

DOCUMENTATIE

TEMA *1*

NUME STUDENT: Anastasiu Andreea-Valentina
GRUPA: 30227

CUPRINS

1.	Obiectivul temei.....	3
2.	Analiza problemei, modelare, scenarii, cazuri de utilizare	3
3.	Proiectare	4
4.	Implementare	4
5.	Rezultate	9
6.	Concluzii.....	10
7.	Bibliografie	10

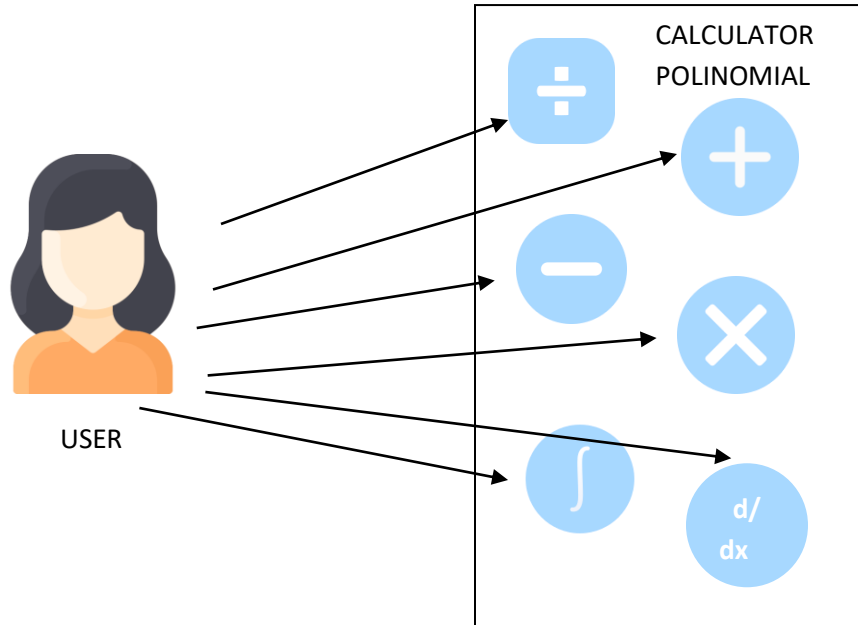
1. Obiectivul temei

Obiectivul principal al proiectului este proiectarea și implementarea unui calculator de polinoame cu o interfață grafică dedicată, prin intermediul căreia utilizatorul poate introduce polinoame și de a selecta operațiile matematice apoi a vizualiza rezultatele.

Sub-obiectivele proiectului sunt:

- a. Analiza Cerintelor
Realizarea unei analize a problemei pentru identificarea functionalitatii necesare și a cerintelor utilizatorului pentru calculul polinoamelor.
- b. Proiectarea
Dezvoltarea unui design pentru calculatorul de polinoame care să includă aspecte legate de cum arată interfața utilizatorului, metodele de introducere a datelor, selecția operațiilor și prezentarea rezultatelor.
- c. Implementarea:
Realizarea designului prin codificarea calculatorului de polinoame
- d. Testarea:
Testarea a calculatorului de polinoame pentru verificarea functionalitatii acestuia, gestionarea cazurilor speciale.

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



Cerințe funcționale:

