

DOCUMENTAȚIE

TEMA 1

NUME SI PRENUME: MOȘILĂ LUCIANA

GRUPA: 30226

CUPRINS

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

1. Obiectivul temei

Obiectivul principal este crearea unei aplicatii numita "Calculator de polinoame" care sa permita adunarea, scaderea, inmultirea, integrarea si derivarea polinoamelor introduse de utilizator.

Obiectivele secundare includ dezvoltarea unei interfete prietenoase, implementarea algoritmilor si testarea aplicatiei

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

Cerințe funcționale:

Clasa Polinom este capabilă să reprezinte un polinom ca șir și ca un Map de coeficienți și exponenți.

Clasa Gui utilizeaza clasa Polinom si permite transformarea datelor adaugate de utilizator din format Sring (operator + sau -) ax^n intr-un HashMap ce salveaza coeficientul intr-o adresa(putere).

Cerințe nefuncționale:

Clasa Polinom:

- trebuie să fie scrisă în Java și să respecte convențiile de denumire Java.
- trebuie să fie modulară și ușor de înțeles, citit și întreținut.

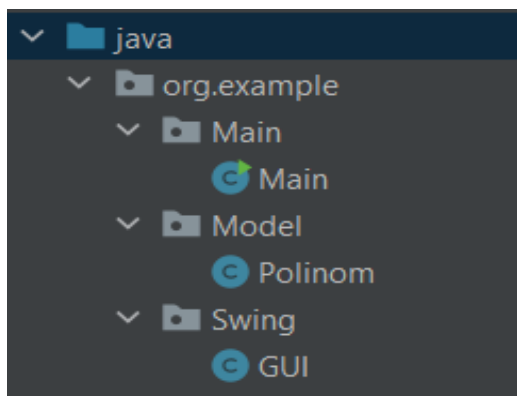
- trebuie să aibă teste unitare care să acopere toate funcționalitățile și cazurile marginale.
- trebuie să fie eficientă în ceea ce privește utilizarea memoriei și timpul de execuție.

În continuare, vor fi prezentate câteva cazuri de utilizare ale acestei clase:

1. Crearea unui polinom dintr-un șir de caractere
2. Conversia unui polinom în șir de caractere
3. Adunarea a două polinoame

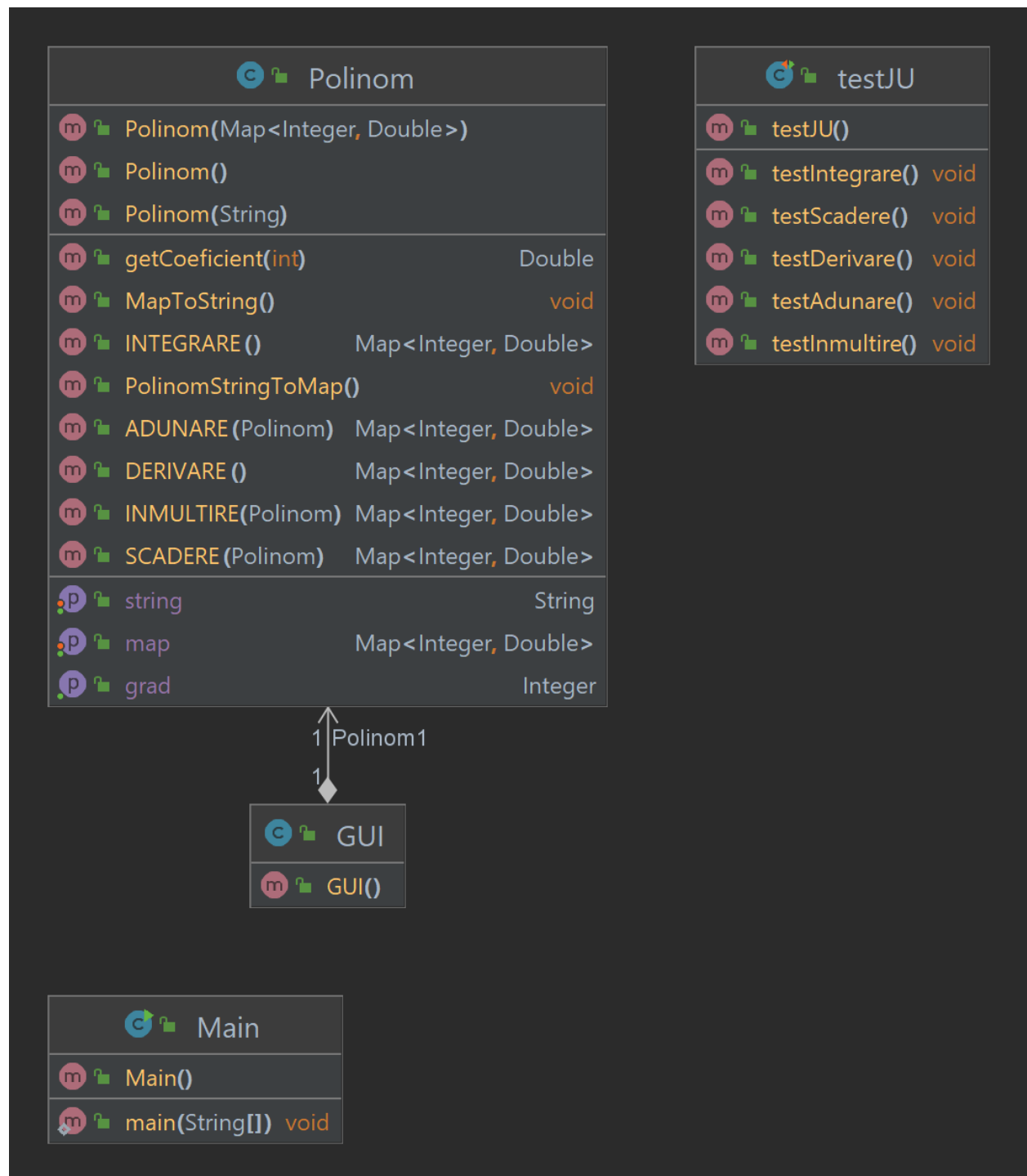
3. Proiectare

Proiectul “Calculator de polinoame” este structurat în Pachete ca în imaginea data mai jos.



