**DOCUMENTATIE**

**TEMA 1**

NUME STUDENT: SPACZAI CARLA-NOEMI

GRUPA: 30229

**CUPRINS**

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

Obiectivul principal al acestei teme este de a implementa un calculator destinat calculelor cu polinoame si de a realiza o interfata grafica care sa usureze utilizarea calculatorului implementat. Utilizatorul va putea sa introduca polinoamele aferente calculului si sa aleaga operatia pe care vrea sa o efectueze cu acestea.

Obiectivele secundare ale acestei teme sunt:

* analiza si modelarea problemei,
* proiectarea problemei,
* implementarea problemei,
* testarea problemei.

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

Am implementat un calculator destinat calculelor cu polinoame care beneficiaza de o interfata grafica cu scopul de a usura utilizarea acestuia. La momentul rularii, se va deschide o fereastra care cuprinde trei TextField-uri, sase butoane si trei etichete. Utilizatorul va introduce de la tastatura unul sau doua polinoame in TextField-urile semnalate cu etichete ca fiind destinate introducerii polinoamelor. Odata cu alegerea operatiei pe care utilizatorul doreste sa o efectueze, acesta trebuie sa dea “click” pe butonul aferent operatiei alese. Dupa apasarea butonului, in cel de al treilea TextField va fi afisat rezultatul calculului.

In cazul in care datele nu au fost introduse conform sablonului implementat, nu se va putea afisa rezultatul dorit. Este necesara reintroducerea datelor de catre utilizator.

Cazuri de utilizare:

* Adunarea a doua polinoame.

Pentru a efectua cu succes adunarea a doua polinoame este nevoie de indeplinirea a catorva pasi. Utilizatorul trebuie sa introduca primul polinom in TextField-ul semnalat cu eticheta aferenta acestuia. Datele trebuie sa fie introduse corect pentru a putea efectua calculul cu succes. In mod analog, utilizatorul va introduce si cel de al doilea polinom. Odata cu finalizarea introducerii datelor, butonul corespunzator operatiei de adunare trebuie sa fie apasat. Dupa acest pas, programul implementat va efectua adunarea celor doua polinoame primite si va afisa in cel de al treilea TextField rezultatul obtinut.

Daca utilizatorul nu a reusit sa introduca polinoamele conform sablonului implementat, adunarea nu va fi efectuata cu succes. In acest caz, utilizatorul este nevoit sa reintroduca datele daca doreste sa afle un rezultat corect.

* Scaderea a doua polinoame.

Pentru a efectua cu succes scaderea a doua polinoame este nevoie de indeplinirea a catorva pasi. Utilizatorul trebuie sa introduca primul polinom in TextField-ul semnalat cu eticheta aferenta acestuia. Datele trebuie sa fie introduse corect pentru a putea efectua calculul cu succes. In mod analog, utilizatorul va introduce si cel de al doilea polinom. Odata cu finalizarea introducerii datelor, butonul corespunzator operatiei de scadere trebuie sa fie apasat. Dupa acest pas, programul implementat va efectua scaderea celui de al doilea polinom primit din primul si va afisa in cel de al treilea TextField rezultatul obtinut.

Daca utilizatorul nu a reusit sa introduca polinoamele conform sablonului implementat, scaderea nu va fi efectuata cu succes. In acest caz, utilizatorul este nevoit sa reintroduca datele daca doreste sa afle un rezultat corect.

* Inmultirea a doua polinoame.

Pentru a efectua cu succes inmultirea a doua polinoame este nevoie de indeplinirea a catorva pasi. Utilizatorul trebuie sa introduca primul polinom in TextField-ul semnalat cu eticheta aferenta acestuia. Datele trebuie sa fie introduse corect pentru a putea efectua calculul cu succes. In mod analog, utilizatorul va introduce si cel de al doilea polinom. Odata cu finalizarea introducerii datelor, butonul corespunzator operatiei de inmultire trebuie sa fie apasat. Dupa acest pas, programul implementat va efectua inmultirea celor doua polinoame primite si va afisa in cel de al treilea TextField rezultatul obtinut.

