DOCUMENTATIE

Calculator Polinoame Tema 1

NUME STUDENT: Nita Iosif-Gabriel

GRUPA: 30223

# CUPRINS

[1. Obiectivul temei 3](file:///C:\Users\Gabriel\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297885)

[2. Analiza problemei, modelare, scenarii, cazuri de utilizare 3](file:///C:\Users\Gabriel\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297886)

[3. Proiectare 3](file:///C:\Users\Gabriel\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297887)

[4. Implementare 3](file:///C:\Users\Gabriel\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297888)

[5. Rezultate 3](file:///C:\Users\Gabriel\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297889)

[6. Concluzii 3](file:///C:\Users\Gabriel\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297890)

[7. Bibliografie 3](file:///C:\Users\Gabriel\Downloads\PT2021-2022_Documentation_Template%20(2).doc#_Toc95297891)

1.Obiectivul temei

Obiectivul acestei teme este proiectarea si implementarea unui calculator de polinoame cat si a unei interfete graficare care ajuta la rezolvarea operatiilor matematice de adunare, scadere,inmultire ,impartire ,derivare si integrare.

Pasii urmati pentru indeplinirea obiectivului sunt:

1. Dezvoltarea unor scenario in care aplicatia ve fi dezvoltata
2. Alegerea structuriilor de date, structura proiectului trebuie sa fie una eficienta
3. Impartirea corespunzatoare a claselor si a rolurilor acestora
4. Implementarea si dezvoltarea algoritmilor corespunzatori pentru toate cazurile
5. Implementarea solutiilor, combinarea algoritmilor
6. Testarea si gasirea cazurilor care pot cauza probleme aplicatie, remedierea pe cate se poate a acestora

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

Analiza problemei si modelarea solutiilor. La problema prezentata trebuie sa descompunem elementele si sa punem intrebari ajutatoare rezolvarii problemei. Un polinom este un sir de monoame, un monom este format dintr-un coeficient si o putere(grad). Forma unui monom este una standard ax^b, a semnificand coeficientul “^” ridicat la puterea b. Ne-a fost recomandata folosirea unei liste de monoame. Practic Polinomul este in sine o lista de monoame. Si ne-a mai fost recomandata folosirea regex-urilor pentru a gasi pattern-ul pentru a identifica coeficientul si puterea fiecarui element.

Pentru identificarea acestora am ales sa implementez in aplicatie o metoda care tranfsorma polinomul de exemplu daca utilizatorul introduce x+1, acea metoda transforma polinomul in 1.0x^1+1.0x^0(coeficientii monoamelor fiind double). Aceasta implementarea ajuta la desfasurarea operatiilor dorite in cadrul aplicatiie si anume operatiia de adunare, scadere, inmultire, impartire, derivare si integrare.

Interfata pe care am realizat-o este foarte simpla, arata exact ca un calculator de buzunar nu este necesara folosirea tastaturii pentru a rula programul. Interfata este dotata cu butoane pentru fiecare cifra, buton pentru fiecare operatie si alte butoane care simplifica utilizarea, de asemnrea aceasta are implementata si mesaje ajutatoare de eroare.

