**Tema 1.Calculator**  **de polinoame**

**Documentație**

Nemțanu Alexandru-Vasile

Grupa 30227

Cuprins

1. Obiectiv

2. Analiza problemei. Cazuri de utilizare

3.Proiectare si Implementare

3.1. Diagrama UML

3.2. Pachete si Clase

4.Concluzii

5.Bibliografie

**1.Obiectiv**

Obiectivul acestei teme este de a dezvolta o aplicatie cu interfata utilizator ce va indeplini rolul unui calculator ce realizeaza operatii cu polinoame.

Pentru a indeplini obiectivul principal, trebuie indeplionite urmatoarele obiective secundare: realizarea interfetei utilizator,implementarea unui mod de transformare a polinoamelor introduse ca tip String in date cu care aplicatia sa poata opera pentru a calcula un rezulat corect, implementarea operatiilor specificate pe polinoame (adunare, scadere, impartire, inmultire, derivare, integrare), debugging si testarea rezultatelor cu JUnit.

**2.Analiza problemei**

Cadrul de cerinte este implementarea unei aplicatii care sa se comporte ca un calculator de polinoame si sa implementeze operatiile de adunare,scadere,inmultire,impartire,derivare si integrare.

Utilizarea aplicatiei se face prin intermediul interfetei utilizator, introducerea polinoamelor fiind posibila doar in casetele de text predefinite. Selectarea operatiei dorite se face prin unul din cele 4 butoane prezente in interfata. Daca input-ul este gresit va aparea un mesaj de atentionare. Rezultatul este afisat in caseta text marcata cu antetul “result”.

**Cazuri de utilizare:**

⦁ Se deschide aplicatia

⦁ Se ruleaza metoda main din clasa Controller aflata in pachetul Controller

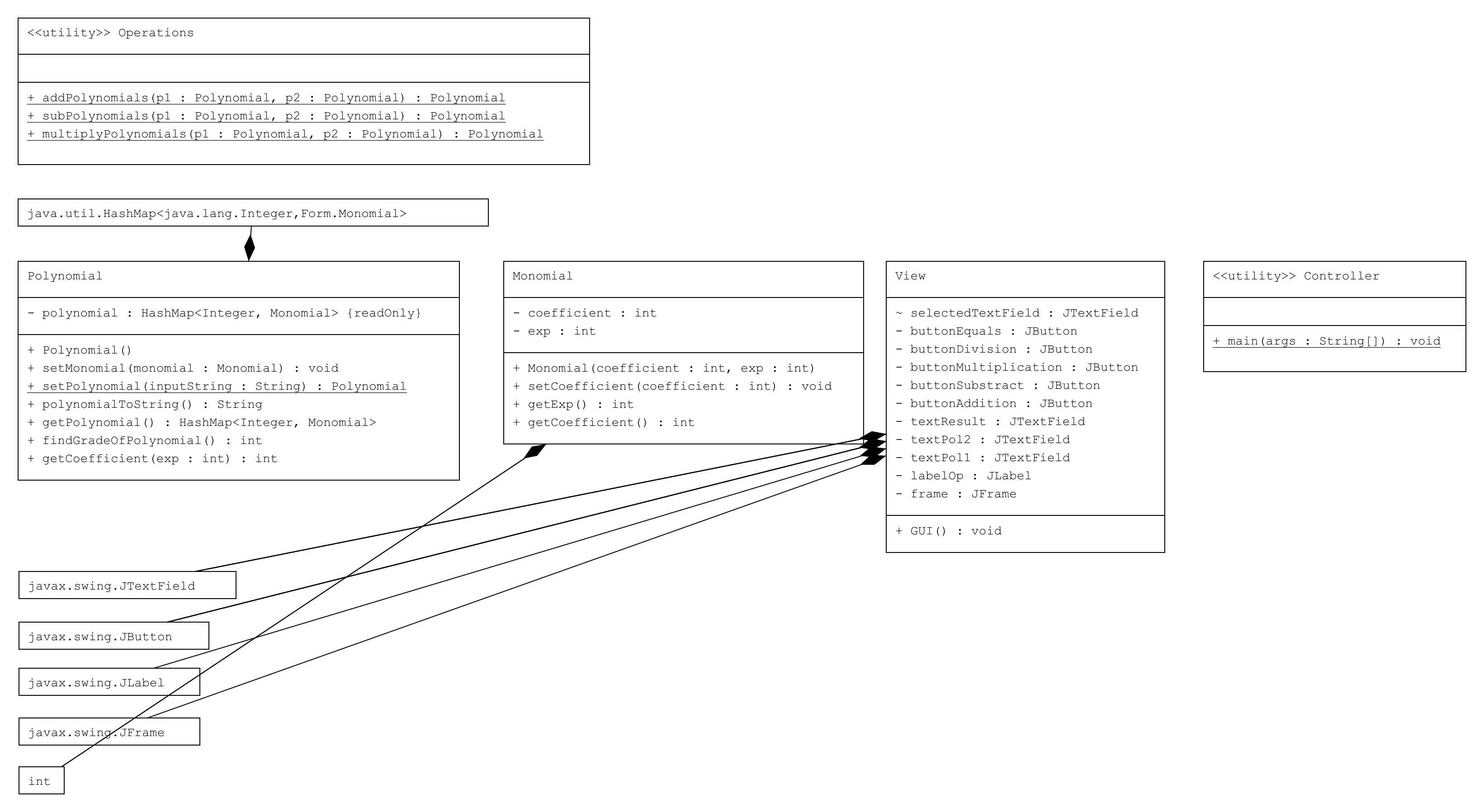
⦁ Se introduce primul polinom in primul textField si al doilea polinom in al doilea textField

