

Documentatie

Tema 1

Calculator de polinoame

Nume: Marinescu Roxana-Maria

Grupa: 30227

**Obiectivul temei:**

Prima tema de laborator are ca și obiectiv principal obținerea unui calculator de polinoame prin care să se poată introduce doua polinoame folosind o interfata grafice, să se poata selecta o operație (adunare, scădere, înmulțire, împărțire, derivare , integrare), pentru ca la final să se afiseze rezultatul corect al calculului.

Obiectivele secundare ale temei sunt urmatoarele:

* Înțelegerea corecta a cerintei
* Găsirea unei soluții care să fie valida, oricare ar fi tipul de polinom introdus
* Crearea unei interfete grafice care sa preia polinoamele și sa afiseze rezultatul
* Scrierea codului conform solutiei găsite
* Obtinerea rezultatului

**Analiza problemei, modelare, scenarii, cazuri de utilizare:**

Analiza problemei - înțelegerea cerinței și generarea unui rezultat.

Am creat o schemă care să prezinte ceea ce se urmărește prin aceasta tema. Este necesar sa se cunoasca structura unui polinom, deoarece utilizatorul trebuie să introducă date valide pentru a se afisa un rezultat corect. Programul trebuie să funcționeze astfel: introducere polinoame, calcul, afisare rezultat.

*Pasii executiei fiecarui use-case:*

-in primul rând, user-ul trebuie sa introduca polinoamele;

-dupa introducerea polinoamelor, user-ul trebuie sa selecteze operația care dorecte să fie aplicata polinaomelor (adunare, scădere, înmulțire, împărțire, derivare sau integrare);

-dupa selectarea operatiei dorite se va afisa rezultatul;

-dupa afisarea rezultatului, user-ul urmează sa iasă din aplicație => exit

Sistemul prezentat este un calculator de polinoame, user-ul este cel care efectuează operații si cel care primeste un răspuns. Introducerea polinoamelor este realizata de user, precum și alegerea operației dorite și ieșirea din program. Rezultatul depinde de operația aleasa și se generează abia după ce una din operatii a fost selectată.

*Scenariu de utilizare:*

* User-ul introduce primul polinom, de exempul: 5x^3 + 2x^2 + 3
* După care urmează introducerea celui de-al doilea polinom, de exemplu: 2x^3 - 4x^1 - 1
* User-ul selectează , de exemplu, operația de adunare.
* Pe ecran urmează sa apara rezultatul adunării polinoamelor:

7x^3 + 2x^2 - 4x^1 + 2

* In final, user-ul iese din aplicație.

**Proiectare:**

Pentru proiectarea în limbaj OOP, s-a creat o diagrama de pachete după modelul MVC (Model-View-Controller).

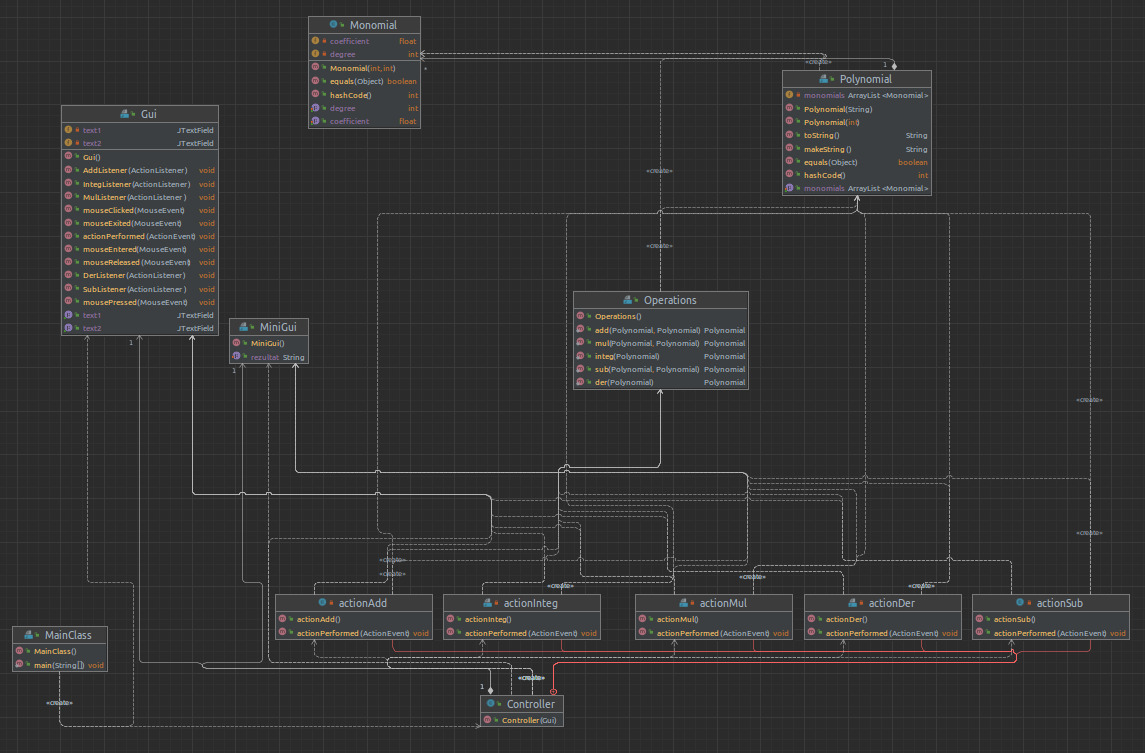
-pentru partea de Model am implementat urmatoarele clase:Monomial(aici am creat un constructor care are un coeficient și un exponent și am creat getter-ele și setter-ele necesare), Polynomial(aici am folosit ca structura de date <ArrayList> pentru a stoca monoamele ce formează polinomul, tot aici am folosit metoda regex care are rolul de a verifica dacă se respecta un anumita pattern, iar în final o metoda toString cu rolul de a crea un string care sa afiseze polinomul rezultat și are loc și initializarea cu 0 a coeficientilor array-ului folosit pentru polinomul rezultat cât și setarea gradelor de la gradul maxim pana la 0), Operations (aici am scris metodele care creeaza polinoamele necesare si implementeaza operatiile).