Pachetul “Main” conține clasa principală Main, pachetul “Model” conține clasa Polinom, iar clasa GUI aparține pachetului “Swing”, toate aceste pachete cu clase, sunt introduse în pachetul principal “org.example”.

Diagrama UML de clase este urmatoarea:



4. Implementare

Implementarea consta intr-o clasă Java numită "Polinom" care conține câteva metode utile pentru a lucra cu polinoame. Clasa are trei câmpuri private: "stringPolinom", care este un șir de caractere care reprezintă polinomul, "mapPolinom", care este un map ce reține perechi (putere, coeficient) pentru fiecare termen din polinom.

Clasa "Polinom" are trei constructori diferiți: primul constructor este cel fără parametri și inițializează câmpurile clasei cu valorile implicite, al doilea constructor primește un șir de caractere și inițializează stringPolinom cu valoarea șirului dat, iar al treilea constructor primește un map și inițializează mapPolinom cu acesta.

Clasa are mai multe metode publice importante, printre care:

getMap() și getString(), care returnează map-ul și șirul de caractere care reprezintă polinomul, respectiv;

PolinomStringToMap() și MapToString(), care transformă șirul de caractere stringPolinom în mapPolinom și invers, respectiv;

ADUNARE(Polinom polinom) și SCADERE(Polinom polinom), care calculează suma și diferența a doua polinoame, respectiv.

Derivarea: DERIVARE(). Această metodă calculează derivata polinomului curent și o returnează sub formă de obiect Polinom.

Înmulțirea: INMULTIRE(Polinom polinom2). Această metodă calculează produsul dintre polinomul curent (this) și polinom2 și returnează rezultatul stocat într-un Map.

Integrarea: INTEGRARE(). Această metodă calculează integrala polinomului curent și o returnează.

Clasa GUI reprezintă interfața grafică a aplicației. Aceasta utilizează framework-ul Swing pentru a crea și afișa elementele grafice, cum ar fi câmpuri de text, butoane și etichete, și permite utilizatorului să interacționeze cu aplicația.

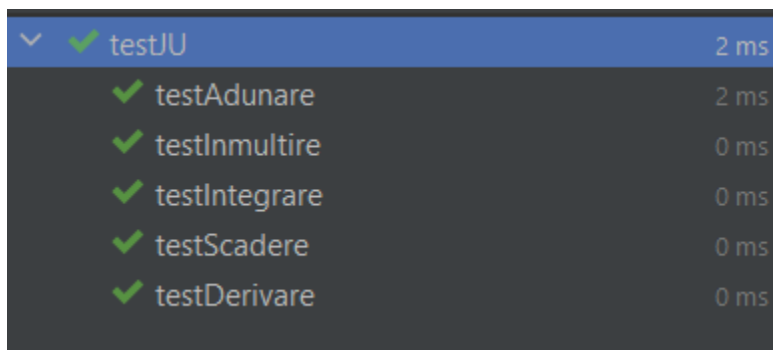
În această clasă sunt definite câmpurile de text și butoanele necesare pentru a introduce și a afișa polinoame, precum și obiecte ale clasei Polinom pentru a efectua operațiile necesare asupra acestora.

Fiecare buton are un ascultător de evenimente asociat care se declanșează atunci când butonul este apăsător. În acest ascultător, se crează obiecte Polinom pe baza polinoamelor introduse în câmpurile de text, se efectuează operațiile corespunzătoare și se afișează rezultatul în câmpul de text corespunzător.

În plus, clasa GUI definește și aspectul general al interfeței grafice, cum ar fi dimensiunea și poziția ferestrei principale.

5.Rezultate

Rezultate testari cu Junit sunt urmatoarele:



▼	✓ testJU	2 ms
	✓ testAdunare	2 ms
	✓ testInmultire	0 ms
	✓ testIntegrare	0 ms
	✓ testScadere	0 ms
	✓ testDerivare	0 ms

Pentru fiecare functie in parte am construit un polinom, respectiv 2 polinoame, unde este cazut, cu date de intrare specifice sub forma unui HashMap si am format obiectele de tip Polinom cu acestea. Am apelat la fiecare test functia coresounzatoare si am asteptat un raspuns sub forma unui String ce reprezinta polinomul rezultat.

6.Concluzii

Proiectarea unui calculator de polinoame este o experiență utilă pentru dezvoltarea abilităților matematice și tehnologice.

Calculatorul de polinoame poate fi îmbunătățit în continuare prin adăugarea de noi funcționalități și îmbunătățirea performanței.

7. Bibliografie

1. <https://www.guru99.com/junit-parameterized-test.html>
2. https://users.utcluj.ro/~igiosan/teaching_poo.html
3. [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
4. <https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP>