**Aspectul de Funcționare al Codului**

Acest program implementează un model **Producător-Consumator** folosind tehnici de programare paralelă pentru a procesa date din mai multe fișiere și a le organiza într-o listă simplu înlanțuită sortată. Obiectivul principal este să agregăm punctajele participanților și să eliminăm participanții aflati în lista neagră (blacklist).

**Structura Principală**

1. **Producători**: Citirea fișierelor de intrare și adăugarea datelor procesate într-o coadă sincronizată (MyQueue).
2. **Consumatori**: Preluarea datelor din coadă și actualizarea unei liste (LinkedList) pentru a adăuga/actualiza scorurile participanților sau pentru a-i elimina dacă se afla în blacklist.
3. **Listă Neagră**: Orice participant cu un scor de -1 este adăugat într-un set de tip blacklist.

**Componente Principale**

**1. MyQueue (Coada sincronizată)**

* Este utilizată pentru a stoca temporar datele procesate de producători.
* Sincronizarea este asigurată cu Lock și Condition:
  + Consumatorii așteaptă când coada este goală.
  + Producătorii așteaptă când coada este plină.

**2. LinkedList (Listă simplu înlanțuită)**

* Structura principală care stochează participanții.
* Funcții importante:
  + **addOrUpdate**: Adaugă un nou participant/actualizează scorul unui participant existent.
  + **removeParticipant**: Elimină participanții din listă dacă sunt în lista neagră.
  + **sort**: Sortează lista finală după punctaj (descrescător) și ID (crescător) doar dupa procesarea tuturor participanților.

**3. MyWorker (Consumatorul)**

* Consumatorii preiau date din coadă și folosesc funcțiile din LinkedList pentru a actualiza lista.

**4. Fișierele de intrare**

* Fiecare fișier conține linii de formă: ID,Score.
* Producătorii procesează aceste fișiere și adaugă în coadă scorurile și țara asociată participantului.

**Fluxul de Funcționare**

1. **Inițializare**
   * Se configurează numărul de producători, consumatori, coada și lista neagră.
2. **Producători**
   * Fiecare producător procesează un fișier și adaugă datele în coadă:
     + ID-ul participantului.
     + Scorul asociat.
     + Țara asociată fișierului.
   * Dacă un scor este -1, participantul este adăugat în lista neagră.
3. **Consumatori**
   * Consumatorii preiau date din coadă și actualizează lista:
     + Participanții în lista neagră sunt eliminați.
     + Participanții noi sunt adăugați cu scorurile și țara asociată.
4. **Finalizare**
   * După terminarea producătorilor, consumatorii primesc un semnal de stop ([-1, 0, 0]).
   * Lista legată este sortată și rezultatele sunt scrise într-un fișier.

**Sincronizare și Siguranță**

* **Blocare**: Operațiile critice asupra cozii și listei sunt protejate cu Lock pentru a evita condițiile de cursă.
* **Lista neagră**: Este sincronizată folosind synchronized pentru acces concurent sigur.

**Optimizări Cheie**

1. **Lista neagră**: Eliminarea participanților imediat ce sunt identificați.
2. **Sortare**: Lista este sortată doar la final, pentru eficiență.
3. **ExecutorService**: Gestionarea eficientă a firelor de execuție pentru producători și consumatori.

**Erori Gestionate**

* **Fișiere lipsă**: Verificarea existenței fișierelor.
* **Intrare invalidă**: Liniile invalide din fișiere sunt ignorate.
* **Interuperea firelor**: InterruptedException este gestionată în mod corespunzător.

**Testare:**

|  |  |  |
| --- | --- | --- |
|  | | Duration |
| p\_r = 2 | p\_w = 2 | 114 ms |
| p\_w = 4 | 125 ms |
| P\_w = 12 | 106 ms |
| p\_r = 4 | p\_w = 2 | 73 ms |
| p\_w = 4 | 136 ms |
| P\_w = 12 | 119 ms |
| Secvential | | 43ms |