DOCUMENTATIE

TEMA *1*

NUME STUDENT : COJOCARU ANA CATALINA

GRUPA: 30226

CUPRINS

1. Obiectivul temei

Obiectivul temei a fost crearea unui calculator de polinoame uşor de folosit de catre utilizator care poate efectua următoarele operaţii*: Adunare, Scadere, Inmultire, Impartire, Derivare, Integrare.*

• Crearea interfeţei grafice	*Cap. 4*
• Crearea clasei polinom	*Cap. 4*
• Stocarea valorilor date în interfaţa grafică in HashMap	*Cap. 4*
• Efectuarea operaţiilor	*Cap. 5*
• Verificarea corectitudinii	*Cap. 5*
• Utilizarea de către utilizator	*Cap. 2*

-Crearea interfeţei grafice: Am folosit Java Swing într-o clasa care implementeaza Action Listener;

-Crearea clasei polinom: Clasa a fost creată cu un singur parametru, HashMap-ul care are două valori intregi respective puterii si coeficientului;

-Stocarea valorilor date in interfaţa grafică in HashMap: se vor stoca cu ajutorul unui regex ce desparte string-ul in monoame;

-Efectuarea operaţiilor: Într-o clasă separata numită *Operaţii*, am implementat static cele 6 operaţii

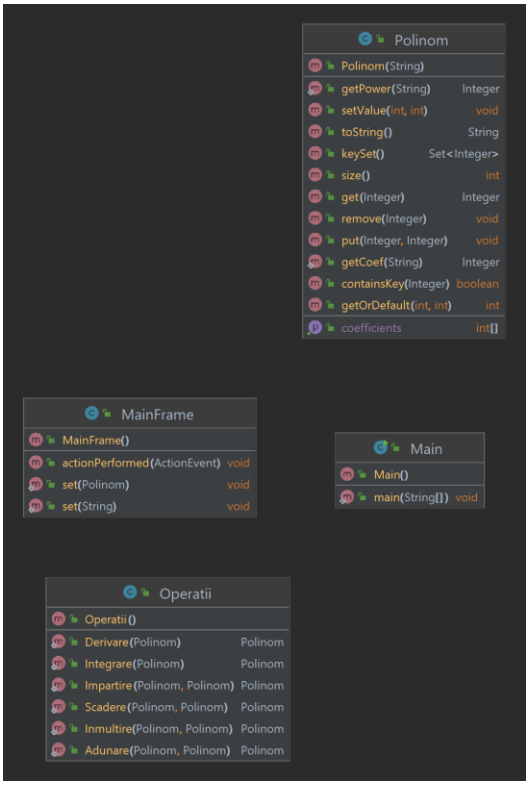
-Verificarea corectitudinii: Folosind testarea unitară, JUnit, am testat cele sase operatii create

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

Calculatorul e functional in toate cazurile.

Brief Description	Calculatorul primeste două polinoame si se doreste calculul unor operatii de baza.
Actors(s)	The user
Trigger	Apasând butonoanele cu titlul operaţiilor dorite, se va efectua operatia.
Pre-conditions	Calculatorul nu afiseaza nimic, toate campurile fiind goale.
Post-Conditions	In aria alocata rezultatului se va afisa polinomul rezultat in urma efectuarii operatiilor.
Basic Flow	*Step 1:* Introduceţi polinomul 1; *Step 2:* Introduceţi polinomul 2; *Step 3:* Apasati butonul cu operatia dorita; *Step 4:* Apasati butonul ‘Clear’ pentru a introduce alte date.
Exception Flow	Impărţirea cu 0 nu e posibila, aceasta va rezulta intr-un mesaj în casuta de Rezultat.

3. Proiectare



In proiect exista mai multe pachete: pachetul *GUI* in care gasim functia *Main* care creeaza un obiect din clasa *MainFrame* (interfata cu utilizatorul), si pachetul *Polinom* care are clasa *Operatii* unde se implementeaza *static* operatiile si clasa *Polinom*. De asemenea, testarile se fac intr-o clasa *Test* numita *OperatiiTest*.