⦁ Se apasa pe butonul de “+” pentru adunare, “-” pentru scadere, “x” pentru inmultire, / pentru impartire

⦁ Rezultatul va fi afisat in cel de-al treilea textField numit “result”.

**3.Proiectare si Implementare**

**3.1.Diagrama UML:**



**3.2.Pachete si Clase**

Pentru a implementa corect din punct de vedere matematic un polinom in viziunea OOP, am impartit polinomul ca o multime de monoame.

Aceasta tema este proiectata in 5 clase cuprinse in 4 pachete:

**Pachetul Form:**

1)Clasa **Monomial**: definește un obiect monom care este utilizat în algebra polinomială. Monomul are un coeficient și o putere asociată, care determină cum se adună sau se scad polinoamele. Constructorul primește coeficientul și puterea monomului și inițializează aceste câmpuri ale obiectului. Clasa are și două metode de acces pentru a returna coeficientul și puterea monomului, precum și o metodă setter pentru coeficient.

2)Clasa **Polynomial**: această clasă implementează un polinom matematic. Clasa contine un map, denumit “polynomial”, care stochează monoamele din polinom, unde cheia este exponentul și valoarea este un obiect de tip Monomial. Fiecare Monomial este reprezentat de un coeficient și un exponent. Clasa Polynomial conține următoarele metode:

* constructorul clasei Polynomial(), care initializează map-ul 'polynomial'.
* metoda setMonomial(Monomial monomial), care adaugă un Monomial la polinom. Dacă polinomul conține deja un monomial cu același exponent, atunci coeficientul noului monomial si se va aduna cu coeficientul monomial-ului existent.
* metoda statică setPolynomial(String inputString), care primește ca argument un șir de caractere reprezentând un polinom și îl convertește într-un obiect de tip Polynomial. În această metodă se utilizează expresii regulate pentru a identifica fiecare monom din polinom și se creează un obiect Monomial pentru fiecare monom găsit.
* metoda polynomialToString(), care convertește polinomul într-un șir de caractere.
* metoda getPolynomial(), care returnează map-ul 'polynomial'.
* metoda findGradeOfPolynomial(), care returnează gradul maxim al polinomului.
* metoda getCoefficient(int exp), care returnează coeficientul monomului cu un anumit exponent dat ca argument.