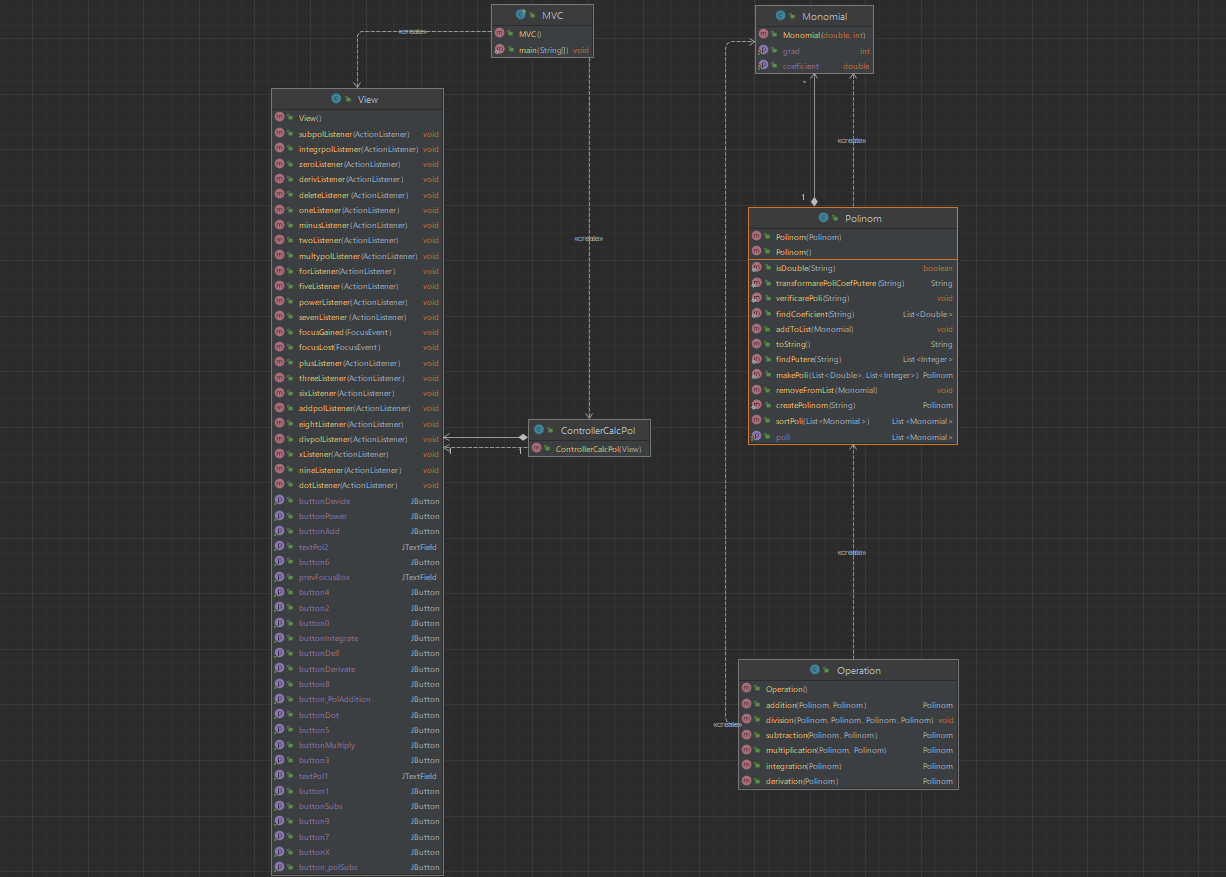
3.Proiectare

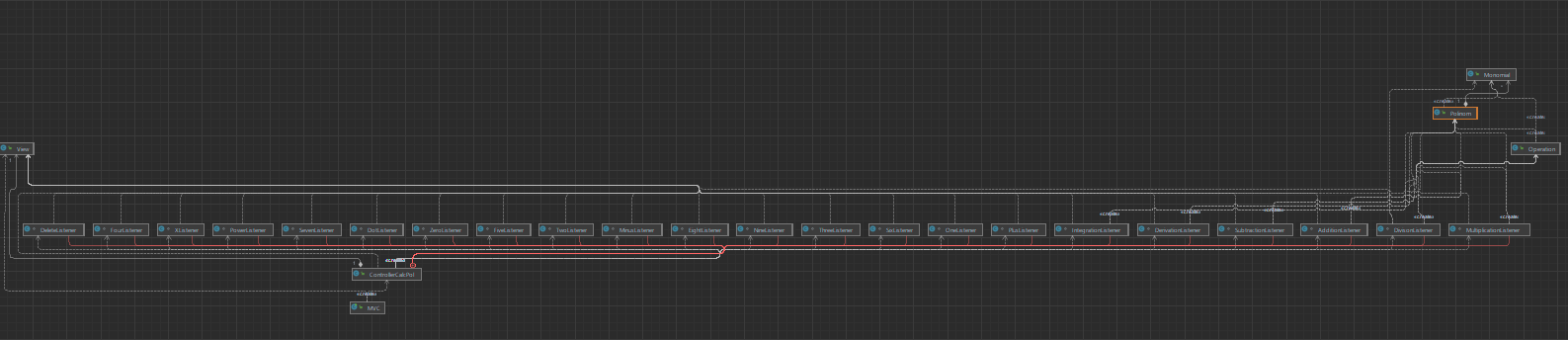
|  |  |
| --- | --- |
| **Package** | **Class** |
| main -> **java** | MVC |
| main -> java -> DataModels | Monomial  Polinom  Operation |
| main -> java -> **controller** | ControllerCalcPol |
| main -> java -> **view** | View |
| test -> **java** | AdditionTest  SubtractionTest  MultiplicationTest  DivisionTest  DerivationTest  IntegrationTest |