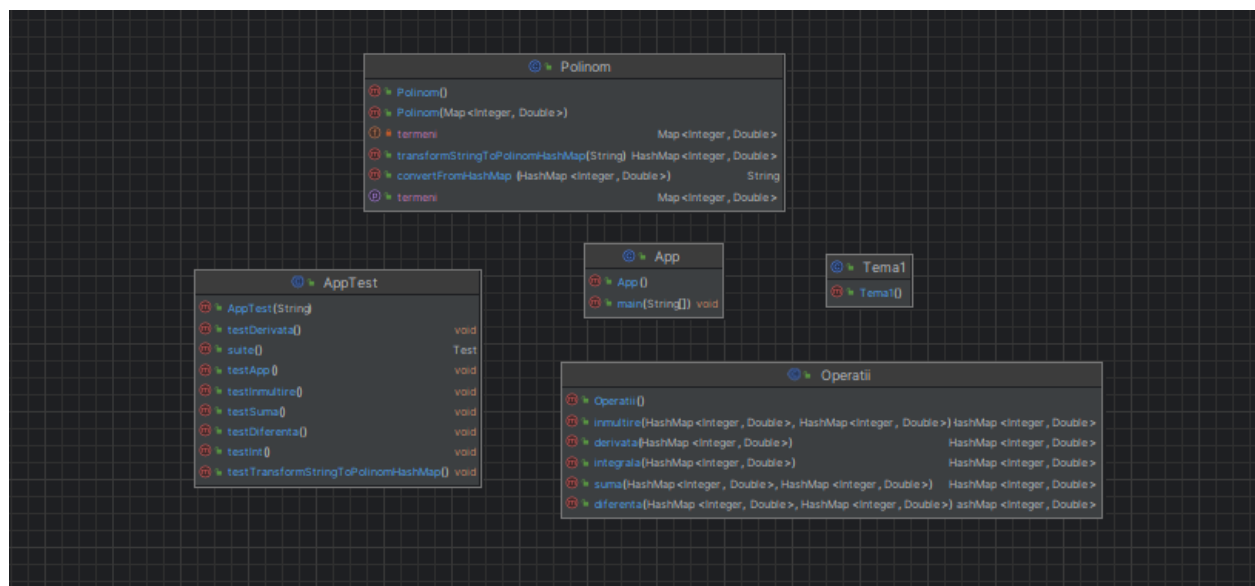
- Calculatorul de polinoame ar trebui să permită utilizatorilor să introducă polinoame.
- Calculatorul de polinoame ar trebui să permită utilizatorilor să selecteze operația matematică.
- Calculatorul de polinoame ar trebui să adune două polinoame.

- Calculatorul de polinoame ar trebui să permită utilizatorilor să scadă două polinoame.
- Calculatorul de polinoame ar trebui să permită utilizatorilor să înmulțească un polinom cu un scalar.
- Calculatorul de polinoame ar trebui să permită utilizatorilor să înmulțească două polinoame.
- Calculatorul de polinoame ar trebui să permită utilizatorilor să împartă un polinom la un scalar nenul.
- Calculatorul de polinoame ar trebui să permită utilizatorilor să împartă un polinom la alt polinom, obținând atât câtul, cât și restul.
- Calculatorul de polinoame ar trebui să permită utilizatorilor să deriveze un polinom.
- Calculatorul de polinoame ar trebui să permită utilizatorilor să integreze un polinom.

Cerinte non-funcționale:

Calculatorul polinomial trebuie să fie intuitiv și ușor de utilizat de către utilizator.

3. Proiectare



4. Implementare

CLASA POLINOM

Clasa Polinom oferă funcționalități pentru transformarea între reprezentările sub formă de șir de caractere și sub formă de HashMap, precum și pentru accesul și modificarea termenilor polinomului.

Variabile de instanță și constructor:

private Map<Integer, Double> termeni: Această variabilă de instanță stochează termenii polinomului sub formă de perechi cheie-valoare, unde cheia reprezintă exponentul și valoarea reprezintă coeficientul corespunzător.

Constructorul Polinom(): Inițializează o instanță a clasei Polinom cu un HashMap gol pentru stocarea termenilor.

Constructorul Polinom(Map<Integer, Double> termeni): Inițializează o instanță a clasei Polinom cu un HashMap dat ca parametru, care conține termenii polinomului.

Metode pentru acces și modificare:

getTermeni(): Returnează HashMap-ul care conține termenii polinomului.

setTermeni(Map<Integer, Double> termeni): Setează HashMap-ul care conține termenii polinomului la o valoare dată.

Metoda transformStringToPolinomHashMap(String strPolinom):

Această metodă primește un șir de caractere ce reprezintă un polinom sub forma unei expresii, și transformă această expresie într-un HashMap care stochează termenii polinomului sub formă de perechi cheie-valoare.

Utilizează expresii regulate pentru a identifica termenii polinomului în șirul dat și îi procesează pentru a obține coeficienții și puterile.

Valorile obținute sunt apoi introduse în HashMap și returnate.

Metoda convertFromHashMap(HashMap<Integer, Double> polinom):

Această metodă primește un HashMap care conține termenii polinomului și îl transformă înapoi într-un șir de caractere reprezentând polinomul.

Parcurge HashMap-ul și construiește șirul rezultat prin adăugarea termenilor în funcție de coeficient și putere.

Asigură că semnul este afișat corect pentru fiecare termen și că termenii cu coeficientul zero nu sunt incluși.

Returnează șirul rezultat, reprezentând polinomul sub forma unei expresii.