Daca utilizatorul nu a reusit sa introduca polinoamele conform sablonului implementat, inmultirea nu va fi efectuata cu succes. In acest caz, utilizatorul este nevoit sa reintroduca datele daca doreste sa afle un rezultat corect.

* Impartirea a doua polinoame.

Pentru a efectua cu succes impartirea a doua polinoame este nevoie de indeplinirea a catorva pasi. Utilizatorul trebuie sa introduca primul polinom in TextField-ul semnalat cu eticheta aferenta acestuia. Datele trebuie sa fie introduse corect pentru a putea efectua calculul cu succes. In mod analog, utilizatorul va introduce si cel de al doilea polinom. Odata cu finalizarea introducerii datelor, butonul corespunzator operatiei de impartire trebuie sa fie apasat. Dupa acest pas, programul implementat va efectua impartirea celor doua polinoame primite si va afisa in cel de al treilea TextField atat catul, cat si restul.

Daca utilizatorul nu a reusit sa introduca polinoamele conform sablonului implementat, inmultirea nu va fi efectuata cu succes. In acest caz, utilizatorul este nevoit sa reintroduca datele daca doreste sa afle un rezultat corect.

* Derivarea unui polinom.

Pentru a efectua cu succes derivarea unui polinom este nevoie de indeplinirea a catorva pasi. Utilizatorul trebuie sa introduca polinomul in TextField-ul semnalat cu eticheta aferenta primului polinom. Datele trebuie sa fie introduse corect pentru a putea efectua calculul cu succes. Odata cu finalizarea introducerii datelor, butonul corespunzator operatiei de derivare trebuie sa fie apasat. Dupa acest pas, programul implementat va efectua derivarea polinomului primit si va afisa in cel de al treilea TextField rezultatul obtinut.

Daca utilizatorul nu a reusit sa introduca polinomul conform sablonului implementat, derivarea nu va fi efectuata cu succes. In acest caz, utilizatorul este nevoit sa reintroduca datele daca doreste sa afle un rezultat corect.

* Integrarea unui polinom.

Pentru a efectua cu succes integrarea unui polinom este nevoie de indeplinirea a catorva pasi. Utilizatorul trebuie sa introduca polinomul in TextField-ul semnalat cu eticheta aferenta primului polinom. Datele trebuie sa fie introduse corect pentru a putea efectua calculul cu succes. Odata cu finalizarea introducerii datelor, butonul corespunzator operatiei de integrare trebuie sa fie apasat. Dupa acest pas, programul implementat va efectua integrarea polinomului primit si va afisa in cel de al treilea TextField rezultatul obtinut.

Daca utilizatorul nu a reusit sa introduca polinomul conform sablonului implementat, integrarea nu va fi efectuata cu succes. In acest caz, utilizatorul este nevoit sa reintroduca datele daca doreste sa afle un rezultat corect.

1. **Proiectare**

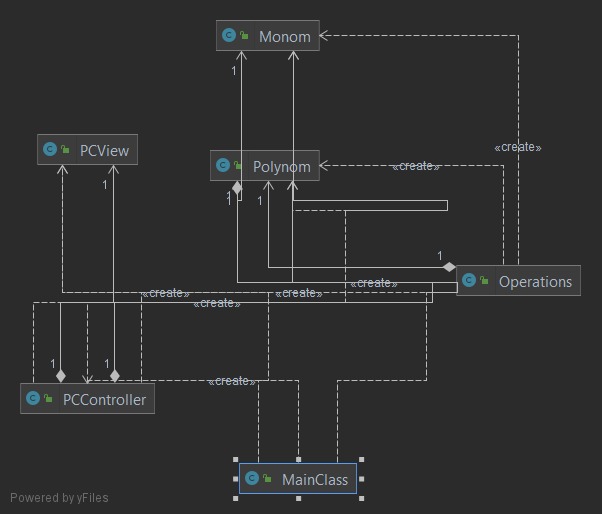
Proiectarea acestei aplicatii a fost realizata cu ajutorul paradigmelor OOP, in stilul Model-View-Controller (MVC).

