**DOCUMENTAȚIE**

**CALCULATOR DE POLINOAME**

**TEMA 1**

NUME STUDENT: **ADELIN MOLDOVAN**

GRUPA:**302210**

**CUPRINS**

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

**1.OBIECTIVUL TEMEI**

Cerinta acestei teme este implementarea unui calculator de polinoame cu ajutorul caruia, un utilizator din exterior poate folosi calculatorul sub forma unei intefete grafice pentru a efectua operatii asupra unor polinoane de diferite grade. Operatiile pe care acest calculator le va putea face sunt: adunarea, scaderea, inmultirea si imparirea a doua polinoame. Pe langa acestea va mai putea face derivarea, respective inegrarea unui dintrea cele doua polinoame. Utilizatorul va avea doua casete in care va introduce de la tastatura cele doua polinoame. Pe langa acestea mai este o casuta unde va aparea rezultatul operatiilor. Si in interfata grafica mai apar si 6 butoane care reprezinta cele 6 operatii cerute. Acestea se apasa dupa introducerea datelor polinoamelor. De retinut polinoamele trebui introduce sub forma: ax^n, unde a reprezinta coeficintul monomului, iar este puterea. Rezultatul va fi afisat sub aceasi forma cum sunt introduce polinoamele.

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

Pentru implementarea calculatorului este obligatoriu sa se implementeze o interfata grafica pentru a facilita utilizarea calculaatorului si pentru al face mult mai usor de utilizat. Acesta putand fi utilizat de oricine fara a fi nevoie de cunostinte de programare sau matematica. Calcualtorul face operatiile fara a avea nevoie de ajutor in plus din exterior. Campurile din interfata grafica poarta o denumire specifica pentru a face mai usoara intelegerea acestuia. De exemplu “adunare +” pentru operatia de adunare, “scadere –“ pentru operatia de scadere, etc. Astfel polinoamele se introduce in doua TextFiled-uri, de asemenea si rezultat va fi afisat tot intr-un text field. Iar operatiile sunt realizate sub forma de simple butoane.

**3. Proiectare**

In cadrul proiectarii pentru a utiliza memoria mai eficient si pentru a administra mai usoar elementele pe care le folosim in cadrul proiectului folosim clasa ArrayList. Astfel polinomul va fi o lista de monoame, pentru a accesa fiecare element mai usor. Adica fiecare element din polinom este un monom iar acestea impreuna formeaza un ArrayList.

Clasele cele mai importante din proict sunt View si Controller, pe langa acestea mai avem clasa Monom unde definim un monom, si clasa Polinom care este definite ca un sir de monoame. Astfel clasa polinom are un sigur parametru, acela fiind lista de monoame. Si mai avem si clasa RezultatImpartire unde avem doua argumente de tip polinom folosite pentru definirea catului si restului in cadrul impartirii care trebuie sa returneze doua valori. Iar dupa cum stim, nu este posibil sa returnam doua valori diferinte. Asa ca cele doua valori sunt definite intr-o clasa separate si mai apoi utilizare in clasa Operatii unde sunt definite cele 6 operatii aplicate asupra polinoamelor.

**4. Implementare**

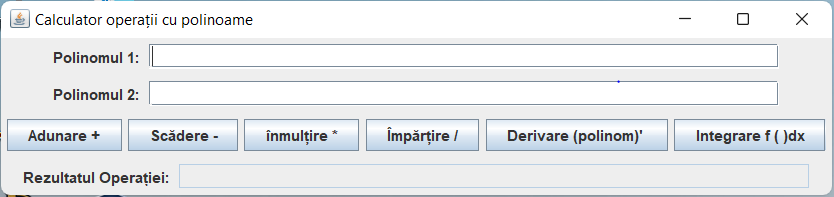
In proiect sunt 7 clase si anume: Monom, Polinom, View, Controller, Main, RezultatImpartire si Operatii. Fiecare clasa are rolul ei si este necesara rezolvarii celor 6 operatii si de asemenea pentru realizarea interfectei grafice.

Clasa “Main” este utilizata drept “mail-ul” programului unde se ruleaza codul si care va determina vizualizarea interfetei grafice unde vor fi introduse date si unde se vor vizualiza cele 6 operatii.

Clasa “Monom” are 2 campuri: coeficintul, care dupa numele specific reprezinta coeficientul fiecarul monom, si putearea, care reprezinta puterea/gradul unui singur monom. Pe langa acestea am realizat gettere si settere folosind “alt+insert” care este o functie predefinita in Intellij.Avem si metoda “compareTo” cu care aranjam un polinom in fucntie de gradul fiecarui monom. Si pe langa asta mai avem si metoda si contructorul “toString” implementat manual pentru a putea afisa in mode eficient si in mod specific fiecare monom. In aceasta metoda am verificat toate cazurile pozibile, adica coeficinetul poate avea una din valorile -1, 0, 1.