CLASA OPERATII

Clasa Operatii are diferite functionalitati pentru efectuarea unor operații matematice de bază pe polinoame, precum suma, diferența, înmulțirea, derivarea și integrarea.

Metoda suma(HashMap<Integer, Double> polinom1, HashMap<Integer, Double> polinom2):

Această metodă primește două polinoame sub forma unor HashMap-uri, polinom1 și polinom2, și returnează suma lor.

Parcurge gradele primului polinom și adaugă coeficienții corespunzători într-un HashMap. Dacă un exponent din polinom2 nu există în polinom1, se adaugă și coeficientul corespunzător din polinom2.

Metoda diferenta(HashMap<Integer,Double> polinom1, HashMap<Integer,Double> polinom2):

Această metodă primește două polinoame sub forma unor HashMap-uri, polinom1 și polinom2, și returnează diferența lor.

Parcurge gradele primului polinom și, pentru fiecare exponent, calculează diferența dintre coeficientul corespunzător din polinom1 și polinom2. Dacă un exponent din polinom2 nu există în polinom1, se adaugă în rezultat cu coeficientul negativ.

Metoda inmultire(HashMap<Integer,Double> polinom1, HashMap<Integer,Double> polinom2):

Această metodă primește două polinoame sub forma unor HashMap-uri, polinom1 și polinom2, și returnează rezultatul înmulțirii lor.

Folosește o buclă dublă pentru a itera prin toți termenii ambelor polinoame și efectuează înmulțirea fiecărui termen cu fiecare termen din celălalt polinom. Rezultatele sunt stocate într-un HashMap.

Metoda derivata(HashMap<Integer,Double> polinom1):

Această metodă primește un polinom sub forma unui HashMap și returnează derivata acestuia.

Parcurge gradele polinomului și calculează coeficienții derivatelor pentru fiecare termen. Aceștia sunt stocați într-un HashMap, unde cheile reprezintă gradele derivatelor.

Metoda integrala(HashMap<Integer,Double> polinom1):

Această metodă primește un polinom sub forma unui HashMap și returnează integrala acestuia.

Parcurge gradele polinomului și calculează coeficienții integralelor pentru fiecare termen. Aceștia sunt stocați într-un HashMap, unde cheile reprezintă gradele integralelor.

CLASA TEMA1

Tema1 reprezintă interfața grafică a unei aplicații care permite utilizatorului să introducă polinoame și să efectueze diferite operații matematice pe acestea prin intermediul butoanelor.

Variabile private:

private JPanel panel1: Un panou în care sunt puse toate componentele interfeței grafice.

private JTextField textfieldpol1: Un câmp text pentru introducerea primului polinom.

private JTextField textField1: Un câmp text pentru introducerea celui de al doilea polinom.

Butoane pentru diferite operații (adunare, scădere, înmulțire, împărțire, integrare, derivare).

private JTextField textField2: Un câmp text pentru afișarea rezultatului operației.

Constructorul Tema1():

Se initializeaza fereastra grafică (JFrame) cu dimensiuni, titlu și modul de închidere.

Se face fereastra vizibilă și se ajustează dimensiunea acesteia.

Evenimente pentru butoane:

Fiecare buton are un ActionListener care definește acțiunile care trebuie executate atunci când butonul este apăsat.

În fiecare eveniment, se efectuează următoarele acțiuni:

Se creează o instanță a clasei Polinom pentru a accesa metodele acesteia.

Se obțin polinoamele din câmpurile de text și se transformă în HashMap-uri folosind metoda transformStringToPolinomHashMap din clasa Polinom.

Se apelează metoda corespunzătoare din clasa Operatii pentru a efectua operația selectată (adunare, scădere, etc.).

Rezultatul este convertit înapoi într-un șir de caractere folosind metoda convertFromHashMap din clasa Polinom și afișat în câmpul de text textField2.

CLASA APPTest

Clasa AppTest extinde TestCase, o clasă de testare unitară din framework-ul JUnit.

Constructorul AppTest(String testName) apelează constructorul clasei de bază TestCase cu numele testului.

Metoda suite() returnează o instanță a testelor definite în această clasă.

Metoda testApp() este un test simplu care verifică dacă valoarea booleană este adevărată. Acest test este pentru a asigura corectitudinea testului.

Metodele de testare pentru metodele din clasele Polinom și Operatii:

În fiecare metodă de testare, pregătim datele , polinom1/polinom2 si un polinom care ar trebuie sa ne dea rezultatul, iar apoi apelam metoda din Operatii.

