

DOCUMENTATIE

TEMA 1

NUME STUDENT: TRIF GABRIELA-IOANA
GRUPA: 30221

CUPRINS

CUPRINS.....	2
1. Obiectivul temei.....	3
2. Analiza problemei, modelare, scenarii, cazuri de utilizare	3
3. Proiectare	5
4. Implementare	6
Clasa Functions	6
Clasa Polynom	7
Clasa PolynomialCalculator.....	8
Clasa PolynomialOperationsTest.....	8
5. Rezultate	8
6. Concluzii	9
7. Bibliografie	9

1. Obiectivul temei

Obiectivul principal al acestei teme este realizarea operațiilor matematice cu polinoame.

Obiectivele secundare sunt:

- Adunarea
- Scăderea
- Înmulțirea
- Împărțirea
- Derivarea
- Integrarea

Obiectivele secundare reprezintă pașii care construiesc obiectivul principal al calculatorului de polinoame. Toate operațiile reprezintă obiectivele ale temei și vor fi prezentate în detaliu în capitolele 2 și 3.

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

Cerințele funcționale ale unui calculator de polinoame sunt:

- Adunarea a două polinoame
- Scăderea a două polinoame
- Înmulțirea a două polinoame
- Împărțirea a două polinoame
- Derivarea unui polinom
- Integrarea unui polinom

Cerințele non-funcționale sunt:

- Interfața ar trebui să fie ușor de înțeles pentru utilizator
- Programul nu ar trebui să producă rezultate incorecte
- Sistemul ar trebui să fie capabil să gestioneze un număr mare de termeni dintr-un polinom

Use-case-uri:

- Adunare
- Scădere
- Înmulțire
- Împărțire
- Derivare

- Integrare

Exemplu:

Descriere use-case: adunare

Actor: user

Descriere: Utilizatorul introduce două polinoame, selectează butonul „add”. Sistemul va afișa rezultatul pe ecran.

Prima problemă pe care o întâmpinăm este stocarea polinoamelor. Am ales să stocăm polinoamele într-un TreeMap, în care cheia reprezintă exponentul, iar valoarea reprezintă coeficientul (numere stocate ca și integer).

Pentru adunare am realizat o metodă statică care ia ca parametri două obiecte Polynom și returnează un Polynom. Scopul metodei este de a adăuga al doilea obiect Polynom la primul și de a returna rezultatul.

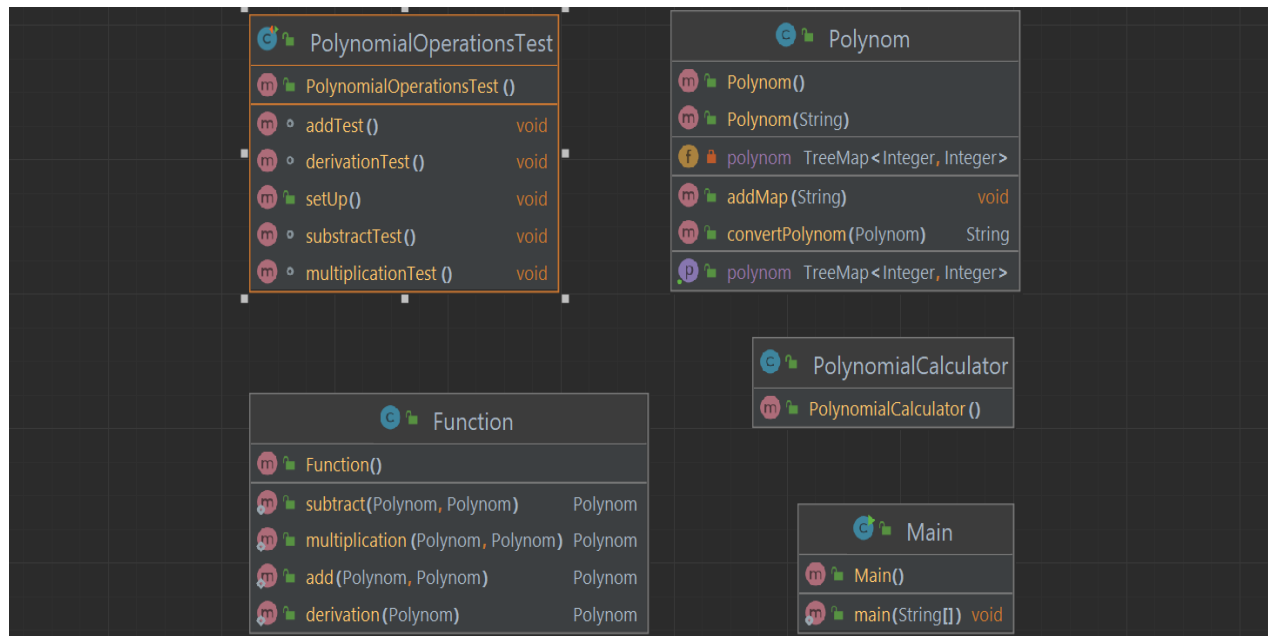
Scăderea folosește o metodă statică numită *subtract* care ia ca parametri două obiecte Polynom și returnează un Polynom. Scopul metodei este de a scădea al doilea obiect Polynom din primul și de a returna rezultatul.