-pentru partea de View am creat doua clase (Gui și MiniGui) care contin doua frame-uri (unul în care se introduc polinoamele și unul în care se afiseaza rezultatul). Gui mai conține și butoane pentru tastarea cifrelor și operatorilor.

-pentru partea de Controller am creat ActionListenerii necesari.

Prin intermediul butoanelor pe care le apas, cele care executa operatii, se face legătura dintre Controller și clasele din Model și View.

În diagrama UML se pot vedea mai clar detaliile, structura proiectului cât și metodele folosite:



**Implementare:**

***=>Pachetul model:***

***(-)Clasa Polynomial:***

In clasa Polynomial am un ArrayList de monoame care formeaza polinoamele, cat si getter-ul si settar-ul necesar. Tot aici folosesc metoda Regex , in felul urmator: in constructor se adauga in ArrayList-ul de monoame, monoamele respectand pattern-ul dat si are loc si conversia din String in Float /Integer a coeficientilor si gradelor monoamelor respective. Tot in acest constructor se arunca o exceptie in cazul in care s-a primit un String gol ca parametru. Pentru al doilea constructor are loc initializarea cu 0 a coeficientilor array-ului folosit pentru polinomul rezultat cât și setarea gradelor de la gradul maxim(cel primit ca parametru) pana la 0. Metoda toString are rolul de a transforma polinomul rezultat in String, pentru scrierea in textfield. In final, metoda equals e folosita in cazul asta in clasa de test numita OperationsTest, de teste JUnit, pentru a verifica daca rezultatul obtinut e egal cu rezultatul asteptat, cu metoda assertEquals.

***(-)Clasa Monomial***

In clasa Monomial am declarat coeficientul monomului de tip float si exponentul de tip intreg, cat si constructorul aferent. Am creat getter-e si setter-e pentru coeficienti si exponenti. Si aici am metoda equals care e folosita in clasa de test pentru a verifica daca rezultatul obtinut e egal cu rezultatul asteptat, cu metoda assertEquals.

***(-)Clasa Operations***

In clasa Operations am 5 metode :

-Metoda ***add*** *--* primeste ca parametri doua polinoame si calculeaza rezultatul adunarii. Metoda functioneaza astfel: i si j retin gradul cel mai mare al polinoamelor primite ca parametru, dupa care se calculeaza maximul dintre ele si se construieste polinomul rezultat, care e un ArrayList, cu coeficientii 0 si gradele incepand cu gradul maxim pana la gradul 0. Urmeaza sa parcurg polinomul rezultat, cat si fiecare dintre polinoamele date ca parametrii, iar in cazul in care gradele coincid cu gradele polinomului rezultat se vor aduna coeficientii la coeficientii polinomului rezultat. In cazul in care exista in array coeficienti 0 se vor sterge monoamele respective. Daca, dupa asta nu exista absolut niciun monom in polinomul rezultat, se va crea un monom cu coeficientul si gradul 0, deci se va afisa 0.0 in textfield-ul rezultatului.

