

Práctica 3: Implementación del TAD Lista

Estructuras de Datos y de la Información

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 *)

CrearVacia: \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 *)

EsVacia: 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 *)

EsVacia (CrearVacia) = CIERTO

EsVacia (Construir (e, lista)) = FALSO

Longitud (CrearVacia) = 0

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

Ultimo (Construir (e, lista)) = SI EsVacia(lista) → e
SI NO Ultimo (lista)

Pertenece (e, CrearVacia) → 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', lista)) = Construir (e', InsertarFinal (e, lista))

FIN ESPECIFICACIÓN Listas

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

- Implementación de la unidad ListaDin, como una lista dinámica enlazada simple.
- Implementación de la unidad ListaDin2, como una lista dinámica doblemente enlazada.
- Implementación de la unidad ListaEst, como una lista estática que simule memoria dinámica.
- Diseñar un programa de prueba único para las tres implementaciones y un TAD Elemento adecuado al enunciado que se describe a continuación. Este programa deberá:
 - Construya una lista de números que contenga a la siguiente secuencia: 7 9 6 8 12 -5 4 0.
 - Informe del número de elementos que contiene la lista.
 - Compruebe si el número 8 está en la lista y muestre un mensaje por pantalla.
 - Compruebe si el número 3 está en la lista y muestre un mensaje por pantalla.
 - Inserte el elemento 3 en la lista.
 - Informe del número de elementos que contiene la lista.

- 7) Muestre por pantalla el último elemento de la lista.
- 8) Muestre por pantalla todos los elementos de la lista.
- 9) Borre el número 0.
- 10) Muestre todos los elementos de la lista por pantalla.
- 11) Inserte el número 2 al final de la lista.
- 12) Muestre por pantalla el número de elementos que tiene la lista.
- 13) Muestre por pantalla el último elemento de la lista.
- 14) Construir una segunda lista con la siguiente secuencia 4 0.
- 15) Concatenar la primera lista y la segunda.
- 16) Mostrar por pantalla la lista resultado.
- 17) Indicar el número de elementos que tiene la lista resultado.

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:
*
* ****
**** }
```