Clasa “Polinom” este clasa in care am implementat un polinom definit ca o lista de monoame(“ArrayList”). Dupa cum am spui mai sus am facut asta pentru a fi mai eficient programul din punct de vedere al memoriei si pentru a fi mai usor de reprezentat polinoamele in interfatagraficai si in cod in sine. In acesta clasa este implementat constructorul. Acesta este un sir fara niciul element in el, fiind doar initializat in aceasta clasa si pe urma populat. Avem o metoda de inserare a unui monom in lista respective(insertMonom). Avem si metoda inmultireCuMonom care face ceea ce ii zice si numele, si care este folosita in cadrul operatiei de inmultire. Si mai avem si metoda “toString” folosita pentru reprezentarea frumoasa a unui polinom.

Clasa “RezultatImpartire” este folosita pentru a realiza operatia de impartire a unor doua polinoame si pentru a afisa atat catul cat si restul in cadrul unei impartiri. Aceasta contine catul si restul si setter si gettere realizare prin functia integrate in Intellij de generearea.

Clasa “View” este clasa unde am realizat interfata grafica care arata ca si mai jos: 

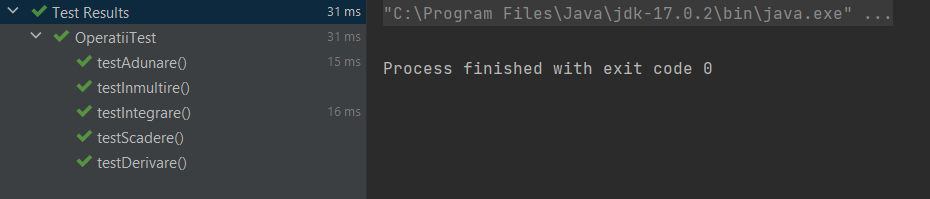
In aceasta sunt realizate 4 paneluri secundare care vor fi intrduse orizonat in panelul principal. Primele doua reprezinta spatial unde se introduce cele doua polioame reprezentate prin “TestField-uri”, Al treiliea panel avea cele 6 butoane representative pentru cele 6 operatii. Dupa cum se vede mai sus avem butonul “Adunare +” pentru operatia de adunare, “Scadere –“ pentru operatia de scadere dintre cele doua polinoame, “Inmultire \*” pentru operatia de inmultire, “Impartire /” pentru operatia de impartire, “Derivare (polinom)’ “ pentru operatia de derivare a primului polinom si ultimul buton “Integrare f ( ) dx” folosit pentru operatia de integrare a primului polinom introdus in primul camp. In ultimul TestField este adaugat label-ul denumit “Rezultat Operatiei” unde se va afisa rezulatul operatie dupa apasarea unuia dintre cele 6 butoane, fiecare dintre acestea are un ActionListener care este implentat in clasa Controller. Aici se va afisa polinomul rezultat in urma uneia dintre operatii reprezentat exact ca si cele introduse sub forma unui ArrayList alcatuit din mai multe monoame de diferite grad si coeficienti diferiti. Denumirea frame-ului este “Calculator operatii cu polinoame” care este apelat in functia main.

Clasa “Controller” implementeaza ActionListener, deoare in ea descriem funcitonalitatile celor 6 butoane descrise mai sus. Astfel in aceasta clasa am facut “Override” la actionPerformed, unde pentru inceput se va stabili ce buton este selectat in interfata de cantre utilizator pentru ca mai apoi sa efectuze operatia. Pentru fiecare buton va fi prezentata o metoda in interiorul clasei unde sunt realizate cele 6 operatii la nivel de cod. Metoda cea mai importanta din clasa Controller este metoda “extrageVal” care realizeaza extractia valorilor din TextField-ul unde sunt introduse cele doua polinoame. Aceasta metoda are un paramentru care poate avea una dintre volorile “1” sau “2” care fiecare reprezinta primul, respective al doilea camp unde utilizatorul interfetei a introdus polinoamele pe care va urma sa aplice operatiile. Pentru a putea fi folosit split-ul corespunzator, string-ul trebuie prelucrat. Acest lucru inseamna adaugarea cifrei 1 in cazul in care avem coeficientul 1 sau -1, respective pentru puterea 1. Dupa ce a fost modificat corect stringul, se realizeaza split-ul acestuia. Este folosit split-ul negat, adica se precizeaza ce elemente sa fie separate(cifrele de la 0 la 9, respective semnul “-“), dar mai ales sa fie luat ca sir in cazul numerelor compuse din mai mult de 2 cifre(problema rezolvata prin folosirea in split a simbolului “+”). Dupa ce a fost separate intr-un sir de stringuri, polinomul este transformat in monoame prin inserarea pe rand a coeficientilor si a gradelor, iar mai apoi acestea sunt introduse intr-un polinom si este returnat polinomul rezultat. In ceea ce priveste actionPerformed, trebuie stabilit care din cele 6 butoane a fost apasat. In cazul in care vor fi selectate primele 4 butoane, in care e nevoie de 2 polinoame pentru realizarea operatiei, vom lua 2 Polinoame prin care citim continutul celor 2 TextFielduri editabile, iar apoi rezultatul returnat ca si polinom sau ca si RezultatImpartire in cazul impartirii, il afisam in TextFieldul needitabil fixat pentru rezultat. Analog procedam si cu butoanele de derivare si de integrare, cu exceptia ca vom citi un singur polinom de pe primul TextField needitabil.