**Pachetul Operations:**

1. Clasa **Operations**: Această clasă definește trei metode statice pentru efectuarea operațiilor cu polinoame:

* addPolynomials(Polynomial p1, Polynomial p2): adună două polinoame și returnează un nou polinom pentru suma lor;
* subPolynomials(Polynomial p1, Polynomial p2): scade două polinoame și returnează un nou polinom pentru diferenta lor;
* multiplyPolynomials(Polynomial p1, Polynomial p2): înmulțește două polinoame și returnează un nou polinom pentru produsul lor.

În fiecare metodă, se folosește obiectul Polynomial pentru a păstra polinoamele, fiecare reprezentat sub forma unui dicționar de monomiale. Dicționarul stochează fiecare monomial sub forma unei chei ce reprezintă exponentul monomului și valoarea corespunzătoare cheii, ce reprezinta obiectul Monomial care stochează coeficientul și exponentul monomului.

În metoda addPolynomials se iterează prin fiecare monom din p1, iar dacă p2 conține deja un monom cu același exponent, coeficientul lor este adunat. In cazul in care coeficientul este diferit de zero, se creează un nou monom pentru rezultatul final.

În metoda subPolynomials, se iterează prin fiecare monom din p1, iar dacă p2 conține deja un monom cu același exponent, coeficientul lor este scăzut. In cazul in care coeficientul este diferit de zero, se creează un nou monom pentru rezultatul final.

În metoda multiplyPolynomials, se iterează prin fiecare monom din p1 și p2, se calculează produsul lor și se adaugă rezultatul în dicționarul rezultatului. Dacă rezultatul final conține deja un monom cu același exponent, coeficientul lor este adunat. In cazul in care coeficientul este diferit de zero, se creează un nou monom pentru rezultatul final.

**Pachetul View:**

1. Clasa **View**:

Aceasta clasa construieste interfata grafica a aplicatiei. Aceasta utilizeaza obiectele din pachetele "Form" si "Operations" pentru a efectua operatiile de adunare, scadere si inmultire a celor doua polinoame.

Interfata grafica consta dintr-o fereastra cu denumirea "Polynomial Calculator", ce are 2 campuri de text pentru introducerea polinoamelor, un camp de text pentru afisarea rezultatului si butoane pentru operatiile de adunare, scadere, inmultire si impartire a celor doua polinoame.

De asemenea, metoda "GUI()" defineste actiunea fiecarui buton in parte. Mai exact, atunci cand se apasa butonul de adunare, se va apela metoda "addPolynomials()" din clasa "Operations", cand se va apasa butonul de scadere, se va apela metoda "substractPolynomials()" iar cand se va apasa butonul de inmultire se va apela metoda "multiplyPolynomials()".

**Pachetul Controller:**

1. Clasa **Controller**: aceasta clasa se ocupă de partea de control a aplicației și creează interfața grafică pentru utilizator prin apelarea metodei "GUI()" din clasa "View".

**4.Concluzii**

Acest proiect a dus la o serie de beneficii pentru dezvoltarea competențelor mele. Am învățat cum să validez un anumit format pentru intrarea de date (în cazul nostru, polinoame introduse de la tastatură), cum să manipulez colecțiile de date într-un mod mai eficient (cum ar fi iterația, adăugarea și actualizarea elementelor) si probabil abilitatile cele mai solicitate au fost cea de a face debugging si cea de gandire matematica si de vedere logica asupra programarii pe notiuni matematice.

Ca dezvoltari ulterioare, ar trebui implementate functia pentru impartire si functionarea calculatorului cu coeficienti reali si complexi.

**5.Bibliografie**

[**https://www.programiz.com/java-programming/library/hashmap/foreach**](https://www.programiz.com/java-programming/library/hashmap/foreach)

https://www.w3schools.com/java/java\_hashmap.asp

[**https://www.javatpoint.com/java-swing**](https://www.javatpoint.com/java-swing)

[**https://www.w3schools.com/java/java\_regex.asp**](https://www.w3schools.com/java/java_regex.asp)