**Proiect Colectiv**

**Tema: Tool Matematic – Generator de Polinoame de grad N**

**Universitatea de Vest din Timișoara**

**Facultatea de Matematică și Informatică**

**Coordonator: Leonard Mada, Syonic**

**Membri echipă:**

* **Dorin Doncenco**
* **Adrian Șereș**
* **Eduard Mirciov**

**Scop**

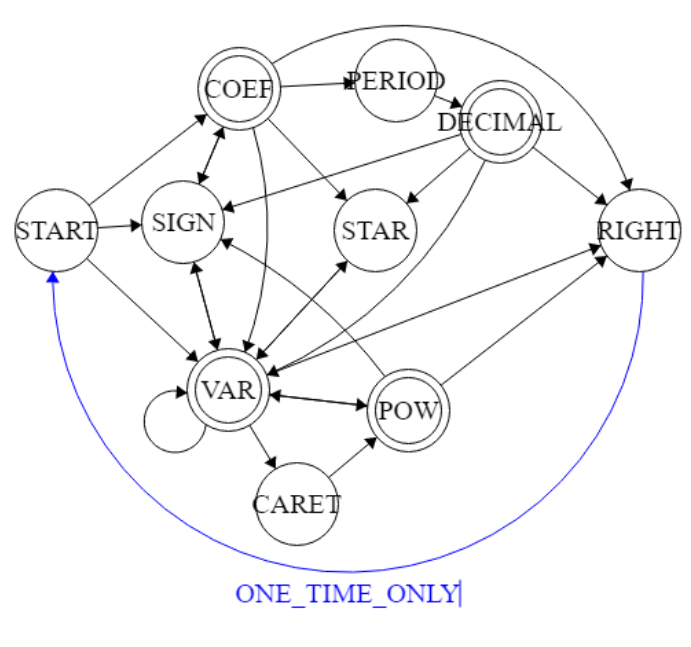
Proiectul are ca scop parsarea unui șir de caractere pentru a converti și a-l stoca într-o structură de date, în cazul de față aceasta fiind un polinom (Polynom), și generarea de polinoame primind ca input un grad (număr întreg) și un o rădăcină Polynom (ce este un String parsat).

**Implementare**

* **Structura polinoamelor**

Codul este implementat utilizând limbajul Java. Polinoamele sunt stocate folosind un TreeMap<Term,Double>, unde Double reprezintă coeficientul unui termen din polinom (i.e. pentru termenul ”2\*x^3\*y” coeficientul este 2) și Term reprezintă un TreeMap<String, Integer>, unde String reprezintă numele variabilei (cum ar fi a, b, k, x, y) și Integer este puterea variabilei (pentru x^3 avem putere = 3).

* **Parsarea polinomului prin automat de stări finite**



* **Generarea polinoamelor (pseudocod)**

function generatePolynom(int n, Polynom root){

int counter = 1, m = n

Polynom result;

Polynom b[m]

b[m] = 0

// rational terms are polynoms whose power of k is a multiple of n (0, n, 2\*n, etc)

Polynom toBeRationalised = root

while(m > 0)

{

b[m-1] = n \* (rationalTerms(toBeRationalised) / counter

Polynom poly = Polynom(“x^m-1”)

result.add(b[m-1] \* poly)

toBeRationalised = (toBeRationalised - b[m-1]) \* root

counter++

m--

}

b[n] = Polynom(“x^n”)

result.add(b[n])

return result

}

* **Interfața pentru utilizator**

În acest proiect, GUI-ul este un instrument care ne aduce beneficii de utilizare a metodelor care se întrepătrund în spatele liniilor de cod. Astfel prin includerea unor librării specifice și crearea unor obiecte de tipul de clasă Polynom ne putem utiliza de o interfață pe care sunt amplasate câteva butoane specifice și un text de care ne ajutăm pentru a crea polinoame noi. Astfel încât existența acelor butoane sunt constituite pe baza unor funcții din proiect care determină rezultatele de vigoare.

În primul rând interfața este creată prin intermediul clasei JFrame care este un container de nivel înalt. Este o extensie a versiunii din java.awt.Frame care adaugă ca și suport componenta pentru arhitectura JFC/SWING. Pe această interfață se află un panou JPanel care este un container intermediar, adică face legătura dintre butoane, text box, și interfață . Pe acesta sunt amplasate toate butoanele care creează mesaje sau alte interfețe pentru afișarea operațiilor corecte.

În al doilea rând căsuța principală de text este prima care se observă, sub un text specific de introducere, în momentul rulării programului, pe aceea orice tip de polinom pe care îl vom introduce cu ajutorul butonului “ADD” se salvează automat într-un ArrayList de tipul clasei Polynom. Butonul getPolynom deschide o nouă interfață în care sunt salvate toate elementele din ArrayList, iar pentru afișarea oricărui element din listă selectam elementul dorit printr-un click stânga iar pentru afișare se apasă click dreapta. Aceasta este o implementare făcută prin ajutorul addMouseListener care transmite o notificare sistemului când utilizatorul utilizează mouse-ul, care interacționează cu o component selectată. Această instanță are în interior o funcție mousePressed cu un argument, obiectul e de tip MouseEvent , și printr-o verificare SwingUtilities.isRightMouseButton(e) sistemul realizează ca notificarea este creată prin atingerea mouse-lui de dreapta. Butonul SUM face suma dintre două polinoame și o printează. Acesta deschide o nouă fereastră în care se vede clar ca sunt două liste amplasate, cea din stânga reprezintă elementele curente ale ArrayList-ului (poate fii nulă) iar cea din dreapta este o listă goală în care se poate adăuga prin intermediul butonului ADD, iar cu ajutorului butonul Reset All se șterg toate componentele curente din lista din partea dreaptă. Astfel că butonul SUM poate fii acesat pentru afișare doar în cazul în care lista din partea dreaptă conține strict doua elemente, deoarece funcția care se aplelează conține exact doua argumente, după care se printează un mesaj cu textul colorat reprezentând rezultatul nou, acesta din urmă se printează și în interiorul consolei.

Pe aceleași principii funcționează și butoanele de DIFF care face diferența dintre două polinoame respective din primul scade pe al doilea, și Multiply care înmulțeste cele doua polinoame transmise în lista din partea dreaptă. Ca și butonul de SUM din interioul interfeței principale.Butoanele DivideByNumber și GeneratePolynom funcționează pe două principii asemănătoare deoarece amândoua deschid cate o fereastră separată . Pe respectiva fereastră deschisă se află un text care ne sugerează să introducem un polinom, iar apoi un numar transmis ca și rădăcină întreagă pentru a determina rezultatul într-un mesaj distinct, și în interiorul consolei . DivideByNumber împarte fiecare coeficient existent în polinom la rădăcina introdusă, iar GeneratePolynom generează un polinom conform algoritmului menționat mai sus.

**Anexe:**

Arhiva .RAR ce conține proiectul este încărcată pe platforma <https://dev.elearning.e-uvt.ro>.