Polinoamele sunt reprezentate printr-o listă de monoame, de forma a*x^b, unde a este coeficientul monomului, iar b este gradul acestuia.*

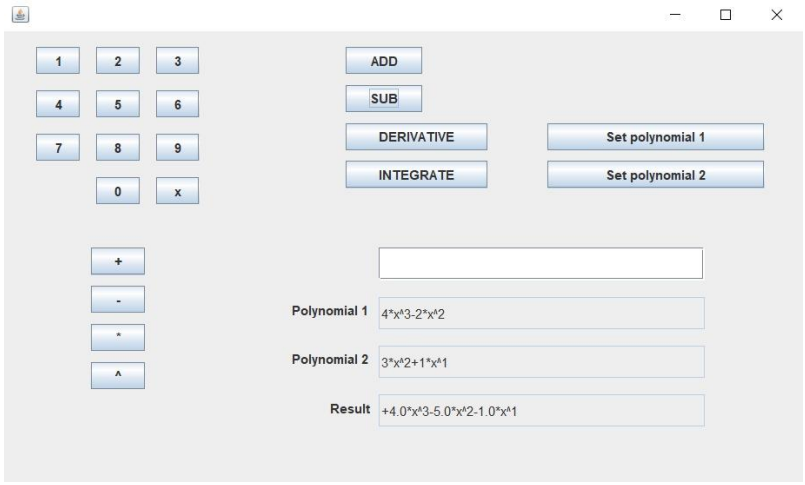
Interfața grafică este ușor de folosit, având butoane cu denumiri suggestive (cifrele de la 0-9, butonul "x", butoanele "+", "-", "", "^" pentru scrierea polinomului, butoanele "ADD", "SUB", "DERIVATIVE", "INTEGRATE" pentru operații și butoanele "Set polynomial 1", "Set polynomial 2" pentru setarea polinoamelor 1, respectiv 2). Utilizatorul poate să introducă polinomul atât de la tastatură, cât și apăsând pe butoane.*

Utilizatorul va introduce, pe rând, câte un polinom în prima casetă text goală. Prin apăsarea butoanelor "Set polynomial 1", "Set polynomial 2", polinoamele dorite vor fi plasate în casetele "Polynomial 1", respectiv "Polynomial 2". Următorul pas este alegerea unei operații, prin apăsarea unuia din butoanele "ADD", "SUB", "DERIVATIVE", "INTEGRATE". Rezultatul va fi afișat în câmpul "Result".