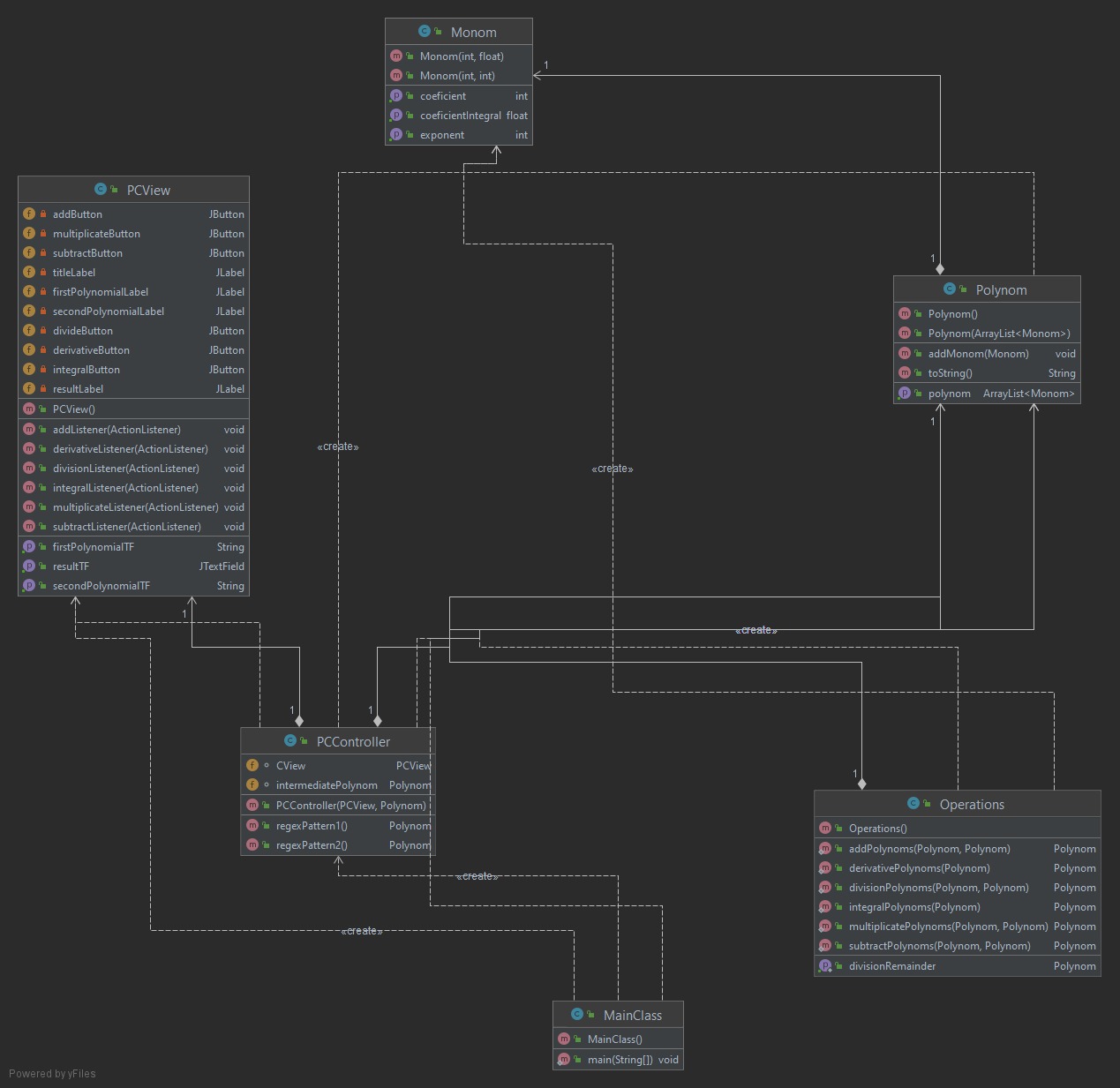
Partea de “Model” cuprinde trei clase (Operations, Monom, Polynom) cu ajutorul carora manipuleaza operatiunile logice si de utilizare de informatie. Partea de “View” cuprinde o singura clasa (PCView) cu ajutorul careia genereaza reprezentarea grafica necesara interactionarii simplificate cu utilizatorul. Partea de “Controller” cuprinde o singura clasa (PCController) cu ajutorul careia realizam legatura dintre Model si View. Prin intermediul PCController-ului oferim butoanelor si intregii interfete grafice o functionalitate exacta si oferim operatiunilor logice o infatisare mai usor de inteles si mai usor de folosit.

