

Curso 2003/2004 Estructura de Datos y de la Información I. Informática, I. T. Informática de Gestión y de Sistemas PRÁCTICA 1

1 Objetivo

El objetivo es practicar la independencia de la implementación en los tipos abstractos de datos (TADs). Para ello se realizarán dos implementaciones distintas del TAD tLista (una estática con arrays y una dinámica con punteros) y se planteará un problema que deberá ser resuelto con las dos implementaciones (es decir, un mismo programa principal alternará la utilización de una implementación u otra modificando su cláusula `uses`).

2 TAD tLista

2.1 Tipos de datos incluidos en el TAD

- tLista. Representa a la lista
- tInfo. Dato de un elemento de la lista
- tPos. Posición de un elemento de la lista.
- NULO. Constante utilizada para representar posiciones nulas.

2.2 Operaciones incluidas en el TAD

Una precondition común para todas las operaciones (salvo listaVacía) es que la lista debe estar previamente creada.

- listaVacía (tLista) → tLista
 - Crea una lista vacía
 - PreCD: la lista no debe de estar previamente creada.
- esListaVacía (tLista) → boolean
 - Determina si la lista está vacía
- primero (tLista) → tPos
 - Devuelve la posición del primer nodo de la lista (o NULO si la lista está vacía)
- ultimo (tLista) → tPos
 - Devuelve la posición del último nodo de la lista (o NULO si la lista está vacía)
- siguiente (tLista, tPos) → tPos
 - Devuelve la posición del siguiente nodo en la lista (o NULO si la lista está vacía o la posición no tiene siguiente)
 - PreCD: La posición tiene que ser válida.

- anterior (tLista, tPos) → tPos
 - Devuelve la posición del anterior nodo en la lista (o NULO si la lista está vacía o la posición no tiene anterior)
 - PreCD: La posición tiene que ser válida.
- insertar (tLista, tInfo) → tLista
 - Inserta un tInfo al final de la lista.
- borrar (tLista, tPos) → tLista
 - Borra de la lista el nodo que está en la posición indicada.
 - PreCD: La posición tiene que ser válida.
 - PostCD: Las posiciones correspondientes a elementos que están a continuación del borrado dejan de ser posiciones válidas.
- obtenerDato (tLista, tPos) → tInfo
 - Devuelve el dato situado en la posición indicada de la lista.
 - PreCD: La posición tiene que ser válida.
- actualiza (tLista, tPos, tInfo) → tLista
 - Dada una posición de la lista actualiza el dato que contiene con el nuevo dato introducido por parámetro.
 - PreCD: La posición tiene que ser válida.

3 Implementación del TAD tLista

Deberán realizarse dos implementaciones del TAD tLista, una estática basada en arrays que deberá situarse en la unit ListaEstatica (fichero ListaEstatica.pas) y una dinámica basada en punteros que deberá situarse en la unit ListaDinamica (fichero ListaDinamica.pas).

La práctica incluirá un único programa principal (fichero principal.pas) con una cláusula uses que podrá ser:

- uses ListaEstatica; Si se utiliza la implementación estática, o
- uses ListaDinamica; Si se utiliza la implementación dinámica

El programa DEBE FUNCIONAR CORRECTAMENTE con ambas implementaciones.

4 Descripción del problema

4.1 Enunciado

Se trata de simular el funcionamiento de una biblioteca a la hora de gestionar el servicio de préstamos.

De cada libro tenemos la siguiente información (tInfo):

- Título del libro.

- Año de edición.
- Número de préstamos realizados

La biblioteca está muy mal gestionada y sólo mantiene una lista NO ORDENADA de los libros que posee. Ten en cuenta que pueden existir varias copias de un mismo título.

El programa realizará las siguientes acciones:

- Obtener un listado de todos los libros que existen en la biblioteca editados en el año actual (si hay varias copias de un libro se incluyen todas en el listado)
- Determinar la media global de préstamos que realiza el servicio de la biblioteca (si un libro está prestado 3 veces y otro 7 la media global es 5 préstamos por libro).
- Eliminar los ejemplares duplicados de la lista (ya que la biblioteca tiene poco espacio) mostrando un listado de los que quedan en la biblioteca.