|  |  |
| --- | --- |
| **Class** | **Descriere** |
| Monomial | Clasa are două atribute, coeficient de tip double și grad (exponent) de tip int, un constructor și metodele standard de set și get. |
| Polynomial | Clasa are un singur atribut, o lista (ArrayList) de monoame, doi constructori, unul care primeste un polinom si il sorteaza dupa putere, si al doilea constructor care apeleaza doar super(). Metode de adaugare si scoatere din lista, metodele de set, metoda set sorteaza lista și get. |
| Operatie | Clasa nu are niciun atribut, însă conține toate metodele operațiilor implementate: adunare, scadere, inmultire, impartire, derivare si integrare. |
| View | Clasa are ca atribute toate componentele de la nivel de interfata care necesita manipularea ulterioara. Contine constructorul interfetei, metode care setează rezultatul în eticheta corespunzătoare și metode care adaugă ascultători pentru fiecare buton în parte. |
| ControllerCalcPol | Clasa care leagă clasele din pachetul model cu clasa View. Are un singur atribut, și anume un obiect de tip View. In cadrul constructorului inițializez atributul de tip View si asociez metodele ascultatoare pentru fiecare buton cu clasa interna corespunzatoare. Afiseza mesaje de eroare, sau atentionare. |
| MVC | Clasa care conține metoda main. Aici creez un obiect de tip View pe care îl pasez constructorului ControllerCalPol. |
| AdunareTest | Clasa care contine testul pentru operatia de adunare. |
| ScadereTest | Clasa care contine testul pentru operatia de scadere. |
| InmultireTest | Clasa care contine testul pentru operatia de inmultire. |
| ImpartireTest | Clasa care contine testul pentru operatia de împartire. |
| DerivareTest | Clasa care contine testul pentru operatia de derivare. |
| IntegrareTest | Clasa care contine testul pentru operatia de integrare. |

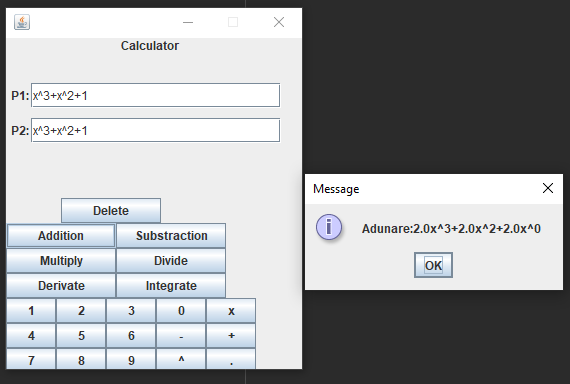
Proiectarea claselor in aceasta forma m-a ajutat sa inteleg mai bine problema si cum trebuie sa dezvolt algoritmii pentru fiecare clasa. In pachetul DataModels am decis sa pun clasa Monomial, Polinom si clasa Operation pentru ca aceste clase rezolva partea calculelor.

Clasa View am decis sa o pun in pachetul View deoarece aceasta are legatura cu interfata grafica. In pachetul Controller am clasa ControllerCalcPol aceasta clasa face legatura dintre interfata si calculele efectuate in spate.





Interfata arata astfel:



Fiecare buton functioneaza corespunzator, butonul delete sterge cate un singur caracter, toate rezultatele se vor afisa in acest format din imagine.

4.Implementare

Clasa “Monomial” are doar doua atribute private si anume grad care este intreg si coeficient care este double.

Constructorul care seteaza coeficientul si gradul, si metodele get si set.

O imagine care conține text

Descriere generată automat

Clasa Polinom

Aceasta este una dintre cele mai complexe clase din aceasta aplicatie, continue lista de monoame, doi constructori, constructorul care are un parametru sorteaza polinomul descrescator dupa grad daca acesta a fost introdus altfel. O metoda de adaugare a monomului in lista de monoame si una de a scoate un monom de pe o anumita pozitie, si mai continue metodele set si get.

O imagine care conține text

Descriere generată automat

Metoda de verificare a polinomului care arunca exceptii daca acesta nu respecta forma unui polinom.