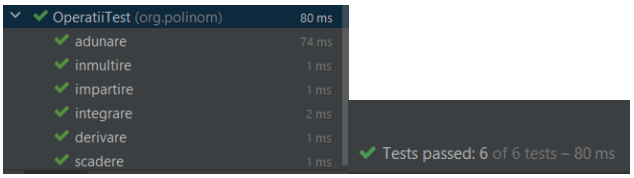
Pentru stocarea polinoamelor se foloseste *HashMap<Integer, Integer>* (coeficientii se vor lua intregi in cazul tuturor operatiilor).

4. Implementare

Clasa *Main* instanţiaza un obiect din clasa *MainFrame* care va crea interfata grafica pentru utilizator. Aceasta are doua campuri unde se introduc cele doua polinoame ce vor participa la operatiile de adunare, scadere, inmultire si impartire. Primul polinom va putea fi derivat, respectiv integrat. Interfata este simpla, are 6 butoane specific fiecarei operatii si un buton de stergere. Campul de rezultat nu poate fi editat de catre utilizator. Cele doua siruri de caractere care se vor introduce de utilizator in campurile pentru polinoame vor fi despartite in monoame printr-un regex care se gaseste in constructorul clasei *Polinom* care primeste ca parametru string-ul. Prin metodele implementate in clasa *Polinom*, fiecare monom e stocat in

HashMap printr-o pereche de valori respective gradului polinomului (cheia) si coeficientul acestuia (valoarea). ///Metoda de gasire a coeficientului monomului ia primele cifre gasite inaintea literei ”x”, memorand semnul numarului. Gasirea puterii este similara: gaseste ultimele cifre inaintea semnului ”^” (intr-un bloc de try-catch se verifica daca exista puterea numarului sau polinomul e de grad 1, respective numar). Se adauga valorile in *HashMap*-ul polinomului. In clasa *Operatii* se implementeaza static toate operatiile. *Adunarea* parcurge elementele primului polinom, iar daca al doilea polinom continue aceeasi putere aduna coeficientii adaugandu-i la rezultat, stergand in cel de-al doilea polinom cheile gasite. La sfarsit se parcurg elementele ramase din al doilea polinom. Similar la *Scadere*. *Inmultirea* parcurge cu doua *foreach*-uri cele doua polinoame, inmultindu-le si adaugand la rezultat numerele obtinute, iar Impartirea imparte cele doua polinoame, restul fiind aflat prin scaderea din primul polinom a produsului catului cu al doilea polinom. Derivarea si integrarea au ca parametru doar un polinom.

5. Rezultate



Testarea s-a facut cu *Junit* pe fiecare operatie in parte cu acelasi set de polinoame, toate operatiile fiind corecte, fara erori. Fiecare operatie a fost testata intr-o metoda separata in clasa de test. Rezultatele asteptate au coincis cu rezultatele calculate de metodele implementate in *clasa Operatii*.

6. Concluzii

Am invatat sa structurez un proiect mai mare decat cele precedente. De asemenea, am invatat sa , sa lucrez cu *HashMaps*, sa imi organizez clasele si metodele pentru o intelegere usoara a codului. O posibila dezvoltare a proiectului este definirea coeficientilor de tip *Double* si extinderea posibilitatilor cu alte operatii cum ar fi Laplace, inlocuirea variabilei x cu un numar anume etc. .

7. Bibliografie

1. *Bruce Eckel, Thinking in Java (4th Edition), Publisher: Prentice Hall PTRUpper Saddle River, NJUnited States, ISBN:978-0-13-187248-6 Published:01 December 2005.*
2. *StackOverflow*
3. *https://users.utcluj.ro/~igiosan/teaching_poo.html*
4. *Fundamental Programming Techniques (dsrl.eu)*