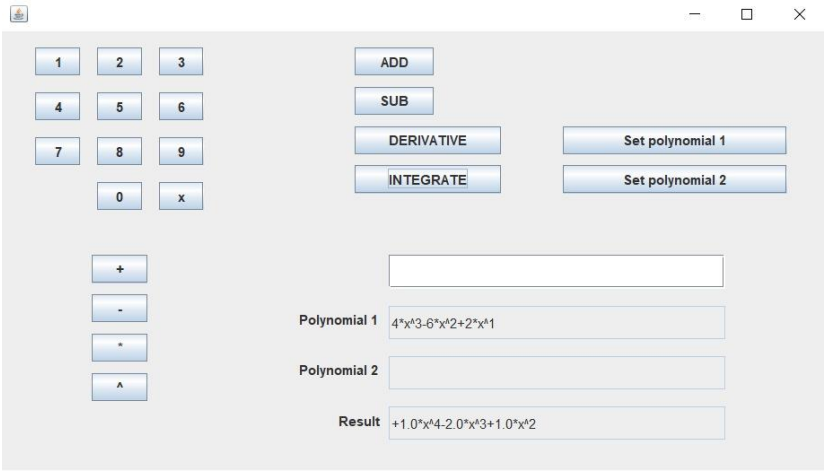
ADUNAREA A DOUĂ POLINOAME



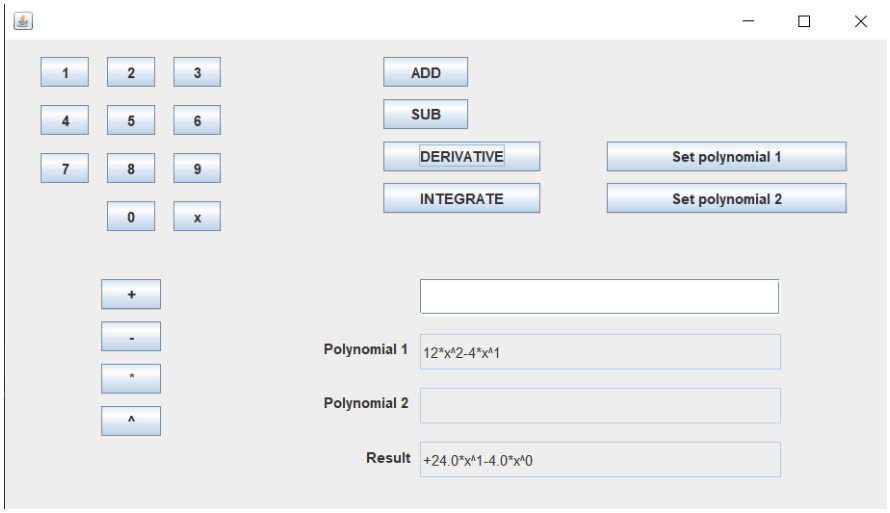
SCĂDEREA A DOUĂ POLINOAME



INTEGRAREA UNUI POLINOM



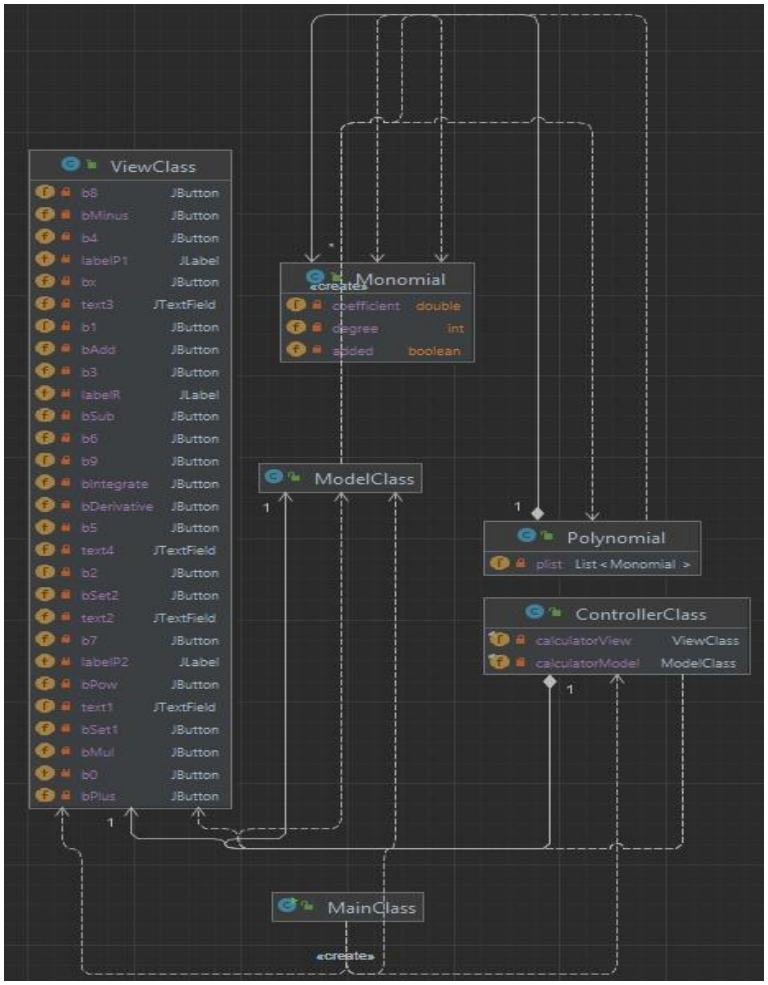
DERIVAREA UNUI POLINOM



2. Proiectare

UML este un limbaj ce descrie modele și specificații pentru software. A fost dezvoltat pentru reprezentarea complexității programelor orientate pe obiect, al căror fundament este structurarea pe clase.

Diagrama UML a calculatorului de polinoame este formată din 6 clase: Monomial, Polynomial, ModelClass, ViewClass, ControllerClass și MainClass. Legăturile dintre acestea sunt prezentate mai jos:



Model-View-Controller (MVC) este un model arhitectural utilizat în ingineria software. Succesul modelului se datorează izolării logicii de business față de considerentele interfeței cu utilizatorul, rezultând o aplicație unde aspectul vizual și nivelele inferioare ale regulilor de business sunt mai ușor de modificat, fără a afecta alte nivele.

Calculatorul are la bază o arhitectură de tip MVC. Astfel, clasa ControllerClass devine principala componentă. Aceasta realizează conexiunile dintre clasele ModelClass și ViewClass. Modelul este în permanență actualizat prin intermediul controllerului, căruia îi transmite datele. Totodată, view-ul este actualizat în permanență de controller, căruia îi transmite informații despre acțiunile utilizatorului.

Modelul controlează operațiile de utilizare a informației (trimise dinainte de către rangul său superior) pentru a rezulta o formă mai ușoară de înțeles.

View-ul corespunde reprezentării grafice, sau mai bine zis, exprimării formei datelor. Acesta este reprezentat de interfața grafică ce interacționează cu utilizatorul final. Rolul său este de a evidenția informația obținută până ce ea ajunge la Controller.