O imagine care conține text

Descriere generată automat

Metoda transformarePoliCoefPutere care primeste un string si transforma coeficientii si puterile monoamelor de tipul x si 1, acestea sunt transformate in 1.0x^1 respectiv 1.0x^0 aceasta metoda returneaza stringul modificat

O imagine care conține text

Descriere generată automat

Metodele findPutere are ca parametru un string. In interior se afla o lista de intregi si regex-ul necesar pentru a scoate din stringul dat ca parametru puterea fiecarui monom, aceasta metoda returneaza lista de puteri

O imagine care conține text

Descriere generată automat

findCoeficient are ca parametru un string. In interior se afla o lista de double si regex-ul necesar pentru a scoate din stringul dat ca parametru coeficientul fiecarui monom din stringul dat ca parametru. Acestea metoda returneaza o lista de coeficienti de tipul double.

Metoda makePoli este metoda care primeste lista de coeficienti si cea de puteri si adauga in lista de monoame a polinomului. Creand astfel polinomul.

O imagine care conține text

Descriere generată automat

Metoda createPolinom foloseste majoritatea metodelor prezentate mai sus, am facut aceasta metoda pentru a face codul sa fie mai lizibil si usor de inteles.

Metoda toString care este o metoda de afisare.

O imagine care conține text

Descriere generată automat

Clasa Operation

Aceasta clasa nu are nici un atribut, aceasta a fost creata pentru a contine toate metodele necesare operatiilor de adunare, scadere, inmultire, impartire , derivare si de integrare.

Metoda addition, aceasta metoda realizaeaza adunarea polinoamelor, primele doua foreach-uri realizeaza adunarea dintre coeficienti, in aceasta metoda am ales sa lucrez pe niste liste auxiliare deoarece am intampinat probleme cand doream sa dau remove la un element din listele polinoamelor date ca parametrii. Am refolosit una dintre liste pentru a pune rezultatul.

Ultimele doua for-uri adauga in lista elementele ramase de la prima adaugare, apoi rezultatul trebuie sortat de aceea am implementat in setPoli o sortare a polinomului descrescatoare dupa grad

O imagine care conține text

Descriere generată automat

Metoda substraction aceasta metoda realizeaza operatia de scadere a polinoamelor, implementarea este asemanatoare ca implementarea de la adunare. In primele doua for-uri se creeaza un nou monom si acesta este adaugat in lista ca urmare monomul mon1 este scos.

O imagine care conține text

Descriere generată automat

Cele doua for-uri de pe urma adauga in lista lmon1 monoamele care au gradul diferit si nu au fost adaugate de primele 2 for-uri. Am refolosit una dintre liste pentru a pune rezultatul.

Metoda integration primeste ca parametru polinomul si realizeaza integrarea acestuia. Metoda parcurge lista de monomi si seteza gradul la gradul actual plus 1 variabila rez imparte coeficientul la grad apoi acesta este setat in polinom cu o precizie de doua decimale pentru a face citirea mai usoara.

O imagine care conține text

Descriere generată automat

Metoda derivare primeste ca parametru polinomul. Aceasta realizeaza derivarea inputului parcurgand lista de monomi si daca gradul polinomului este mai mare decat zero seteaza coeficientul la gradul monomului inmultit cu coeficientul iar gradul il scade cu 1 daca nu metoda seteaza coeficientul la 0.

Metoda multiplication primeste ca parametri cele doua polinoame. Din nou am ales implementarea cu doua liste auxiliare pentru a evita erorile pe care le-am primit la adunare.

Primele 2 for-uri realizeaza inmultirea coeficientilor si adunarea gradelor creand cate un nou monom si acesta este adugat. in lista auxiliara rez.

Iar ultimele doua for-uri realizeaza adunarea elementelor care au acelasi grad pentru a returna un rezultat optim. Rezultatul este sortat apoi dupa grad de metoda setPoli.

O imagine care conține text

Descriere generată automat

Metoda division realizeaza impartirea polinoamelor primite ca parametri si trimite tot prin parametri cat-ul si restul impartiri.

Conditia de oprire este gradul polinomului1 sa fie diferit de zero si gradul polinomului1 sa fie mai mic decat gradul polinomului1 a doua conditie fiind negata.

Am realizat impartirea coeficientilor si scadere puterilor primului monom din polinomul1 si am create un al polinom, polinomul re, cat-ul este elementul de pe pozitia zero din polinomul nou

Dupa am realizat operatia de inmultire a polinomului re si polinomul2 apoi am realizat operatia de scadere dintre polinomul re si polinomul1. Dupa a scos din lista monomul de pe pozita 0

Dupa iesirea din while restul se afla in polinomul1, daca size-ul acestuia este diferit de zero restul va fi pus in Polinomul rest din antet daca nu in Polinomul rest se va adauga 0.

