PRÁCTICA Nº 5: 2 sesiones

(del 8 de Mayo al 17 de Mayo de 2001)

Estructuras de datos lineales. Listas (con punto de interés): Representación estática y dinámica. Gestión de una biblioteca

0. OBJETIVO

El objetivo de esta práctica es la realización de dos clases en C++ que implementen el tipo abstracto de datos LISTA con punto de interés (estática y dinámica), así como su utilización en un programa para la 'gestión' de libros.

1. EL TIPO ABSTRACTO DE DATOS LISTA

Las Listas con punto de interés son estructuras lineales generales, en las que dentro del conjunto de elementos, existe uno destacado, al que denominaremos punto de interés, y que será la referencia para todas las operaciones que se realicen en la lista.

Las operaciones de las que consta el TAD LISTA con punto de interés, tal y como vimos en clase, pueden ser las siguientes

INICIAR_LISTA ( ) → Lista
Inicia la lista a vacía, con solo el punto 'final de la lista' y el punto de interés sobre este 'final de la lista'

INSERTAR (Lista , Valor) → Lista
Inserta el Valor delante del elemento señalado como punto de interés, dejando el punto de interés sobre el elemento en el que estaba.

ELIMINAR (Lista) → Lista, Valor
Modifica la lista eliminando el elemento señalado por el punto de interés, dejando el punto de interés sobre el elemento siguiente al eliminado. Si el punto de interés señalaba el final de la lista devolverá error.

CONSULTAR (Lista) → Valor
Devuelve la información señalada por el punto de interés. Si el punto de interés señalaba el final de la lista devolverá error.

LISTA_VACIA (Lista) → Lógico
Nos informa si existe información en la lista o no (nos dice si el número de elementos de la lista es cero o diferente de cero.)

IR_A_INICIO (Lista) → Lista
Mueve el punto de interés al primer elemento de la lista.

AVANZAR (Lista) → Lista
Avanza el punto de interés al siguiente elemento de la lista.

FINAL_LISTA (Lista) → Lógico
Nos dice si el punto de interés está o no al final de la lista.

2. IMPLEMENTACIÓN DE LAS CLASES: LISTA ESTÁTICA Y DINÁMICA

Se trata de implementar dos clases en C++ que implementen el tipo abstracto de datos Lista con punto de interés, de forma estática (vector sin cursores) y dinámica (punteros.) Ambas clases tendrán el mismo nombre, de forma que después pueda ser utilizada una clase u otra en el programa principal, sin necesidad de cambio alguno en el código.

Los métodos de la clase Lista, tanto en el caso estático, como en el dinámico, serán:

```
class Lista
{
   public:
      Lista (void);
```

```
        Lista (const Lista &);        /* Constructor de copia si fuese necesario*/
        ~Lista (void);                /* Destructor de clase  */

        bool Insertar (Valor x);
        bool Eliminar (Valor & x);
        bool Consulta (Valor & x);
        bool ListaVacia (void);

        void IrAInicio (void);
        bool Avanzar (void);
        bool FinalLista (void);

    private:
        ????
};
```

Si se cree conveniente, y solo para efectos de depuración del programa podría implementarse un método que mostrase todo el contenido de la lista sin modificar el punto de interés, e indicando la situación de éste.

3. PROGRAMA DE GESTIÓN DE BIBLIOTECA

Se trata de escribir un programa en C++ que utilizando la estructura de datos 'lista', implemente una sencilla base de datos de libros, sobre la que se podrán realizar las siguientes operaciones:

- Cargar el fichero en una lista. El nombre del fichero lo indica el usuario, en ningún caso debe ser una información incluida en el código del programa.
- Mostrar lista ordenada de libros por título.
- Mostrar lista de libros publicados en un cierto año.
- Buscar un cierto libro y mostrar su ficha completa.
- Dar de alta un determinado libro, pidiendo todos sus datos.
- Dar de baja un determinado libro, buscándolo por su título, pidiendo confirmación antes de borrarlo. Si hubiese títulos repetidos, se buscarían todos los libros con ese título y se pediría confirmación para cada uno de ellos.
- Guardar, si se desea y la lista ha sido modificada, la lista de libros en el fichero original al terminar el programa.

La información de cada libro incluida en el fichero es la siguiente:

```
Nombre del libro
Autor
ISBN
Año de publicación
```

ocupando cada libro cuatro líneas del fichero en formato ASCII.

Evidentemente habrá que controlar los errores. Por ejemplo, no se podrá consultar la base de datos si no se ha cargado un fichero en la lista.

4. ENTREGA DE PROGRAMAS

Al inicio de la sesión de prácticas correspondiente a la semana del 22 al 24 de mayo se entregarán al profesor los archivos '**`list_e##.h`**', '**`list_e##.cpp`**', '**`list_d.h`**', '**`list_d##.cpp`**' correspondientes al TAD lista con vectores y con punteros, y el programa principal '**`biblio##.cpp`**'.

5. TRABAJO ADICIONAL

Se propone como tarea adicional la implementación de la clase Lista (con punto de interés) basada en el tipo Pila.

El trabajo adicional supondrá la entrega de 4 ficheros más: "**`pila##.h`**", "**`pila##.cpp`**", "**`lista_p##.h`**"

y "**`lista_p.cpp`**".

La implementación que se haga de las pilas podrá ser estática o dinámica.