Rezultatul obținut este comparat cu rezultatul așteptat folosind metoda assertEquals pentru a vedea daca se fac calculele corecte.

5. Rezultate

Pentru a verifica daca scrierea operatiilor a fost corecta , am folosit Junit si am implementat metodele de testare in clasa APPTest. Am ales sa folosesc testarea parametrizata , transmitand cele doua polinoame si rezultatul dorit ca imputuri .

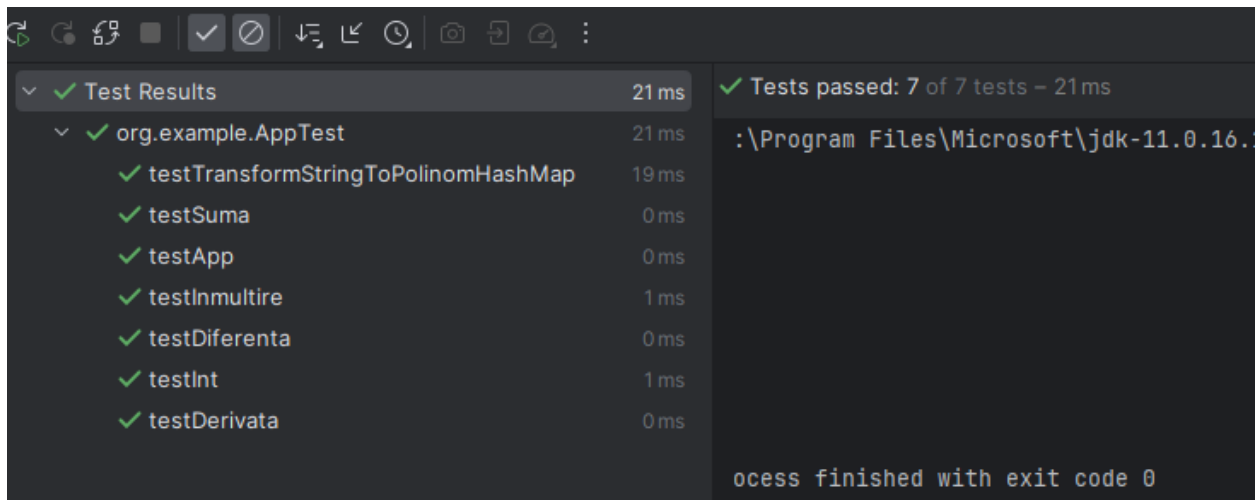
```
public void testDiferenta() {
    HashMap<Integer, Double> polinom1 = new HashMap<>();
    polinom1.put(3, 2.0);
    polinom1.put(1, 5.0);

    HashMap<Integer, Double> polinom2 = new HashMap<>();
    polinom2.put(2, -3.0);
    polinom2.put(1, 3.0);
    polinom2.put(0, -7.0);

    Map<Integer, Double> asteptat = new HashMap<>();
    asteptat.put(3, 2.0);
    asteptat.put(2, 3.0);
    asteptat.put(1, 2.0);
    asteptat.put(0, 7.0);

    Map<Integer, Double> result = new Operatii().diferenta(polinom1, polinom2);

    assertEquals(asteptat, result);
}
```



Test Results	Time
✓ org.example.AppTest	21 ms
✓ testTransformStringToPolinomHashMap	19 ms
✓ testSuma	0 ms
✓ testApp	0 ms
✓ testInmultire	1 ms
✓ testDiferenta	0 ms
✓ testInt	1 ms
✓ testDerivata	0 ms

✓ Tests passed: 7 of 7 tests – 21 ms

:\\Program Files\\Microsoft\\jdk-11.0.16..

rocess finished with exit code 0

6. Concluzii

Pot spune ca dupa terminarea temei am invatat destul de multe lucruri. Acestea constau in faptul ca am lucrat pentru prima data cu HashMap-uri, cu Regex si Junit. Am invatat ce fac anumite metode si cum pot sa sustrag anumite string-uri dintr-un string mai mare.

7. Bibliografie

- ✓ [Fundamental Programming Techniques \(dsrl.eu\)](https://dsrl.eu)
- ✓ [regex - Learning Regular Expressions - Stack Overflow](https://stackoverflow.com/questions/1732349/regex-learning-regular-expressions)
- ✓ [geeksforgeeks.org/java-util-hashmap-in-java-with-examples](https://www.geeksforgeeks.org/java-util-hashmap-in-java-with-examples)