Controllerul are rolul de a controla întreaga aplicație. Se pot controla fișiere, scripturi, programe (în general cam orice tip de informație permisă de interfață). În acest fel putem diversifica conținutul nostru de o formă dinamică și statică, în același timp.

3. Implementare

Calculatorul este construit din 6 clase: Monomial, Polynomial, ModelClass, ViewClass, ControllerClass și MainClass, fiecare având metodele specifice.

-Clasa Monomial

Implementează caracteristicile specific unui monom: grad și coefficient. Variabilele sunt de tip private (pot fi văzute doar în interiorul clasei).

Metode:

- *public Monomial(int degree, double coefficient) – constructorul clasei;*
- *public int getDegree() – returnează gradul monomului;*
- *public voi setDegree(int degree) – setează gradul monomului;*
- *public double getCoefficient() – returnează coeficientul monomului;*
- *public void setCoefficient(double coefficient) – setează coeficientul monomului;*
- *public boolean isAdded() – verifică dacă monomul a fost adăugat în polinomul final.*

-Clasa Polynomial

Un polinom este alcătuit dintr-o listă de monoame, listă de tip private.

Metode:

- *public Polynomial(List<Monomial> plist) – constructorul clasei;*
- *public List<Monomial> getList() – returnează lista de monoame ale polinomului.*

-Clasa ModelClass

Conține metodele necesare pentru implementarea calculatorului.

Metode:

- *public Polynomial toPolynomial(String str) – transformă un șir de caractere în polinom;*
- *public String toString(Polynomial polinom) – transformă un polinom în șir de caractere;*
- *public Polynomial Derivative(Polynomial polinom) – realizează derivarea unui polinom primit ca parametru. Rezultatul returnat este tot un polinom.*
- *public Polynomial Integrate(Polynomial polinom) – realizează integrarea unui polinom;*
- *public Poluynomial Add(Polynomial polinom1, Polynomial polinom2) – realizează adunarea a două polinoame;*
- *public Poluynomial Sub(Polynomial polinom1, Polynomial polinom2) – realizează scăderea a două polinoame;*

-Clasa ViewClass

Această clasă implementează interfața grafică a calculatorului. Astfel, aceasta are ca și parametri butoanele, textfield-urile, label-urile și panel-ul interfeței grafice. Butoanele au denumiri sugestive, implementând totodată și metode de ActionListener pentru evenimentele de apăsare a butonului. Butoanele sunt de mai multe feluri: butonul x (inserează în caseta polinomului textul x), butoanele de cifre (inserează în caseta polinomului cifrele de la 0 la 9), butoanele de operatori (inserează în caseta polinomului operatorii +-^), butoanele pentru operații pe polinoame (realizează operația dorită asupra polinoamelor).*

Pe lângă avcestea, mai sunt prezente 4 TextField-uri goale. Utilizatorul va introduce, pe rând, câte un polinom în prima casetă text goală. Prin apăsarea butoanelor "Set polynomial 1", "Set polynomial 2", polinoamele dorite vor fi plasate în casetele "Polynomial 1", respectiv "Polynomial 2". Următorul pas este alegerea unei operații, prin apăsarea unuia din butoanele "ADD", "SUB", "DERIVATIVE", "INTEGRATE". Rezultatul va fi afișat în câmpul "Result".

-Clasa ControllerClass

Această clasă realizează conexiunea dintre clasele ViewClass și ModelClass. Principalele componente ale acestei clase sunt două obiecte de private, unul de tip View și unul de tip Model. Obiectul de tip View este răspunzător pentru interfața grafică, în timp ce obiectul de tip Model este răspunzător pentru operațiile realizate. Obiectul de tip model este în permanență actualizat prin intermediul clasei ControllerClass, căreia îi transmite datele. Obiectul de tip View este actualizat în permanență prin intermediul clasei Controller, căreia îi transmite informații despre acțiunile utilizatorului. În cadrul acestei clase sunt definite răspunsurile la acțiunile de click ale utilizatorului. Astfel, conform butonului apăsat de utilizator se va realiza operația dorită.

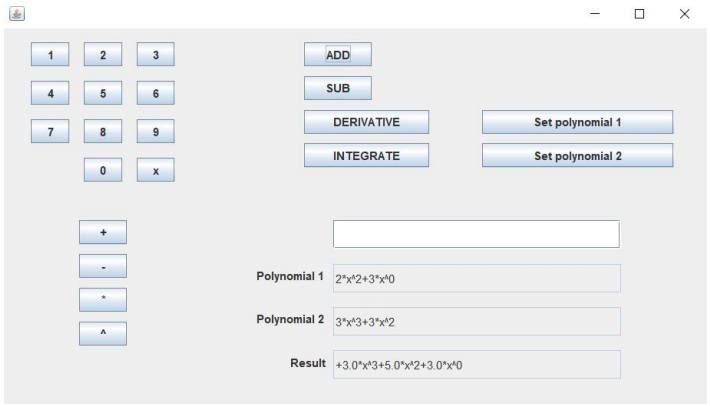
-Clasa MainClass

Această clasă este implementarea propriu zisă a calculatorului. Clasa conține o metodă statică main care creează trei obiecte: un obiect de tip Mode, un obiect de tip View și un obiect de tip Controller.

