Programare paralela si distribuita

Laboratorul 3

**Analiza cerintei:**

Enuntul problemei:

* Implementati o clasa “SortedLinkedList” –lista inlantuita sortata cu elemente de tip real, astfel incat sa fie “thread-safe”. Pe aceasta lista se vor implementa operatiile de adaugare, stergere si parcurgere. Clasa Iterator va fi definite corespunzator cu specificatia clasica aunui iterator (implementeaza interfata IIterator cu operatiile: getNext(); isValid(); getElement()).
* Pe aceasta clasa se vor implementa doua variante:

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

**Rezolvarea problemei:**

Pentru sincronizarea la nivel de lista se va folosi clasa SortedLinkedListSync in care hederele metodelor de insert, delete si getHead(Iterator) vor fi precedate de synchronized.

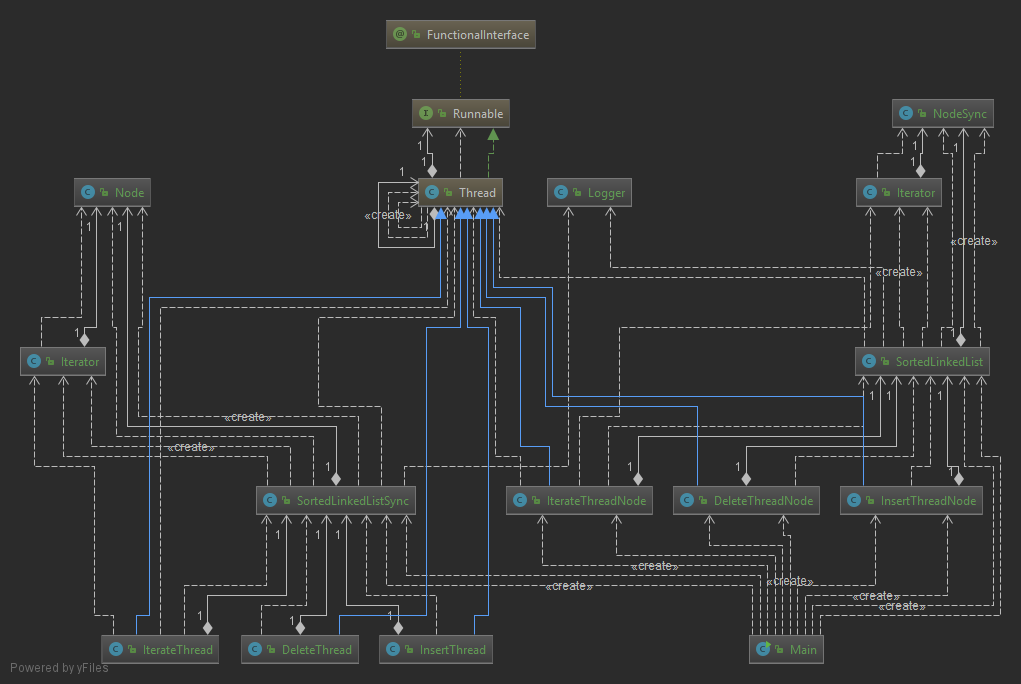
Pentru sincronizarea la nivel de nod sau portiuni din lista se va folosi clasa

SortedLinkedList in care in interiorul metodelor se va folosi synchronized(nod)

{bloc de cod care opreaza asupra nodului}. Astfel in inteiorul metodei insert se va bloca nodul anterior si nodul urmator fata de pozitia unde va fi introdus elementul si se va creea legatura dintre ele, in interiorul metodei delete se va bloca nodul anterior si urmatorul iar la getHead se va bloca nodul current a carei valoare este accesata.

**Graficele ce indica comportamentul operatiilor:**

**Structura programului - Java**:

****