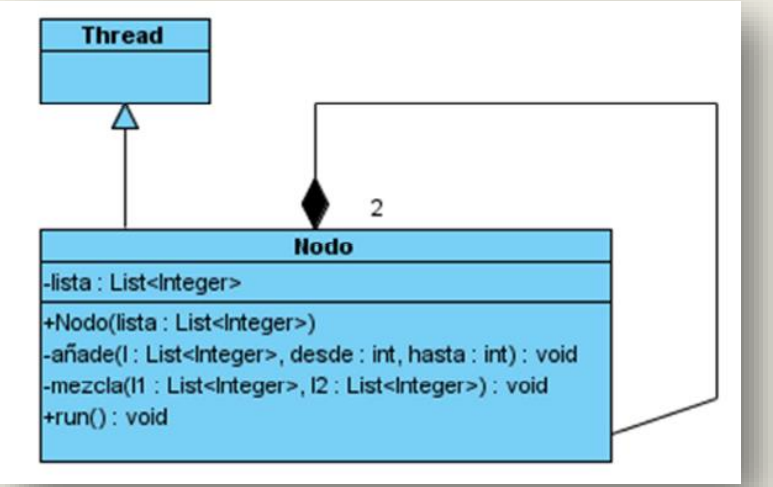
Programación de aplicaciones multiproceso

Ejercicio 5

Diseña un programa java que implemente el algoritmo recursivo de ordenación por mezcla de forma concurrente. El programa debe construir un árbol binario de hebras, de profundidad variable, dependiente del número de elementos de la lista a ordenar. El sistema sólo necesita una clase Nodo como la descrita en el diagrama siguiente:



Programación de aplicaciones multiproceso

Ejercicio 5

Inicialmente, el método **main()** crea un **primer nodo** (el nodo raíz del árbol) al que se le pasa una **lista de números aleatoria**. Cada uno de los nodos que se vayan creando en el sistema se comporta de la misma forma.

- Si la lista que se le pasa al nodo en el constructor tiene 0 ó 1 elemento, no hay nada que hacer (la lista ya está ordenada).
- En otro caso, la hebra crea dinámicamente dos nuevas hebras, y le pasa a cada una de ellas una de las dos mitades de su lista. Una vez que cada hebra hija ha ordenado sus mitades, la hebra padre las mezcla de forma ordenada para producir el resultado esperado.

Programación de aplicaciones multiproceso

Ejercicio 5

Como **ayuda** para la implementación del método run(), se sugiere implementar los métodos privados **añade(l,d,h)** que añade a la lista l todos los elementos de lista desde la posición d (**incluida**) hasta la posición h (**excluida**), y **mezcla(l1,l2)** que mezcla de **forma ordenada** los elementos de las listas l1 y l2 dejando el resultado en lista.

Nota: utiliza la interfaz List<Integer>, y la clase ArrayList<Integer> para representar las listas manejadas por tu programa. Para facilitar la implementación puedes usar los métodos add, addAll, subList de estas clases.