

Práctica Listas

Estructuras de Datos

El comportamiento de un TAD Lista viene determinado por su especificación formal, que se muestra a continuación:

ESPECIFICACIÓN Listas

PARÁMETROS FORMALES

TIPOS TipoElemento

FIN PARAMETROS

TIPOS TipoLista

OPERACIONES

(* operaciones constructoras generadoras *)

CrearVacía: \rightarrow TipoLista

Construir: TipoElemento x TipoLista \rightarrow TipoLista

(* operaciones observadoras selectoras *)

PARCIAL Primero: TipoLista \rightarrow TipoElemento

PARCIAL Resto: TipoLista \rightarrow TipoLista

(* operaciones observadoras no selectoras *)

EsVacía: TipoLista \rightarrow Booleano

Longitud: TipoLista \rightarrow Natural

PARCIAL Ultimo: TipoLista \rightarrow TipoElemento

Pertenece: TipoElemento x TipoLista \rightarrow Booleano

(* operaciones constructoras no generadoras *)

Concatenar: TipoLista x TipoLista \rightarrow TipoLista

BorrarElemento: TipoElemento x TipoLista \rightarrow TipoLista

InsertarFinal: TipoElemento x TipoLista \rightarrow TipoLista

VARIABLES

lista, lista2: TipoLista;

e, e': TipoElemento;

ECUACIONES DE DEFINITUD

DEF(Primero (Construir (e, lista)))

DEF(Resto (Construir (e, lista)))

DEF(Ultimo (Construir (e, lista)))

ECUACIONES

(* operaciones observadoras selectoras *)

Primero (Construir (e, lista)) = e

Resto (Construir (e, lista)) = lista

(* operaciones observadoras no selectoras *)

EsVacía (CrearVacía) = CIERTO

EsVacía (Construir (e, lista)) = FALSO

Longitud (CrearVacía) = 0

Longitud (Construir (e, lista)) = 1 + Longitud (lista)

Ultimo (Construir (e, lista)) =

SI EsVacía(lista) \rightarrow

e

SI NO

Ultimo (lista)

Pertenece (e, CrearVacía) = FALSO

Pertenece (e, Construir (e', lista)) = (e = e') O (Pertenece (e, lista))

```

(* operaciones constructoras no generadoras *)
Concatenar (CrearVacia, lista) = lista
Concatenar (Construir (e, lista), lista2) =
    Construir (e, Concatenar (lista, lista2))

BorrarElemento (e, CrearVacia) = CrearVacia
BorrarElemento (e, Construir (e', lista)) =
    SI e = e' →
        lista
    SI NO
        Construir (e', BorrarElemento (e, lista))

InsertarFinal (e, CrearVacia) = Construir (e, lista)
InsertarFinal (e, Construir (e', l)) = Construir (e',
InsertarFinal (e, l))

```

FIN ESPECIFICACIÓN Listas

Para manejar la implementación de listas, en esta práctica, se pide

- a) Implementación de la unidad `ListaDin`, como una lista dinámica doble enlazada con puntero cabecera y final.
- b) Implementación de la unidad `ListaEst`, como una lista estática que simule memoria dinámica.
- c) Diseñar un programa que lea un fichero binario con tipo base:

```

TAlumno=RECORD
    expediente: integer;
    curso: integer;
    nombre: string;
    edad: integer;
    nota: real;
END;

TArchivo=FILE OF TAlumno;

```

y que a partir de los datos leídos en este archivo, por cada implementación, construya una lista de elementos que contenga a todos los estudiantes y además implemente la funcionalidad necesaria para:

- 1) Determinar si un estudiante está matriculado.
- 2) Calcular la nota media de un curso.
- 3) Determinar el número de estudiantes aprobados por curso.
- 4) Conocer el nombre del estudiante con notas más elevadas.
- 5) Obtener el número de expediente del estudiante más joven.

Nota 1: Se considera necesario un nivel elevado de encapsulación y abstracción.

Nota 2: Se hará uso de las normas de estilo dictadas en clase (cabecera del fichero, interfaz de la unidad con precondiciones, postcondiciones, excepciones, implementaciones con el análisis de complejidad de cada operación, nombres coherentes de variables y operaciones,...)

Plantilla de cabecera del fichero:

```

{*****
*
*      Módulo:
*      Tipo: Programa ()      Interfaz-Implementación TAD ()      Otros ()
*      Autor/es:
*      Fecha de actualización:
*      Descripción:
*
*****}

```