O imagine care conține text

Descriere generată automat

Clasa View

Aceasta clasa are ca atrbute toate elementele din interfata, buton pentru fiecare cifra si operatie si alte butoane utile. Aceast clasa extinde clasa JFrame si implementeaza FocusListener.

In constructor am initializat fiecare buton l-am adaugat in interfata si am setat pozitia fiecaruia pentru a obtine o interfata cat mai simplu de utilizat chiar daca aceal utilizator nu este specializat in utilizarea interfetelor grafice.

O imagine care conține text

Descriere generată automat

Aceasta clasa contine metode care adauga ActionListener si metoda get pentru fiecare componenta in parte.

O imagine care conține text

Descriere generată automat

O imagine care conține text

Descriere generată automat

Una dintre metodele importante in aceasta clasa este metoda focusGained care este de folos atunci cand doresti sa scrii in textfield-uri deoarece atunci cand apesi pe un buton focus-ul se transfera pe buton si nu poti sa scrii. In prevFocus se afla textfield-ul pe care a fost focusul inainte de a apasa pe buton.

O imagine care conține text

Descriere generată automat

Clasa ControllerCalcPol

Aceasta clasa contine un singur atribut acesta fiind o instanta a clasei View. In interiorul constructorului acestei clase ControllerCalcPol se intializeaza singurul atribut adica view si se implenteaza ascultatorii pentru butoanele interfetei.

O imagine care conține text

Descriere generată automat

Aceasta clasa contine clasele pentru ascultatorii butoanelor interfetei O imagine care conține text

Descriere generată automat

La apasarea butonului in functie de textfield-ul pe care a fost focusul se adauga 0. Este la fel la toate celelate butoane de tipul cifra sau simbol.

Clasa pentru butonul delete implementeaza stergerea unei cifre sau simbol.

In schimb clasele pentru implementarea ascultatorilor butoanelor specifice operatiilor sunt putin diferite

O imagine care conține text

Descriere generată automat

Daca se apasa spre exemplu butonul de adunare se colecteaza inputul utilizatorului si se pun in cele doua string-uri acestea sunt verificate de metoda verifPoli din clasa Polinom si daca nu indeplinesc conditiile se afiseaza o eroare pe ecran.

Daca conditiile sunt indeplinite se realizeaza operatia de adunare care din nou este afisata pe ecran.

O imagine care conține text

Descriere generată automat

Clasa care implementeaza integrarea si derivarea sunt putin diferite deoarece acestea primesc input doar de la textfield-ul polinomului1 user-ul este atentionat daca doreste integrarea sau derivarea polinomului2 acesta trebuie sa copieze polinomul2 in polinomul1 pentru a efectua operatia de derivare sau integrare.

Clasa care implementeaza impartirea afiseaza un mesaj care specifica faptul ca polinomul1 trebuie sa fie polinomul care are gradul mai mare, aceasta afiseaza un mesaj pentru a instinta utilizatorul de acest lucru.

O imagine care conține text

Descriere generată automat

5.Rezultate

Pentru testarea rezultatelor am utilizat Junit. In aceasta metoda test am creat doua String-uri dupa am apelat metoda de creare a polinoamelor. La impartire am creat si Polinomul rest si cat apoi in string-urile rest si cat am scris respunsurile. Am folosit Assertions.assertTrue deoarece nu am reusit sa import cealalta clasa.

O imagine care conține text

Descriere generată automat

6.Concluzii

In urma rezolvarii acestei teme, pot spune ca mi-am dezvoltat gandirea practica si impartirea problemei in probleme mai mici. Am acumulat cunostinte noi cu am fi utilizarea regex-ului si a Pattern-ului si Matcher-ului dar mi-am adus aminte si cunostinte uitate cum ar fi dezvoltarea interfetei grafice.

Ca dezvoltari ulterioare as implementa o noua metoda toString care sa ajute utilizatorul sa vada mai clar rezultatul, rezultatul meu este de forma a.0x^2+b.0x^1+c.0x^0 cred ca ar fi mai usor de vizualizat un rezultat de tipul ax^2+bx+c. Plus as mai adauga mesaje de erori si exceptii.

7.Bibliografie

Nu am avut nici o sursa din care sa ma inspir la redactarea acestei documentatii. Acestea sunt ideile mele care au venit pe parcursul proiectarii si implemetarii proiectului si anumite ideei care mi-au venit pe moment.