-Metoda ***sub*** *--* primeste ca parametri doua polinoame si calculeaza rezultatul scaderii. Metoda functioneaza astfel: i si j retin gradul cel mai mare al polinoamelor primite ca parametru, dupa care se calculeaza maximul dintre ele si se construieste polinomul rezultat, care e un ArrayList, cu coeficientii 0 si gradele incepand cu gradul maxim pana la gradul 0. Urmeaza sa parcurg polinomul rezultat, cat si fiecare dintre polinoamele date ca parametrii, iar in cazul in care gradele coincid cu gradele polinomului rezultat se vor aduna coeficientii primului polinom dat ca parametru si se vor scadea coeficientii celui de al doilea, la coeficientii polinomului rezultat. In cazul in care exista in array coeficienti 0 se vor sterge monoamele respective. Daca, dupa asta nu exista absolut niciun monom in polinomul rezultat, se va crea un monom cu coeficientul si gradul 0, deci se va afisa 0.0 in textfield-ul rezultatului.

-Metoda ***mul*** *--* primeste ca parametri doua polinoame si calculeaza rezultatul inmultirii. Metoda functioneaza astfel: i si j retin gradul cel mai mare al polinoamelor primite ca parametru, dupa care se construieste polinomul rezultat, cu gradul maxim egal cu suma gradelor maxime ale celor doua polinoame parametrii si coeficientii 0. Se parcurg atat polinoamele parametrii cat si polinomul rezultat, astfel incat sa se ia pe rand fiecare monom din primul polinom cu fiecare monom din al doilea polinom, iar daca gradele polinomului rezultat sunt egale cu suma dintre gradele celor doua polinoame, coeficientii polinomului rezultat vor fi egali cu produsul dintre coeficientii celor doua polinoame. In cazul in care exista in array coeficienti 0 se vor sterge monoamele respective. Daca, dupa asta nu exista absolut niciun monom in polinomul rezultat, se va crea un monom cu coeficientul si gradul 0, deci se va afisa 0.0 in textfield-ul rezultatului.

-Metoda ***der --*** primeste ca parametru un polinom,cel de pe primul textfield, si calculeaza derivata acestuia, in felul urmator: Se parcurge polinomul si se verifica daca gradele sunt diferite de 0. In caz afirmativ coeficientii devin egali cu produsul dintre acestia si gradele respective, iar gradele scad cu 1.In caz negativ coeficientii devin egali cu produsul dintre acestia si gradele respective, iar gradele raman la fel. In cazul in care exista in array coeficienti 0 se vor sterge monoamele respective. Daca, dupa asta nu exista absolut niciun monom in polinomul rezultat, se va crea un monom cu coeficientul si gradul 0, deci se va afisa 0.0 in textfield-ul rezultatului. Se return-eaza polinomul.

-Metoda ***integ*** -- primeste ca parametru un polinom,cel de pe primul textfield, si calculeaza integrala acestuia, in felul urmator: Se parcurge polinomul , coeficientii devin egali cu raportul dintre acestia si gradele respective adunate cu 1, iar gradele cresc cu 1. Se return-eaza polinomul.

***=>Pachetul view:***

***(-)Clasa Gui***

Clasa Gui mosteneste clasa JFrame si implementeaza interfetele ActionListener si MouseListener. Interfata MouseListener e folosita pentru a scrie in al doilea textfield atunci cand dau click in acesta, folosind butoane pentru cifre si operatori. Se folosesc conditii in metoda actionPerformed pentru a scrie intr-un singur textfield atunci cand folosesc butoanele. La apasarea butoanelor va scrie valoarea acestora in textfield-uri. Metodele AddListener, SubListener, MulListener, DerListener si IntegListener fac legatura dintre partea de view si cea controller astfel: la apasarea butonului cu operatia aferenta se va apela metoda addActionListener, apoi vom apela metodele(AddListener, SubListener, MulListener, DerListener si IntegListener) in clasa Controller si vom crea clase interne, pentru fiecare operatie, care implementeaza ActionListeneri.

***(-)Clasa MiniGui***

In clasa MiniGui am un textfield in care se afiseaza rezultatul si o metoda setRezultat care primeste si seteaza rezultatul in textfield.

