Pertea Alexandra-Ioana
Grupa 235/2

Laborator 4 PPD

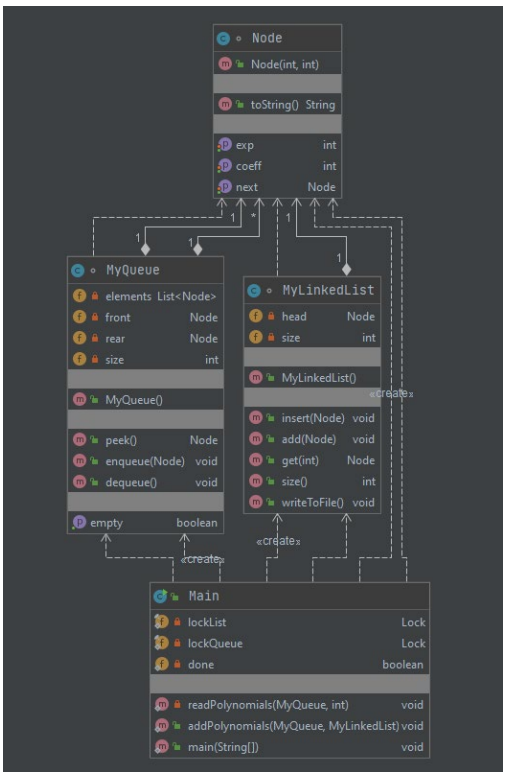
Deadline: săptămâna 9

1. Cerința problemei

Se consideră n polinoame reprezentate printr-o listă de monoame. Se cere adunarea polinoamelor folosind o implementare multithreading (p threaduri).

Polinoamele se citesc din fișiere – câte un fișier pentru fiecare polinom – astfel: un fișier conține informații de tip (coeficient, exponent), pentru fiecare monom al unui polinom.

2. Proiectarea aplicației



Primul thread citește câte un monom din fișier și îl adaugă în coada de tip MyQueue.

Celelalte thread-uri preiau câte un monom din queue și îl adună la polinomul reprezentat prin lista MyLinkedList, astfel:

- Dacă există un monom cu același exponent, se face adunarea între monoame

- Altfel, se inserează monomul în listă

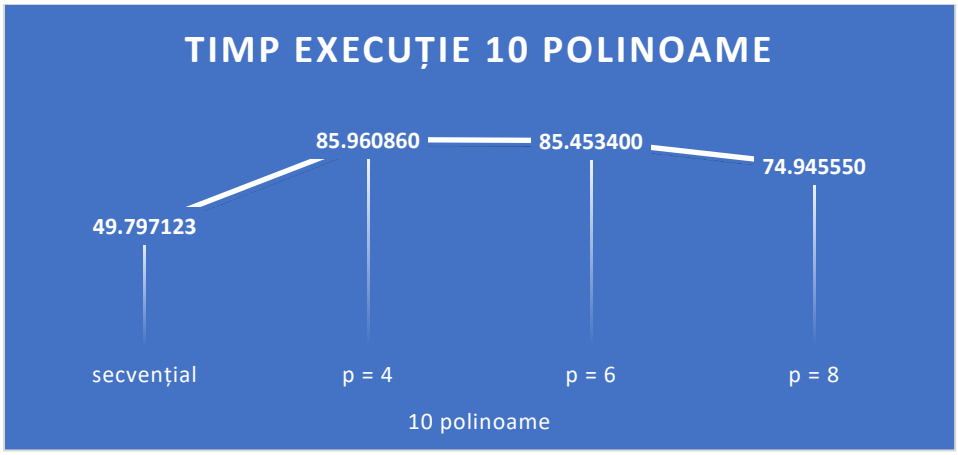
Sincronizarea se face **la nivel de listă**, folosind lock-uri de tip ReentrantLock, pentru **listă** și pentru **coadă**

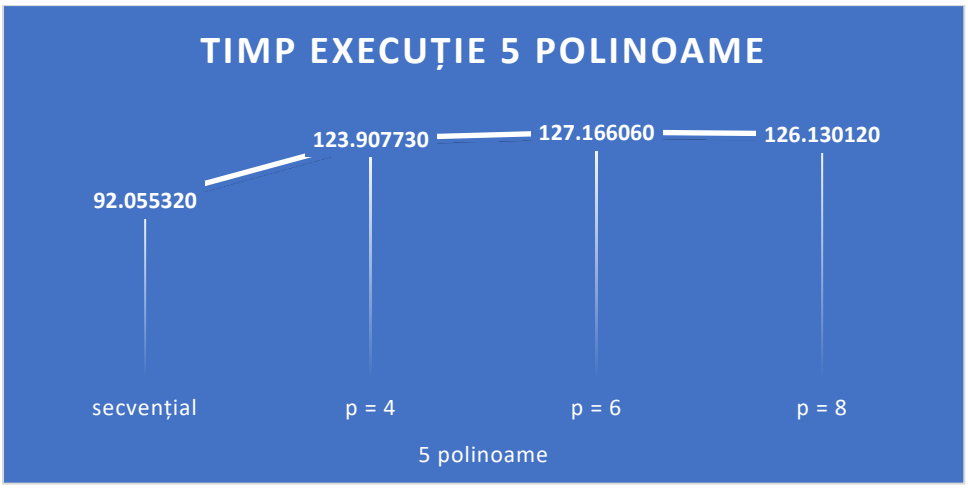
La final, rezultatul este scris în fișierul *output.txt*.

3. Rezultatele testelor

Numărul de polinoame	Numărul de procese	Timp execuție (ms)
10 polinoame	secvențial	49,797123
	p = 4	85,96086
	p = 6	85,4534
	p = 8	74,94555

5 polinoame	secvențial	92,05532
	p = 4	123,90773
	p = 6	127,16606
	p = 8	126,13012





4. Analiza rezultatelor

Pentru adunarea a 10 polinoame:

Timpul de execuție scade odată cu creșterea numărului de thread-uri. Varianta secvențială este cea mai rapidă.

Pentru adunarea a 5 polinoame:

Timpul de execuție are un peak la 6 thread-uri, dar scade odată cu creșterea numărului de thread-uri la 8. Varianta secvențială este cea mai rapidă.