

Laborator 3 (PPD)

Problema

Implementati o clasa "SortedLinkedList" – lista inlantuita sortata cu elemente de tip real, astfel incat sa fie "thread-safe".

Operatiile furnizate de clasa sunt:

- insert(a: Real)
 - o pre: this=[a(0),...,a(n)] a.i. pentru orice i: $0 < i < n \Rightarrow a(i-1) \leq a(i) \leq a(i+1)$
 - o post: a este in lista pe pozitia corespunzatoare:
this=[a(0), ...,a(i), a, a(i+1)...,a(n)] a.i. $a(i) \leq a \leq a(i+1)$
- delete(a: Real)
 - o pre : exista pos o pozitie valida in lista a.i. $lista(pos) = a$
 - o post: elementul 'a' nu mai este in lista
- getIterator():Iterator
 - o pre : lista valida
 - o post: result = it , it in Iterator pe lista

Clasa Iterator va fi definita corespunzator cu specificatia clasica a unui iterator (implementeaza interfata IIterator cu operatiile: getNext(); isValid(); getElement()).

Variante

Se cere implementarea a doua variante:

1. sincronizare la nivel de nod sau portiune din lista,
2. sincronizare la nivelul intregii liste.

La testarea iteratorului se va folosi intotdeauna sincronizarea intregii liste.

Testare:

Test 1

- Program care porneste 4 threaduri care lucreaza pe aceeasi lista:
 - o T1 adauga 10 de valori
 - o T2 adauga 5 de valori
 - o T3 sterge 7 de valori
 - o T4 itereaza lista in mod repetat la interval Δt (ex. de 3s – se alege a.i. sa fie egal cu timpul necesar adaugarii a 3 valori) si se termina dupa ce threadurile T1, T2, T3 se termina.
- Programul actualizeaza un fisier <List.log> in care se adauga fiecare o evidenta a fiecarei operatii executate (tip operatie +parametrii) impreuna cu numarul threadului care a executat-o precum si timpul la care a inceput executia acelei operatii; o operatie de iterare se considera o operatie atomica si se vor salva toate elementele iterate).

Se cere evaluarea corectitudinii executiei pentru cele 2 variante prin verificarea operatiilor executate – analiza fisier <List.log>

Test 2

- Program care porneste 4 threaduri care lucreaza pe aceeasi lista:
 - o T1 adauga 100 de valori
 - o T2 adauga 50 de valori

- T3 sterge 50 de valori
- T4 itereaza lista in mod repetat la interval de Δt (ex. 5s – se alege a.i. sa fie egal cu timpul necesar adaugarii a 10 valori) si se termina dupa ce threadurile T1, T2, T3 se termina (elementele iterate se salveaza intr-un fisier <Lista.res> fiind precedate de numarul iteratiei).
- Se cere evaluarea timpului de executie pentru cele doua variante (fiecare evaluare presupune repetarea executiei de cel putin 10 ori).

Implementare: la alegere implementare Java sau C++11(sau versiune mai noua).

Scop: Exemplificare sincronizare:

comparatie *fine grain synchronization* (la nivel de nod/portiune din lista) versus *course grain synchronization* (la nivel de lista).

Deadline:

Saptamana 7