Înmulțirea polinoamelor se face cu ajutorul metodei *multiplication* care este o metodă statică care ia ca parametri două obiecte Polynom și returnează un Polynom. Scopul metodei este de a multiplica cele două obiecte Polynom și de a returna rezultatul.

Operația care realizează împărțirea a două polinoame și apoi returnează rezultatul sub forma unui polinom are ca parametri de intrare două polinoame.

Derivarea și integrarea polinoamelor este relativ simplă, se rezumă doar la parcurgerea polinomului, la înmulțirea/împărțirea coeficienților și scăderea/adunarea gradului cu o unitate.

3. Proiectare



Clasa Polynom:

- Conține TreeMap-ul în care se stochează un polinom
- Metoda de conversie din String în Polynom
- Metoda de conversie din Polynom în String

Clasa Function

- Metode pentru toate operațiile

Clasa PolynomialCalculator

- Conține implementarea interfeței grafice

Packages

- Un package pentru clasa de controller, care controlează interfața "PolynomialCalculator"
- Un package pentru tot ce are legătură cu clasa Polynom
- Un package care stochează implementarea operațiilor
- Un package pentru clasa care implementează testele

Structuri de date

- Folosim un TreeMap pentru a stoca coeficienții și exponenții prezenți într-un polinom.

4. Implementare

Clasa Functions

Adunarea

Am implementat această operație folosind metoda static numită *add*. Metoda realizează acest lucru prin iterarea fiecărui termen din al doilea obiect Polynom (stocat ca TreeMap) și adăugându-l la primul Polynom. Dacă un termen cu același exponent există deja în primul polinom, se adună coeficienții celor doi termeni. Dacă coeficientul rezultat este zero, termenul este eliminat din polinom. Dacă termenul nu există în primul polinom, se adaugă pur și simplu. Metoda modifică primul obiect Polynom și îl returnează, astfel încât obiectul Polynom original este schimbat prin această metodă.

Scăderea

Este implementată cu ajutorul unei metode statice. Metoda realizează acest lucru prin iterarea fiecărui termen din al doilea obiect Polynom și scăderea acestuia din termenul corespunzător din primul Polynom. Dacă un termen cu același exponent există deja în primul polinom, se scad coeficienții celor doi termeni. Dacă coeficientul rezultat este zero, termenul este eliminat din polinom. Dacă termenul nu există în primul polinom, se adună cu coeficientul său înmulțit cu -1.

Înmulțirea

Metoda statică *multiplication* realizează acest lucru prin iterarea fiecărui termen din primul obiect Polynom (stocat ca TreeMap) și apoi iterarea peste fiecare termen din al doilea obiect Polynom. Apoi înmulțește coeficienții fiecărui termen și adună exponenții. Termenul rezultat este apoi adăugat la un nou obiect Polynom, *p*. Dacă un termen cu același exponent există deja în *p*, se adună coeficienții celor doi termeni. Metoda creează un nou obiect Polynom și îl returnează, astfel încât obiectele Polynom originale nu sunt modificate prin această metodă.