Aceasta aplicatie a fost proiectata cu ajutorul a sapte clase: Operations, Monom, Polynom, PCView, PCController, MainClass, TestClass. Clasa Monom presupune trei atribute private corespunzatoare coeficientului intreg, coeficientului real si exponentului unui monom. Clasa Polynom cuprinde un ArrayList de obiecte de tip Monom si doua metode. Clasa Operations cuprinde un atribut privat corespunzator restului impartirii si sase metode statice, cate una pentru fiecare operatie. Clasa PCView presupune existenta a sase butoane, patru etichete, trei TextField-uri si sase metode, cate una pentru adaugarea unui “listener” fiecarui buton. Clasa PCController presupune doua atribute de tip Polynom si PCView si doua metode. Clasa MainClass cuprinde “main”-ul. Clasa TestClass presupune sapte teste pentru a verifica corectitudinea operatiilor.

Toate aceste atribute private au fost accesate si de celelalte clase prin intermediul getter-elor.

In urmatoarele imagini, se poate observa diagrama UML a claselor:





1. **Implementare**

Pentru implementarea acestui calculator polinomial a fost nevoie de sapte clase: Operations, Monom, Polynom, PCView, PCController, MainClass, TestClass. Acestea au fost organizate si legate cu ajutorul MVC-ului si a paradigmelor OOP.

Clasa Monom a fost realizata cu ajutorul a trei atribute private (exponent, coeficient, coeficientIntegral) si doi constructori. Primele doua atribute sunt de tip “int” si cel de al treilea este de tip “float”. Am implementat doi constructori pentru a putea tine evidenta mai usor daca un monom are coeficientul de tip intreg sau real. Daca monomul creat va avea un coeficient real, atributul “coeficient” va fi setat pe 1, iar daca monomul creat va avea un coeficient intreg, atributul “coeficientIntegral” va fi setat pe 1.

Clasa Polynom a fost implementata cu ajutorul unui ArrayList de obiecte de tip Monom deoarece am vrut sa retinem separat fiecare monom al polinomului, dar in aceeasi structura de date. In aceasta clasa mai gasim doi constructori: unul va fi folosit daca vrem sa avem un polinom gol la care sa putem adauga monoame mai tarziu, iar celalalt va fi folosit cand avem un ArrayList de obiecte de tip Monom si vrem doar sa cream un polinom cu acesta. Am definit si o metoda de adaugare a unui monom intr-un polinom pentru a usura operatiile cu polinoame.

