**Mihai Matyas**

**Gr 231**

**Adunarea si inmultirea polinoamelor**

**Cerinta problemei:**

Implementarea unui sistem ce permite aplicarea de operatii pe polinoame, anume adunarea si inmultirea acestora. Scopul primordial al problemei este implementarea unui sistem care sa asigure aceste operatii intr-o maniera cat mai eficienta si cat mai rapida, astfel ca se va apela la folosirea programarii concurent-paralele pentru a asigura aceasta eficienta. De asemenea se va realiza o comparatie intre eficienta problemei in implementarea secventiala fata de cea implementata prin programare concurent-paralele.

**Datele de intrare:**

Datele de intrare constau in polinoamele pe care se vor aplica operatii. Aceste polinoame vor fi considerate ca fiind de forma:

Ax^n+Bx^n-1 +... +Zx^0

A,B,C…Z reprezinta coeficientii polinomului. Acesti coeficienti se reprezinta sub forma de numere naturale.

n, n-1, n-2…,0 reprezinta exponentii polinomului. Acestia dau gradul polinomului si sunt reprezentati de numere naturale.

Datele se citesc dintr-un fisier .txt si au urmatorul **format:**

>Polinom 1

>Polinom 2

Un polinom in fisier are formatul:

A B C D E F

Unde {A,...F} reprezinta coeficientii polinomului, separati de cate un spatiu, gradul polinomului fiind determinat de lungimea listei coeficientilor fiecarui polinom.

Un polinom este reprezentat de clasa **Polinom** care are ca si parametrii gradul polinomului si o lista care reprezinta o insirare a coeficientilor acestuia.

**Prelucrarea datelor**

Dupa introducerea datelor in sistem, se va putea selecta operatia care se doreste a fi efectuata.

Pentru **implementarea secventiala**, operatiile se realizeaza in urmatorul mod:

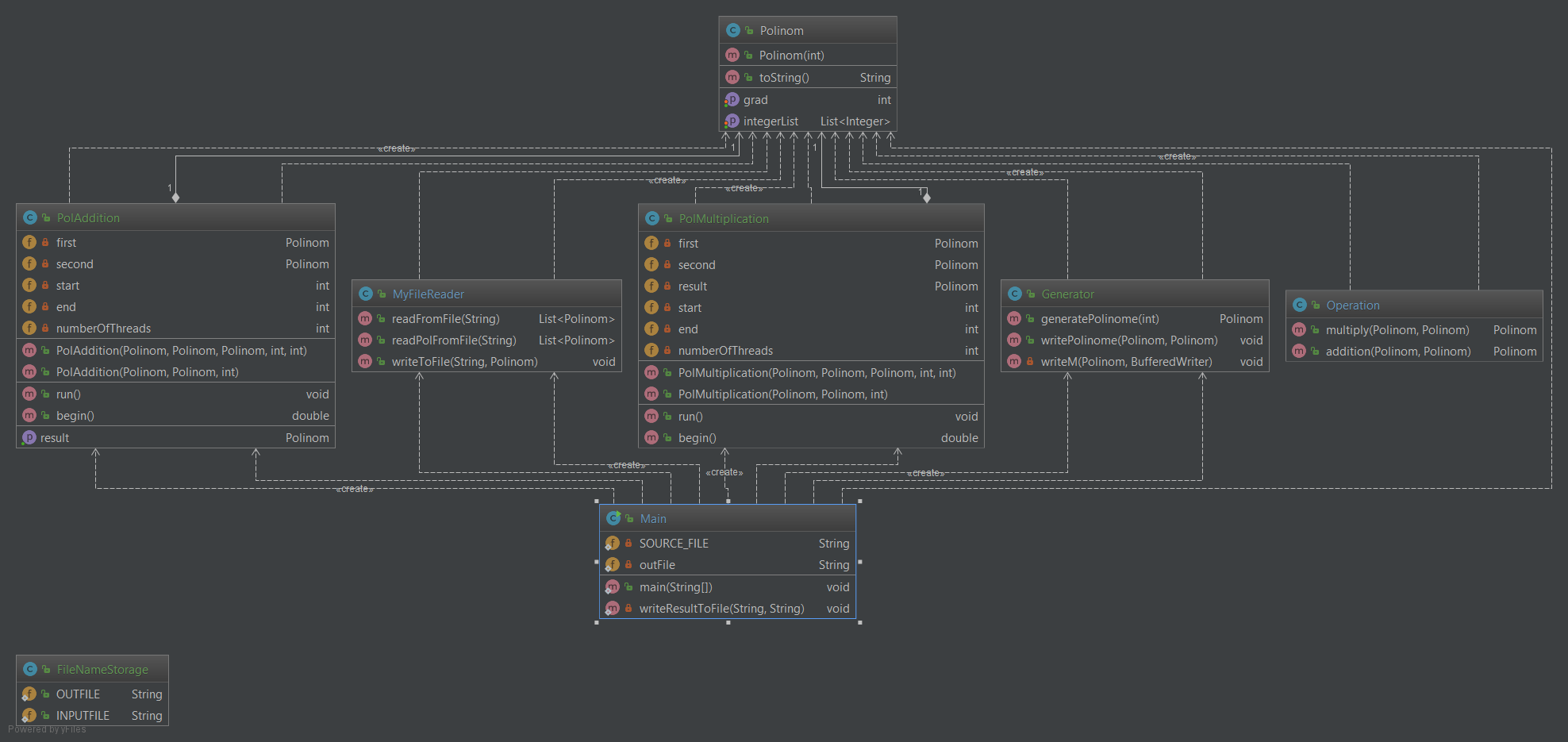
**Adunarea** se realizeaza prin adunarea coeficientilor de acelasi rang, polinomul rezultat fiind un polinom de acelasi grad cu gradul maxim dintre cele doua polinoame care se aduna.