Derivarea

Operația este implementată utilizând o metodă statică numită *derivation* care ia un obiect Polynom ca parametru și returnează un Polynom. Scopul metodei este de a lua derivata polinomului și de a returna rezultatul. Metoda realizează acest lucru prin iterarea fiecărui termen din obiectul Polynom și calculând derivata fiecărui termen. Derivata unui termen se obține prin înmulțirea exponentului (cheia din TreeMap) cu coeficientul (valoarea din TreeMap), iar apoi prin reducerea exponentului cu 1. Termenul rezultat este apoi adăugat la un nou obiect Polynom, p . Dacă exponentul rezultat este negativ, termenul nu este adăugat la noul obiect Polynom, deoarece termenii cu exponenți negativi nu sunt validi într-un polinom. Metoda nu modifică obiectul original Polynom.

Clasa Polynom

Această clasă reprezintă un polinom. Polinomul este reprezentat intern ca un TreeMap unde cheile sunt exponenții polinomului, iar valorile sunt coeficienții. Metoda addMap ia un String care reprezintă un polinom și îl convertește într-un TreeMap. Metoda convertPolynom preia un obiect Polynom și returnează reprezentarea lui ca String. Clasa Polynom are un singur constructor care ia un String care reprezintă un polinom.

Metoda addMap este apelată de la constructor pentru a inițializa TreeMap intern. Metoda addMap folosește expresii regulate pentru a analiza șirul de intrare care reprezintă polinomul. Face bucla peste subșirurile potrivite și pentru fiecare subșir, extrage coeficientul și exponentul termenului și îl adaugă în TreeMap.

Metoda convertPolynom trece peste intrările din TreeMap intern și construiește reprezentarea șir a polinomului. Se ocupă de cazuri speciale pentru coeficienți de 1 sau -1 și exponenți de 0 sau 1. Returnează șirul rezultat.

În general, clasa Polynom oferă o modalitate convenabilă de a lucra cu polinoame în Java, permițând conversia din șiruri de caractere în obiecte polinomiale și invers.

Clasa PolynomialCalculator

Această clasă definește o interfață GUI simplă pentru un calculator de polinoame, în care utilizatorul poate introduce două polinoame și poate efectua operații precum adunare, scădere, înmulțire, împărțire, derivare și integrare. GUI constă dintr-un JFrame, mai multe JTextFields pentru intrare și ieșire și mai multe JButtons pentru fiecare operație. Fiecare JButton are un ActionListener care ascultă clic-uri și efectuează operația corespunzătoare pe baza polinoamelor de intrare. Rezultatul este afișat în resultField JTextField. Codul definește, de asemenea, clasele Polynom și Function, care sunt folosite pentru a reprezenta polinoame și, respectiv, pentru a efectua operații polinomiale.

Clasa PolynomialOperationsTest

Este o clasă de testare Junit pentru testarea operațiilor implementate în clasa Functions. Clasa conține patru metode de testare pentru adunarea, scăderea, înmulțirea și derivarea a două polinoame.

Fiecare dintre cele patru metode de testare efectuează o operație polinomială utilizând metoda corespunzătoare din clasa Funcție și compară rezultatul cu un rezultat așteptat predefinit. Rezultatul așteptat este definit folosind clasa Polynom și este stocat în variabila resultCorrect. Clasa Assertions este folosită pentru a compara rezultatele reale și cele așteptate. În fiecare metodă de testare, există două afirmații. Prima afirmație verifică dacă rezultatul real se potrivește cu rezultatul așteptat, iar a doua afirmație verifică dacă rezultatul real nu se potrivește cu un rezultat așteptat diferit.

Adnotarea @BeforeEach este utilizată pentru a defini o metodă care va rula înainte de fiecare metodă de testare.

5. Rezultate

Cum se poate observa prin teste, obținem rezultate corecte pentru orice operație pe care vrem să o efectuăm.

Testele cuprind orice caz posibil, deci pentru orice modificare efectuată asupra programului acestea ne pot asigura de funcționalitatea completă a programului.

6. Concluzii

În concluzie, programul nostru efectuează toate operațiile și are o interfață prietenoasă cu utilizatorul.

7. Bibliografie

<https://dsrl.eu/courses/pt/>

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

<https://stackoverflow.com/questions/40860056/coding-polynomial-division>