Clasa Operations presupune sase metode statice care corespund celor sase operatii cu polinoame. Metoda specifica adunarii a doua polinoame ia fiecare monom din cel de al doilea polinom si verifica daca exista deja un monom cu acelasi exponent in primul polinom sau nu. Daca exista, vom aduna coeficientii celor doua monoame cu acelasi exponent, iar daca nu exista, vom adauga si acest monom la primul polinom. Metoda specifica scaderii a doua polinoame ia fiecare monom din cel de al doilea polinom si verifica daca exista deja un monom cu acelasi exponent in primul polinom sau nu. Daca exista, vom scadea coeficientii celor doua monoame cu acelasi exponent, iar daca nu exista, vom scadea acest monom din primul polinom. Metoda specifica inmultirii a doua polinoame ia fiecare monom din cel de al doilea polinom si il inmulteste cu fiecare monom din primul polinom. Pentru a inmulti doua monoame, vom aduna exponentii si vom inmultii coeficientii. Pe parcurs, vom aduna toate aceste produse de cate doua monoame si vom obtine produsul celor doua polinoame. Metoda specifica impartirii a doua polinoame verifica care dintre polinoame are monomul cu cel mai mare exponent, iar polinomul in cauza va fi impartit la celalalt. Impartirea va fi realizata in mod recursiv. Cat timp exponentul celui mai mare monom din primul polinom este mai mare sau egal cu cel mai mare monom din al doilea polinom, vom imparti cel mai mare monom ramas cu primul monom din al doilea polinom. Rezultatul va fi trecut la cat si se va scadea din primul polinom produsul dintre ultimul rezultat si al doilea polinom. La final, vom avea catul impartirii si restul (ceea ce a ramas in primul polinom). Metoda specifica derivarii a unui polinom ia fiecare monom din polinom, ii scade exponentul cu 1 si ii inmulteste coeficientul cu exponentul anterior. La final, vom avea polinomul derivat. Metoda specifica integrarii a unui polinom ia fiecare monom din polinom, ii creste exponentul cu 1 si ii imparte coeficientul cu noul exponent. La final, vom avea polinomul integrat.

Clasa PCView contine tot ceea ce este legat de reprezentarea grafica a calculatorului polinomial. Aceasta interfata grafica presupune sase butoane corespunzatoare fiecarei operatii implementate (utilizatorul va alege butonul pe care il va apasa in functie de operatia pe care vrea sa o efectueze), trei etichete care indica titlul si scopul a doua TextField-uri, trei TextField-uri (doua dintre ele sunt pentru cate un polinom, iar cel ramas este pentru rezultat) si “listener”-e pentru fiecare buton.

Clasa PCController a fost implementata cu ajutorul a doua metode care primesc cate un String si il transforma intr-un obiect de tip Polynom. Tot in aceasta clasa mai gasim si cate o clasa pentru fiecare buton de tipul ActionListener. Aceasta clasa primeste de la PCView datele introduse de catre utilizator sub forma unui String/a doua String-uri si il/le transforma in obiect(e) de tip Polynom. Odata transformat(e), este/sunt trimis(e) claselor din Model unde va fi efectuata operatia aferenta butonului apasat. La final, obtinem rezultatul operatiei il trimitem inapoi pentru a fi afisat mai pe urma in TextField. Astfel, existenta acestei clase este esentiala in legatura claselor.

Clasa MainClass initializeaza trei obiecte (PCView, Polynom, PCController) cu ajutorul carora creeaza legatura MVC

Clasa TestClass presupune folosirea JUnit-ului cu scopul de a verifica corectitudinea operatiilor implementate. In fiecare test avem unul/doua polinom/polinoame pe care aplicam o operatie si inca unul in care punem noi rezultatul asteptat. Daca cele doua coincid, testul s-a efectuat cu succes.

1. **Rezultate**

Pentru a verifica daca rezultatele obtinute sunt corecte, ne folosim de clasa TestClass care presupune testarea cu JUnit. Avem sapte teste destinate adunarii, scaderii, inmultirii, catului impartirii, restului impartirii, derivarii si integrarii. Fiecare test presupune initializarea unui polinom/a doua polinoame pentru a efectua operatia dorita pe el/ele. Ne folosim de inca un polinom pentru a retine rezultatul asteptat. Transformam rezultatul operatiei si rezultatul asteptat in String-uri si le comparam. Daca coincid, testul s-a efectuat cu succes, dar daca nu coincid, operatia nu a fost implementata corect.

1. **Concluzii**

Aceasta tema a fost foarte interesanta deoarece a presupus implementarea a ceva destul de apropiat de ceea ce vedem zilnic. Totodata, a fost foarte utila deoarece ne-a facut sa recapitulam si sa mai exersam cunostintele noastre din Java, dar ne-a facut sa si invatam cateva lucrui noi.