



Estructuras de Datos
Grado en Informática
Segundo Curso, segundo cuatrimestre
Escuela Politécnica Superior de Córdoba
Universidad de Córdoba
Curso académico 2014-2015



Práctica 2. Implementación y uso de una lista ordenada doblemente enlazada

Objetivos

El estudiante debe ser capaz de

- implementar el tipo abstracto lista ordenada ascendente doblemente enlazada a partir de las especificaciones de la lista ordenada ascendente
- y codificar un programa que utilice dicho tipos abstracto para resolver un determinado problema.

Enunciado

Implementa la clase ***ListaOrdenadaDoblementeEnlazada*** a partir de las siguientes especificaciones para una lista ordenada:

- ***TAD ListaOrdenada:***
Observadores:

Bool esUltimo(Integer pos) // ¿Es pos la última posición?

Bool esValida(Integer pos) // ¿Es pos una posición válida?

item(Integer pos) // Devuelve el ítem de la listaOrdenada que está almacenado en posición

Precondición: esValida(pos) = true.

Integer buscaItem (T item, Integer from) // devuelve la posición del ítem or -1, a partir de from

Precondición: esValida(from) = true

*Postcondición: retVal != -1 implica esValida(retVal) y
item(retVal)=item*

Modificadores:

inserta(T item) // Inserción ordenada de ítem.

*postcondicion: pos=buscaPosicion(it) y esValida(pos-1)= true
implica item(pos-1)<=it.*

*postcondicion: pos=buscaPosicion(it) y esValida(pos+1) = true
implica it<=item(pos+1).*

borraPosicion(Integer pos) //Borra el ítem que está en la posición pos

Precondición: estaVacía() = false

Precondición: esValida(pos) = true.

Postcondición: old.esValida(pos+1) = true

implica new.item(pos)=old.item(pos+1)

- **Programa de uso de las clases**

- Implementa un programa que haga uso de una clase denominada **Asignatura**, que contendrá una serie de alumnos, implementada usando el tipo **ListaOrdenadaDoblementeEnlazada**. Dicho programa tendrá las siguientes opciones:
 - Crear Asignatura. En esta opción se implementará usando los siguientes pasos:
 - Se pedirá al usuario el número de alumnos **n** que tiene la asignatura.
 - Con el valor de **n** (menor o igual que 100) y con el nombre de un fichero de texto ("**personas.txt**") que contiene los datos de 100 personas, se invocará a la función *generarPersonas()*, que devuelve un vector de la STL que contendrá **n** personas seleccionadas aleatoriamente del fichero de texto.
 - La clase persona, las funciones necesarias para manejarla y el fichero de texto con los datos de 100 personas son suministrados.
 - Se recorrerá el vector creado en el paso anterior y a la vez se irán introduciendo las personas como alumnos de la asignatura.
 - Mostrar un listado ordenado ascendentemente de los alumnos de la asignatura.
 - Mostrar un listado ordenado descendentemente de los alumnos de la asignatura.
 - Introducidos los datos completos de dos alumnos, mostrar ordenados ascendentemente los alumnos comprendidos entre ambos.
 - Introducidos los datos completos de dos alumnos, mostrar ordenados descendentemente los alumnos comprendidos entre ambos.
 - Modificar los datos de un alumno. Para ello se seguirán los siguientes pasos:
 - Se introducirán los datos del alumno a modificar.
 - Se introducirán los nuevos datos.
 - Se borrará el alumno con los datos antiguos.
 - Se insertará el alumno con los datos nuevos.

Desarrollo de la práctica

- Tiempo de realización: 3 semanas.
- Comienzo de la práctica: semana del 2 al 6 de marzo.
- Entrega de la práctica: semana del 23 al 27 de marzo.