Laborator 4

Adunare n polinoame. Sincronizare la nivel de lista

* Cerinta

Se considera n polinoame reprezentate prin lista de monoame (reprezentare: lista inlantuita ordonata dupa exponentii monoamele). Se cere adunarea polinoamelor folosind o implementare multithreading (p threaduri). Polinoamele se citesc din fisiere – cate un fisier pentru fiecare polinom: - un fisier contine informatii de tip (coeficient, exponent) pentru fiecare monom al unui polinom.

* Preconditie, Postconditie

Fisierele de intrare si cel de iesire nu contin monoame cu coeficient egal cu 0.

* Structura

Exista o entitate principal numita Polinom pentru care se retine o lista inlantuita de obiecte de tip Element (retine un numar intreg pentru coeficient si un numar intreg pentru monom).

Pentru fiecare tip de executie (secventiala sau paralela) exista o clasa care contine un obiect de tip Polinom ce va fi rezultatul adunarii si un numar intreg (n) ce reprezinta numarul de polinoame. In clasa Paralel avem in plus un numar intreg (p) ce reprezinta numarul de threaduri si o lista de obiecte de tip Polinom .

* Detalii implementare

La executia secventiala se citeste fiecare polinom din fisier si se creeaza un obiect de tip Polinom. Pentru fiecare element al polinomului se verifica daca coeficientul current exista deja in polinomul rezultatul sau nu. Daca exista, atunci se adauga valoarea monomului la ce exista deja in rezultat, altfel se adauga o noua intrare in lista inlantuita. Inainte de a ascris in fisier se sorteaza polinomul in functie de valorile coeficientilor.

La executia paralela exista un thread principal care citeste toate polinoamele din fisiere si le adauga in lista retinuta care are rolul de coada. Astfel, in momentul in care un thread vrea sa faca calculele de adunare prima data verifica daca mai sunt elemente in coada; daca nu mai sunt atunci i se opreste executia. Altfel, se extrage primul polinom din coada si se face adunarea la polinomul rezultat, exact la fel ca in cazul executiei secvemtiale.

Atat pentru extragerea din coada, cat si pentru realizarea calculelor se va folosi synchronized fie pe coada, fie pe polinomul rezultat.

**Java**

|  |  |  |  |
| --- | --- | --- | --- |
| **Executie** | **P** | **Timp** | **n** |
| secventiala | **-** | 53.72294 | 10 |
|  |  |  |  |
| paralela | 4 | 20.20392 | 10 |
| paralela | 6 | 23.92143 | 10 |
| paralela | 8 | 23.99035 | 10 |

Se poate observa ca executia paralela cu 4 threaduri a fost mai rapida decat cea secventiala, dar si cea mai rapida dintre toate cele paralele.