Clasa “Operatii” Clasa **Operatii** contine toate metodele representative pentru fiecare operatie in parte. Astel in aceasta clasa sunt descrise cele 6 operatii: adunare, scadere, inmultire, impartire, derivare, integrare. Aplicate pe doua polinoame introduse de la tastatura de catre utilizator, respective primul polinom pentru operatiile de integrare si derivare. Operatia de adunare are rolul de a aduna cele doua polinoame prin parcurgerea ArrayList-ului de monoame, monom cu monom, iar mai apoi se va verifica gradul celor doua monoame, si daca este egal se vor aduna coeficientii, iar daca nu se va merge mai departe cu parcurgerea pana se va gasi gradul identic. La fel se intampla si pentru operatia de scadere doar ca coeficientii se scad in loc sa se adune. De asemenea metoda de scadere, respective cea de adunare este valabila si functionala si in cazul in care un monom are un coefieiecnt de voalare negativa. Mai departe metoda de inmultire este mai complicata. In cazul acesteia se vor lua succesiv monoamele din primul polinom si se va inmulti cu fiecare element din al doilea polinom exact cum se inmultet doua paranteze in matematica. Pe urma daca stam sa ne gandim in urma rezultatlui inmultirii se vor obtine mai multe monoame cu grad identic. Acestea for fi adunate intr-un monom nou pe care il vom si afisa in campul “Rezulatul operatie. Pentru metoda de deriare se va lua primul polinom introdus si se va parcurge tot polinomul exact ca la adunare si scadere, monom cu monm si mai apoi gradul se va scadea cu 1, iar coeficientul va fi inmultit cu gradul initial. In cazul derivarii monoamele cu gradul 0 vor fi sterse deoarece acestea sunt egale cu 0 si nu mai este necesara scrierea lor in rezultatul operatie. Analog se va intampla aproape la fel si in cazul operatie de integrare doar ca in cazul gradului se va aduna un 1, adica gradul intial va fi incrementat cu 1, iar coeficientul va fi impartit cu gradul nou(cel caruia i s-a adaugat 1). Iar in cazul integrarii trebuie sa tinem cont ca apare o cantata “C” dupa cum ne amintitm de la matematica asfel ca se va mai afisa in campul rezultat si “+C”;

**5.REZULTE**

Testele pentru operatii au fost realizate prin utilizarea Framework-ului utilizat pentru testare Junit, iar dupa cum se vede in imagine teste sunt positive. Astfel concluzia acestor teste este ca operatiile au fost realizate cu success si au fost implementate corent. Pentru realizarea testelor am introdus doua polinoame si mai apoi relultatele fiecarei operatii. Acestea le-am verificat pe un calculator web specializat in calculul cu polinoame si pe foaie din propriile cunostinte unde am adaugat si cateva video-uri recapitulative pentru a realiza corect operatiile.



**6.CONCLUZIE**

In urma acestui proiect realizat in cel mai fiabil si mai eficient limbaj de programare am realizat un calculator de polinoame care poate efectua in mod eficient si corent cele 6 operatiii aplicabile pe doua, respetiv un polinom(adunare, scadere, inmultire, impartire, derivare si integrare). Eficienta este una ridicata, deoarece am tinut cont si de utilizarea memoriei care s-a facut intr-un mod cat mai performant folosindu un ArrayList pentru a nu ocupa mai mult spatiu decat este nevoie.De asemea acest proiect poate fi un exemplu pentru noi daca vreodata vom dori sa realizam un altfel de calculator, nu neapart de polinoame. Eventual unul care va putea fi utilizat in viata de zi cu zi. In acest proiect am folosit majoritatea conceptelor si paradigmelor invatate in semestrul anterior la materia “Programare Orientate pe Obiect” care avea ca si rol introducerea in limbajul de programare java, iar acum am perfectionat utilizarea acelor concepte si metode de programare pentru a putea realiza un program care sa poate fi utiliat si de o persoana care nu are deloc cunostinte de programare. Sunt de parere ca acest proiect a fost un proiect introductive pentru cele ce vor mai urma atat semestrul acesta cat si in semestrele ce vor urma si de ce nu si dupa terminarea studiilor cand va veni vorba de un job in acest domeniu.