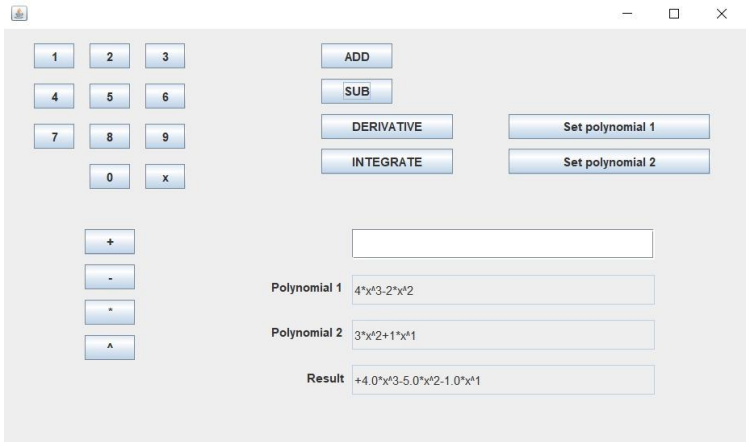
4. Rezultate

S-au testat operațiile pe care le poate realiza calculatorul.

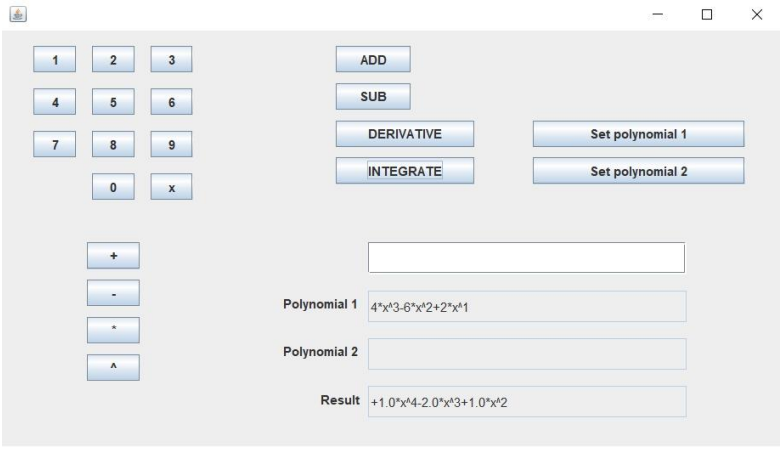
ADUNARE



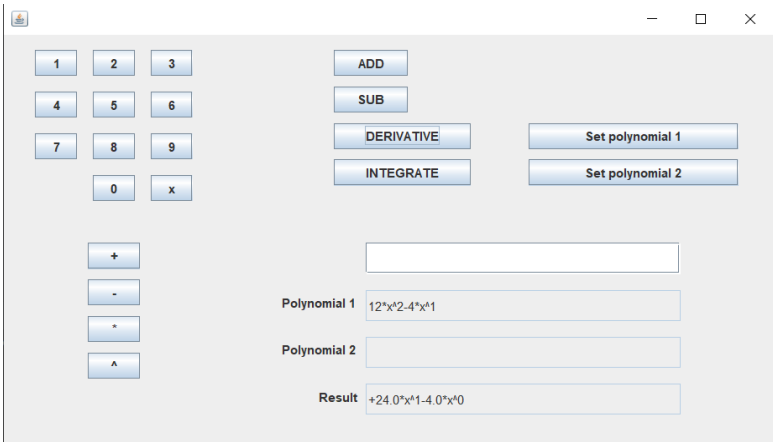
SCĂDERE



INTEGRARE



DERIVARE



5. Concluzii

Am reușit să aprofundez paradigma programării pe obiecte. Mi-am dezvoltat abilitățile de lucru cu liste, cu interfața grafică și cu structura de tip MVC.

Calculatorul poate să fie îmbunătățit. Interfața poate fi modernizată, ar putea fi introduce noi operații, iar coeficienții ar putea fi reali și gradele întregi.

6. Bibliografie

https://www.geeksforgeeks.org/java/
https://www.w3schools.com/
https://stackoverflow.com/
https://study.com/
https://www.tutorialspoint.com/index.htm