DOCUMENTATIE

*TEMA 1*

NUME STUDENT: Miritoiu Cristian George

GRUPA:30225

Cuprins

[1. Obiectivul temei 2](#_Toc97680761)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 2](#_Toc97680762)

[3. Proiectare 2](#_Toc97680763)

[4. Implementare 3](#_Toc97680764)

[5. Rezultate 5](#_Toc97680765)

[6. Concluzii 5](#_Toc97680766)

[7. Bibliografie 5](#_Toc97680767)

# Obiectivul temei

Obiectivul principal al temei reprezinta realizarea unui calculator in vederea lucrului cu polinoame.Acest problema presupune construi clasele necesare precum :monom si polinom impreuna cu metodele acestor precum si o interfata user-friendly pentru manipularea si afisarea polinoamelor.

# Analiza problemei, modelare, scenarii, cazuri de utilizare

Problema presupune ca utilizatorul sa introduca polinoamele in cadrul celor 2 textfields sub forma unor substringuri repetate de tip +/-cX^e (c=coeficient,e=exponent).Acestea sunt transformate in clase polinoame formate dintr-o lista de monoame si gradul polinomului.Odata introduse cele doua polinoame(sau unul in cadrul operatilor unare) utilizatorul poate sa selecteze una dintre cele 6 operati puse la dispozitie :adunare,scadere,inmultire,impartire,derivare si intergrare. Rezultatul va fi afisat in cadrul unei noi fereste.In cazul introduceri unui polinom ce nu respecta patternul specificat anterior sau unul dintre polinoame nu este introdus va fi afisata o noua fereasta cu mesajul “Polinom invalid”.Metoda de rezolvare foloseste un pattern arhitectural precum Model-View-Controller pentru o mai buna claritate a codului separand elementele de interfata in clasa View,operatile de calcul in cadrul clasei Model ,iar Controller asigura legatura dintre interfata si operatiile efectuare.

# Graphical user interface, treemap chart Description automatically generatedProiectare

Proiectul este structurat in arhitectura Model-View -Controller.

Interfata prezinta 2 text\_field box pentru ca utilizatorul sa introduc polinoamele fie prin tastatura oferita de interfata avand 15 butoane pentru a introduce date unul dintre butoane fiind folosit pentru a sterge ultimul caracter introdus (Del) sau alte Metoda alternative furnizate de utilizator.Interfata prezinta de asemenea 6 butoane pentru efectuarea operatiior asupra polinoamenlor.Operatia de intergrare si derivare se efectueaza doar asupra primului polinom ,iar celelalt asupra celor doua polinoame, iar in cazul in care unul dintre acestea nu este introdus si se incearca efectuarea operatiilor ,se va afisa o noua fereasta cu mesajul “Polinom Invalid”.Rezultatul operatiilor va fi afisat cu ajutorul unei noi ferestre avand un text\_field box ce afiseaza reprezentarea String a polinomului si un buton ok pentru confirmare rezutatului si inchiderea ferestrei.Dupa fiecare operatie utilizatorul poate sa continue efectuarea altor operatii dupa inchiderea ferestrei de rezultat.

Programul contine 6 clase principale si o clasa de teste . Clasele monom si polinom sunt utilizate pentru a construi si stoca diverse polinoame introduse se utilizator.Monom are 2 atribute exponent si coeficient pecum si settere,gettera constructor si o metoda Comparable pentru care clasa Monom implementeaza Interfata COMPARABLE.Pentru Polinom avem 2 atribure grad,v contructori setter etc si 4 metode ce vor fi prezentare in detaliu in capitolul urmator: simplify ,sentinel, valid\_p ,generate .Clasa main este folosita pentru rularea programului. A screenshot of a video game

Description automatically generatedCelelate 3 clase :Model View si Controller. Clasa Model prezinta doar metodele ce reprezinta operatiile aupra polinoamelor.Clasa View este formate dintr-ul numar larg de atribute ce reprezinta componentele interfertei deasemenea prevede metode pentru a adauga difreriti action\_listener asupra butoanelor pentru o claritate mai buna a codului si metode pentru crearea noilor ferestere de rezultate si erori.Clasa Controller utilizeaza cele doua clase prezentate anterior in vederea controlului fluxului de intructiunii.Aceaste contine 6 subclase utilizate pentru a asigura legatura dintre Model si View prin folosirea metodelor din View pentru a adauga functionalitatea butoanelor.Fiecare subclasa preia informatiile din View construieste polinoamele apeleaza operatia ceruta fi trimite datele inapoi catre View in vedera afisarii acestora.

# Implementare

**Monom** este o clasa utilizata in cadrul clasei Polinom pentru a construi polinomul din bucati mai mici.Fiecare Monom prezinta coeficient si exponent.Cea mai imporatanta metoda din cadrul aceste clase este metoda Compare\_to utilizata pentru a sorta descrescator in functie de exponent elemntele unei colectii de monoame.

**Polinom** este clasa utilizata pentru reprezentarea polinoamelor simple cu exponenti pozitivi intregi si coeficienti zecimali.Acesta are atributul de grad si un arraylist pentru a stoca o cantitate variabila de monoame.

