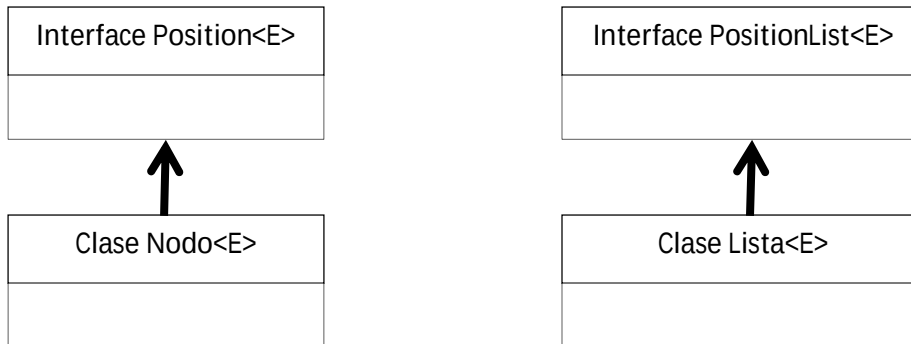


Estructuras de Datos – 1er Parcial (2011)

Ejercicio 1

a) Dado el siguiente diagrama de clases:



En JAVA, complete la clase Nodo y Lista (u otra que considere necesaria) para implementar una lista doblemente enlazada (no circular y con centinelas o celda inicial y final vacía). Los elementos de la lista son de tipo genérico E.

- Para la clase Nodo, defina los atributos, escriba las firmas pero no implemente el cuerpo de las consultas y comandos triviales (es decir, aquellos de simple acceso a atributos). En cambio implemente el o los constructores. La interfaz Position corresponde a la dada en [GT].
- Para la clase Lista, defina los atributos e implemente solamente los constructores. No defina firma ni implemente el cuerpo de consultas ni comandos a menos que los vaya a utilizar para resolver el inciso b). La interfaz PositionList corresponde a la dada en [GT] y no es iterable.

b) Implemente en JAVA una operación para la clase Lista definida en el inciso a) que recibe una lista B del mismo tipo y que modifique la lista A receptora del mensaje de acuerdo a:

Eliminar todos los elementos x de A que estén presentes en B y que originalmente ocupan posiciones pares en A. Considerar que el primer elemento de la lista ocupa la posición 1. Recuerde que está agregando un método a la clase lista, por lo tanto tiene total acceso a su estructura; si utiliza otras operaciones del TDA debe implementarlas.

Esta operación no debe generar una nueva lista, sino que debe modificar la lista A receptora del mensaje

- c) Justificando adecuadamente, indique el orden del tiempo de ejecución del procedimiento solución al inciso b) asumiendo que las operaciones del tipo E son de tiempo constante.
- d) Extienda (solamente en UML) el diagrama de clases del inciso a) para una lista iterable. Defina las firmas de los métodos en las nuevas clases pero no implemente el cuerpo de las consultas, comandos ni constructores.

Ejercicio 2

- a) Dado el esquema de definición de clases:

<pre>Public class TNode<E> Implements Position<E> { } </pre>	<pre>Public class Arbol<E> Implements Tree<E> { } </pre>
--	--

Disponiendo que posee la implementación del TDALista presentado en [GT] y las interfaces Position y Tree de [GT], complete las clases TNode y Arbol (u otra que considere necesaria) para implementar un árbol general, cuyos rótulos son de tipo genérico E, utilizando la representación de colección de hijos y referencia al padre

- Para la clase TNode, defina los atributos, escriba las firmas pero no implemente el cuerpo de las consultas y comandos triviales (es decir, aquellos de simple acceso a atributos) pero si implemente los constructores.
 - Para la clase Arbol, defina los atributos e implemente solamente los constructores. No defina firma de consultas ni comandos ni implemente su cuerpo a menos que los vaya a utilizar para resolver el inciso b).
- b) Agregue una operación a la clase Arbol definida en el inciso a) que por cada nodo con rótulo R del árbol receptor original del mensaje, debe eliminar su hijo extremo derecho. Si el nodo a eliminar es un nodo interno, entonces sus hijos deben subir al nivel del padre eliminado ocupando su lugar y manteniendo su orden relativo entre hermanos. Esta operación no debe generar un nuevo árbol sino, en cambio, modificar el árbol que recibe el mensaje. Para este ejercicio no tiene acceso a la estructura de la clase lista, pero si puede utilizarla como un TDA. Recuerde que está agregando un método a la clase Arbol, por lo tanto tiene total acceso a su estructura. Si utiliza otras operaciones del TDAArbol debe implementarlas (pero no es necesario implementar las operaciones usadas de TDALista).

- c) Justificando adecuadamente, indique el orden del tiempo de ejecución del procedimiento solución al inciso b) asumiendo que las operaciones del tipo E son de tiempo constante, e indicando apropiadamente el orden del tiempo de ejecución de las operaciones utilizadas del TDALista. Para ello justificar indicando desde su punto de vista cual es la estructura de datos subyacente al TDALista.

De considerarlo necesario, puede asumir que la lista es iterable.

Ejercicio 3

- a) Defina en JAVA una interfaz para el TDACola donde los elementos son de tipo genérico $\langle E \rangle$.
- b) Defina en JAVA todas las estructuras de datos (clases y atributos) necesarias para implementar colas de acuerdo a la interfaz definida en el inciso a) utilizando un arreglo circular de tal manera que las operaciones de encolar y desencolar tengan tiempo de ejecución constante.
- c) Utilizando el lenguaje JAVA, implemente el constructor de la cola y la operación encolar para la estructura de datos propuesta en el inciso b) y de acuerdo a la interfaz definida en el inciso a) de tal manera que la misma tenga tiempo de ejecución constante.
- d) Suponiendo que posee la implementación de los TDA Pila (como se define en [GT]) y de la cola definida en el inciso a), implemente solamente en términos de las operaciones de los TDA Pila y Cola, un método que recibe una cola de pilas de enteros Q y un entero X e invierte la cola y elimina de cada pila todo elemento igual a x. Esta operación debe modificar la cola Q pasada por parámetro.