

Seguimiento #4

Gabriel Alejandro Delgado Alvarez

Juan Felipe Sinisterra

TAD de la lista enlazada ordenada.

TAD OrderedLinkedList

LinkedList = { First = {first}. Size = {size} }

{ inv: size \geq 0 }

Operaciones primitivas:

- OrderedLinkedList
- add Node
- remove Node
- getSize
- nodesToString Node
- toString

LinkedList()

“Constructora”

“Crea una nueva lista enlazada con los datos vacíos”

{ pre: TRUE }

{ post: list = { First: null, size = 0 } }

add(element)

“Modificadora”

“Añade un nuevo elemento a la lista enlazada, poniéndolo en la posición necesaria para que la lista siempre esté ordenada.”

{ pre: list = { First: null, ... } \wedge element \neq null }

{ post: list = { First: Node<element>, ... size = size + 1 } \wedge element \neq null }

remove(element)

“Modificadora”

“Elimina el nodo con el elemento buscado”

{ pre: list = {First: Node<element>, ... size = 1 } \wedge element \neq null }

{ post: list = {First: null, ... , size = size - 1 } \wedge element \neq null }

size()

“Analizadora”

“Devuelve el tamaño de la lista enlazada”

{ pre: list = {First: Node<>, size \geq 0} }

{ pos: < size > }

nodesToString()

“Analizadora”

“Devuelve el String con la lista enlazada y sus elementos en forma de array”

```
{ pre: list = {First: Node<>, size >= 0} }  
{ pos: {"Node1, Node2, Node3 ... NodeN"}}
```

toString()

“Analizadora”

“Hace el llamado al método nodesToString y devuelve un String de los nodos de la lista enlazada cubiertos con corchetes '[...]’.”

```
{ pre: list = {First: Node<>, size >= 0} }  
{ pos: {"[ Node1, Node2, Node3 ... NodeN ]"}}
```