* Metoda valid\_p testeaza cu ajutorul claselor java Pattern si Matcher pentru a verifica daca stringul furnizat este un polinom valid in caz contrar acesta apeleaza functia de afisare a unui ferestre de erori si returneaza false.
* Metoda simplify este utilizata in vederea aduceri polinomului la o forma mai simpla. Cu 2 iterati prin lista se cauta elemetele care au acelasi grad acestea se aduna ,astfel ramanem cu primul element gasit iar celalat element este scos si lista.In cazul unei liste goale vom adauga un element santinela cu coeficient zero si grad zero.
* Metoda sentinel a fost creata pentru a umple lista cu termeni santinela avand coeficientii zero in vederea simplificarii operatiilor asupa polinoamelor.Ordonam descrescator si simplificam lista v.Daca gradul a doi termeni consecutivi nu este de asemenea consecutiv descrescator intre acestia se adauga noi elemente.Iar daca ultimul element nu are gradul zero se vor adauga elemente in lista pana cand conditia este indeplinita.
* Metoda generate este folosita pentru a construi un polinom dintr-un string.Avand doua variabile coeficient si exponent initializate.Vom impartii stringul de monoame in functie de + si – cu tot cu aceste (simbouri pentru numere negative) si substringurile obtinute vor fi salvate intr-un array.Vom itera prin acest array si in functie de diferite conditii vom modifica valorile variabilelor initializate anterior si vom adauga un nou monom cu aceste variabile in list.Conditiile pentru modificarea valorilor sunt: daca nu contine X se va lua valoarea efectiva, daca nu contine exponent se va schimba exponent in 1 si vom lua valoarea dinaintea lui X,iar daca acesta contine exponent ^ se va rupe in doua substringul in functie de “X^” si se vor lua valorile efective ale acestor substringuri.

**Model**

* Metoda adunare primeste doua polinoame si returneaza rezultatul adunari acestora.Se gaseste gradul ce mai mic.Se parcuge ambele polinoame odata si pentru acelasi index reprezinta aceeasi putere iar elementele se aduna si adauga la rezultat. Se cauta apoi polinomul cu gradul cel mai mare si se adauga elementele ramase si neschimbate la rezultat.
* Metoda de scadere este asemanatoare cu adunarea dar elementele se scad.
* Metoda de inmultire se inmultesc pe rand prin doua for-uri fiecare element cu fiecare se adauga la rezultat.La final rezultatul se simplifica si se adauga santinelele.
* Metoda de derivare presupune ca pentru fiecare element sa se decrementeza exponentul iar coenficientul este inmultit cu gradul.
* Metoda de integrare presupune ca pentru fiecare element sa se incrementeze exponentul ,iar coeficientul se imparte la exponent.
* Metoda de impartire ,se creaza un Polinom avand doar un monom care este impartirea monoamelor cu grad superior din fiecare polinom.Se aduna monomul la cat.Din rest se scade inmultirea dintre impartitor la si cat.Acest proces este repetat pana cand restul are un grad mai mic decat impartoru sa este null.

**VIEW** contine atribute ce sunt componentele interfatei.Metodele ce au ca argument paramentru de tip action\_listener primes o subclasa a clasei Controller in vederea controlului fluxului de intructiunii.

Controller are ca atribute obiecte ale claselor View si Model.Acesta are contine 6 sublclase ce au in vedere preluarea datelor de la utilizator,construirea unor polinoame cu acestea ,apelul functiilor din model si trimiterea datelor inapoi la interfata pentru a putea fi afisate pentru utilizator.

# Rezultate

Testarea unitara sa efectuat cu ajutorul framework-ului Junit.In cadrul fisierului Test se testeaza cele 6 operati asupra polinoamelor,fiecare cu un scenariu de testare diferit. Fisierul contine arg1,arg2 doi vectori de stringuri ce reprezinta argumentele testari a 5 operatii ,iar in cadrul impartiri se foloseste un set nou de argumente pentru a respecta conditiile impartiri.In fiecare test se afla un vector de rezultate asteptate din partea operatiilor rezultate obtinute cu ajutorlui siteului <https://www.emathhelp.net/en/calculators/algebra-1/polynomial-calculator/>. Fiecare element din vector este generat ca vector si apoi returnat ca string pentru a efectua assert-uri corespunzatoare.Deasemenea este folosita comnada assertall pentru a efectua toate testele indiferent daca unul din acestea esueaza.

# Concluzii

In cadrul acestei teme am invatat sa utilizez un limbaj orientat pe obiecte si importanta unei arhitecturi de design. Problema a pus la testare cunostiintele dobandite in cadrul materiei de Programare Orientate pe Obiecte si obiectivele cheie in cadrul proiectarii si mentinerii proiectarii unui program.

# Bibliografie

* <https://dsrl.eu/courses/pt/>
* <https://www.emathhelp.net/en/calculators/algebra-1/polynomial-calculator/>
* Curs programare Orientata pe Obiecte-Raluca Brehar