Clasele:

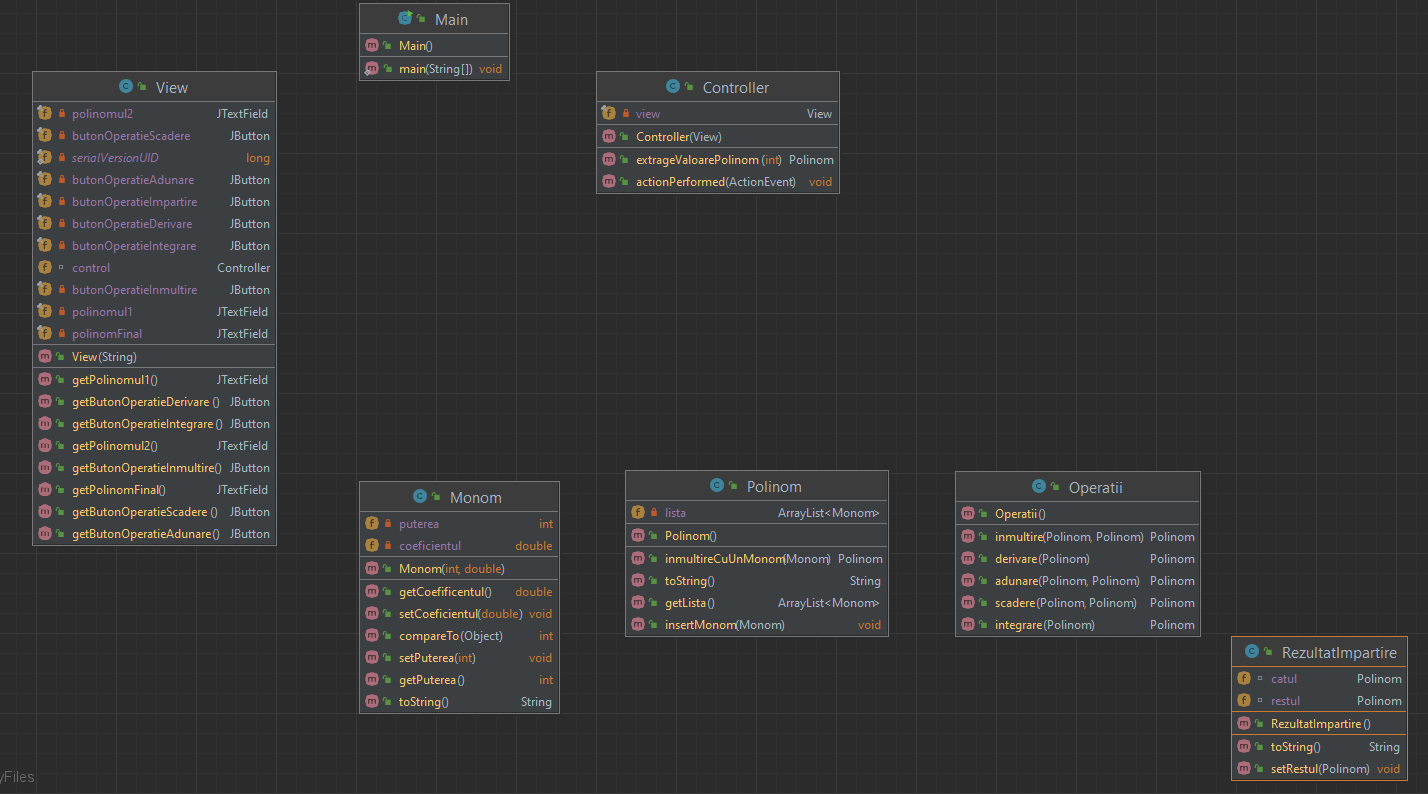


Diagrama UML:

Main

Controller

View

RezultatImpartire

Operatii

Polinom

Monom

**7.BIBLIOGRAFIE**

* Notite si sfaturi din cele 3 laboratoare aferente primei teme
* <https://dsrl.eu/courses/pt/materials/soon.html> (unde am primit materiale utilie pentru partea de testare folosint Junit)
* <https://www.referatele.com/referate/informatica/online8/Polinoame---c---referatele-com.php>
* <https://www.symbolab.com/solver/integral-calculator/%5Cint%203x%5E%7B2%7D%2B4x%20dx?or=input>
* <https://ro.wikipedia.org/wiki/Tabel_de_integrale>
* <https://www.geeksforgeeks.org/>
* https://www.youtube.com/