**Multiplicarea** se realizeaza prin inmultirea fiecarui coeficient cu toti ceilalti coeficienti din celalalt polinom, polinomul rezultat fiind un polinom de rang egal cu produsul dintre gradele polinoamelor care se inmultesc.

**Complexitatea** algoritmului secvential este data de gradele polinoamelor utilizate ca date de intrare. Astfel, complexitatea se poate defini ca fiind O(n\*m), unde n si m reprezinta gradele polinoamelor.

Pentru implementarea paralela, numarul de operatii pe care le executa fiecare fir de executie este determinat de marimea polinoamelor prelucrate ca si date de intrare si numarul total de fire de executie. Task-urile pe care le executa fiecare fir de executie este dat de raportul dintre numarul total de task-uri si numarul de fire de executie existente. Numarul total de task-uri este dat de operatia efectuata si marimea datelor de intrare.

**Diagrama de clase a proiectului**



**Modelul PRAM**

PRAM modeleaza calculul pentru sistemele cu memorie comună, dar s-au facut

extinderi și pentru modelarea arhitecturilor care se bazează pe transmiterea de mesaje.

Produsul și polinoamele pe care s-au aplicat operatii, first și second pot sa fie stocate într-o memorie partajată și după obtinerea rezultatului, polinomul rezultat poate sa fie găsit în aceeași zonă de memorie partajată.

Pentru inmultire:

**for ( i = 1 to n ) do in parallel**

**for ( j = 1 to m ) do in parallel**

**result(i,j) <- result(i,j) + first(i,j) \* second(i,j)**

**end for**

**end for**

**Proiectare si testare**

Pentru testarea implementarii paralele, atat pentru adunarea cat si pentru inmultirea de polinoame, s-au parcurs urmatoarele etape:

* S-au generat polinoame cu date aleatoare de dimensiune aleasa
* Polinoamele generate au fost introduse intr-un fisier, care ulterior va servii ca fisier de date de intrare
* Polinoamele generate sunt citite si prelucrate, fie prin adunare, fie prin inmultire polinomiala
* Fiecare fir de executie va executa defapt o portiune secventiala reprezentata de blocul secvential de adunare sau inmultire a unei subliste de coeficienti
* Durata operatiilor este inregistrata si introduse intr-un fisier cu date de test, care va contine:
  + Gradul polinoamelor utilizate
  + Numarul de fire de executie utilizat
  + Durata procesului de executie

Aceste date au fost generate pentru grade diferite de polinoame si numar diferit de fire de executie, astfel incat sa se poata observa variatia de timp odata cu marimea polinoamelor si cu numarul de fire de executie.

**Analiza Comparativa**

Se observa din datele de test generate de program ca timpul de executie in varianta paralela este mai scazut fata de timpul de executie pentru varianta secventiala. Empiric, se observa o scadere a timpului de 50% intre rularea programului in varianta secventiala si rularea programului in varianta paralela, cu 2 fire de executie.

Odata cu cresterea numarul de fire de executie, timpul de executie scade pana la atingerea unui punct de echilibru intre timpul de executie si numarul de fire de executie. Totusi, daca se folosesc prea multe fire de executie, timpul va creste - deoarece apar variatii de timp datorate timpului de creare a firelor de executie suplimentare si timp de asteptare - cauzat de limitarile procesorului pe care este rulat programul.