***=>Pachetul controller:***

***(-)Clasa Controller***

In clasa Controller am, in constructor, apelurile metodelor din Gui (AddListener, SubListener, MulListener, DerListener si IntegListener) care creaza clase interne(actionAdd, actionSub, actionMul, actionDer, actionInteg) pentru fiecare operatie. Acestea implementeaza bineinteles ActionListeneri.

***(+)Clasa interna actionAdd***

Aici am doua string-uri care primesc textele inserate in textfield-urile din Gui. Dupa asta sunt create polinoame aferente. In polinomul rezultat se retine rezultatul adunarii celor doua polinoame. Polinomul rezultat este convertit in String si urmeaza sa fie setat in textfield-ul rezultat din MiniGui. Toate acestea sunt puse intr-o metoda try-catch pentru a arunca o exceptie in momentul in care nu se introduce nimic in textfield-uri (se afiseaza un mesaj intr-un JOptionPane).

***(+)Clasa interna actionSub***

Aici am doua string-uri care primesc textele inserate in textfield-urile din Gui. Dupa asta sunt create polinoame aferente. In polinomul rezultat se retine rezultatul scaderii celor doua polinoame. Polinomul rezultat este convertit in String si urmeaza sa fie setat in textfield-ul rezultat din MiniGui. Toate acestea sunt puse intr-o metoda try-catch pentru a arunca o exceptie in momentul in care nu se introduce nimic in textfield-uri (se afiseaza un mesaj intr-un JOptionPane).

***(+)Clasa interna actionMul***

Aici am doua string-uri care primesc textele inserate in textfield-urile din Gui. Dupa asta sunt create polinoame aferente. In polinomul rezultat se retine rezultatul inmultirii celor doua polinoame. Polinomul rezultat este convertit in String si urmeaza sa fie setat in textfield-ul rezultat din MiniGui. Toate acestea sunt puse intr-o metoda try-catch pentru a arunca o exceptie in momentul in care nu se introduce nimic in textfield-uri (se afiseaza un mesaj intr-un JOptionPane).

***(+)Clasa interna actionDer***

Aici am un string care primesc textul inserat in primul textfield din Gui. Dupa asta este creat polinomul aferente. In polinomul rezultat se retine rezultatul derivarii polinomului. Polinomul rezultat este convertit in String si urmeaza sa fie setat in textfield-ul rezultat din MiniGui. Toate acestea sunt puse intr-o metoda try-catch pentru a arunca o exceptie in momentul in care nu se introduce nimic in textfield-uri (se afiseaza un mesaj intr-un JOptionPane).

***(+)Clasa interna actionInteg***

Aici am un string care primesc textul inserat in primul textfield din Gui. Dupa asta este creat polinomul aferente. In polinomul rezultat se retine rezultatul integrarii polinomului. Polinomul rezultat este convertit in String si urmeaza sa fie setat in textfield-ul rezultat din MiniGui. Toate acestea sunt puse intr-o metoda try-catch pentru a arunca o exceptie in momentul in care nu se introduce nimic in textfield-uri (se afiseaza un mesaj intr-un JOptionPane).

***+Clasa MainClass***

In clasa MainClass am creat un obiect g de tip Gui si un obiect controller de tip Controller.

**Rezultate:**

Pentru testare am folosit **JUnit** și am creat clasa OperationsTest. În această clasa am făcut câte o metoda de test pentru fiecare operație. Am creat doua Polinoame pentru scadere, adunare si inmultire. Am creat un al treilea polinom(expected) care reprezintă rezultatul pe care ma astept sa il primesc in urma efectuarii operatiei respective. In *acctual* se retine rezultatul operatiei efectuate pe cele doua polinoame date initial. Am folosit funcția assertEquals pentru a testa dacă rezultatul obținut este egal cu al treilea polinom creat(expected). Am procedat la fel și pentru celelalte operații (derivare si integrare), dar creand doar un polinom pe care urmeaza sa fie efectuata operatia.

O imagine care conține text

Descriere generată automat

**Concluzii:**

Proiectul ar trebui sa funcționeze pentru orice caz introdus de user.

Am folosit pentru prima data metodele regex și am învățat cum se folosesc și cum pot verifica dacă un text respecta anumite criterii. De asemenea, am învățat cum se încarca proiecte pe platforma GitLab și mi-am dat seama cum trebuie să arate o structura de proiect de tip MVC.

**Bibliografie:**

Regex : <https://stackoverflow.com/questions/36490757/regex-for-polynomial-expression>

*UML Diagram:* [*https://youtu.be/v8G7FWdiwqs*](https://youtu.be/v8G7FWdiwqs)