**Laborator 5**

Problema de la Laboratorul 4 cu modificarile:

Clasamentul final trebuie sa contina informatiile:

**ID, punctaj, tara**

Pentru fiecare participant.

1. Fine grain synchronization in loc de coarse grain synchronization

Adica

Sincronizare la nivel de nod in loc de sincronizare la nivel de lista

- pentru o operatiile pe lista (insert, delete, get, set )

Pentru aceasta este necesar sa faceti o implementare a unei liste inlantuite cu noduri care au pe langa informatie si legatura spre urmatorul nod si un ‘lock’(mutex).

Implementarea se simplifica daca folositi lista inlantuita cu santinele (nod start si nod final fara informatie )

ATENTIE si la situatia in care pot fi doi ‘workers’ care lucreaza pe lista pentru 2 perechi cu acelasi ID!

Si cu impunerile :

1. coada in care se adauga perechile citite are o capacitate maxima (MAX =50, MAX=100)
2. a se folosi variabile conditionale pentru implementarea “producator-consumator”!!!
3. Pentru varianta de implementare in Java se va folosi un executor (thread pool cu maxim p\_r threaduri) pentru a rezolva taskurile de citire

un task de citire este format din citirea tuturor inregistrarilor dintr-un fisier

=> 1 task de citire pentru fiecare fisier

si simplificarea:

1. Lista nu trebuie sa fie tot timpul ordonata – poate fi ordonata doar la final

**Testare**

5 tari cu cate maxim 100 de concurenti si minim 80 de concurenti in fiecare tara si 10 probleme – nu este acelasi numar de concurenti in fiecare tara!

1. p\_r= 4 p\_w= 2, 4, 12
2. p\_r= 2 p\_w= 2, 4, 12