4.2 Implementación

El funcionamiento del programa será en modo batch (sin interactividad con el usuario):

- El primer paso consistirá en leer un fichero de texto (libros.txt) con datos sobre los libros y representarlo en memoria como una lista.
- El segundo paso consistirá en realizar en la lista las acciones indicadas en el apartado anterior.
- Los resultados de cada acción se mostrarán por pantalla.

Para facilitar el desarrollo de la práctica se proporciona un fichero “lectura.pas” situado en el directorio P0 del directorio de prácticas (ver normas de entrega). Este fichero contiene la siguiente operación que se proporciona para facilitar las operaciones de lectura de libros:

- leerLibros(tLista, string) → tLista
 - Acepta como parámetro una variable de tipo tLista sin inicializar y un string que representa el nombre de un fichero en el que se incluyen los libros. Devuelve la lista que representa el contenido del fichero en memoria.
 - Es IMPORTANTE tener en cuenta que la operación leerLibros llama a funciones del TAD tLista. En consecuencia, para que su funcionamiento sea correcto la sintaxis de las llamadas a las operaciones de tLista, deben coincidir con lo implementado en las units.

5 Normas de realización y entrega

- Las prácticas serán INDIVIDUALES y OBLIGATORIAS.
- La entrega de TODAS las prácticas en las FECHAS indicadas es requisito IMPRESCINDIBLE para aprobar la asignatura en la convocatoria ordinaria de JUNIO. Para las convocatorias de SEPTIEMBRE y DICIEMBRE las prácticas serán las mismas pero se fijarán otras fechas de entrega.
- Las prácticas entregadas tendrán que ser ejecutables en Free Pascal en la máquina LIMIA.

- **Fecha límite de entrega: 30 de Abril de 2004.**
- Forma de entrega: Las prácticas quedarán depositadas en la red. No se admitirán discos ni papel. El proceso para depositarlas será el siguiente:
 1. Conectarse al host ALBA y situarse en los directorios:
 - /PRACTICAS/ETIX/EDI/P1 (alumnos de **I.T.I. de Gestión**)
 - /PRACTICAS/ETIS/EDI/P1 (alumnos de **I.T.I. de Sistemas**)
 - /PRACTICAS/EI/EDI/P1 (alumnos de **I. Informática**)
 2. Situarse en el directorio que coincida con el login del usuario y copiar allí la práctica (sólo los ficheros fuentes, no los ejecutables). **IMPORTANTE:** El nombre del fichero fuente que contenga el programa principal deberá ser principal.pas
 3. Pasada la fecha de entrega no se permitirá el acceso a estos directorios.
- Estructura que deberá de tener cada programa desarrollado de la práctica
 - Encabezamiento del programa. Constará la siguiente información entre comentarios


```
TÍTULO: Prácticas de EDI
SUBTÍTULO: Practica 1
AUTOR: _____
LOGIN: _____
GRUPO: E.I / E.T.I.X. / E.T.I.S
FECHA: __/__/____
```
 - Cuerpo del programa
 - El código irá comentado. Los comentarios han de ser concisos pero explicativos.
 - Después de la cabecera de cada procedimiento o función se incluirá lo siguiente: objetivo, entradas, salidas, precondiciones (condiciones que han de cumplir las entradas para el correcto funcionamiento de la subrutina) y poscondiciones (otras consecuencias de la ejecución de la subrutina que no quedan reflejadas en la descripción del objetivo o de las salidas)
- Criterios de valoración de la práctica:
 - *Eficacia:* que se cumplan las especificaciones
 - *Control de Errores:* Control intensivo de todos los errores de ejecución que sea posible detectar
 - *Claridad:* que el programa se pueda entender con facilidad, que contenga comentarios oportunos, indentación adecuada, nombres de variables significativos, etc.
 - *Modular:* que esté construido con módulos intercambiables y reutilizables.