Práctica 11

Queremos gestionar los datos de una bibliografía de libros (concretamente, para cada libro de la bibliografía almacenaremos el ISBN – una cadena de caracteres de tamaño 10 que actuará como identificador del libro – el título y el año de publicación), de forma que queremos disponer de listados de libros ordenados de manera creciente por ISBN y listados de manera decreciente por año (teniendo en cuenta que vamos a usar bastante más los ordenados por ISBN).

La aplicación comenzará mostrando el siguiente menú, a partir del cual el usuario podrá elegir distintas operaciones.

```
1.- Crear bibliografía
2.- Añadir libro
3.- Modificar año de libro
4.- Listar libros por ISBN
5.- Listar libros por orden decreciente de año
6.- Terminar
Elegir opción:
```

Vamos a dar una posible solución a este problema.

1. Definir el TAD Libro, cuyo género es `tlibro`, para poder trabajar con la información relacionada con los libros de una determinada bibliografía.
    1. Considerar una **especificación** que contenga al menos las siguientes operaciones: crear un libro, modificar el año de publicación de un libro, obtener el título de un libro, obtener el año de publicación de un libro, y obtener el ISBN de un libro.
    2. Proporcionar una **implementación** de este TAD.

    Para implementar el TAD Libro, crea los ficheros Libro.cpp y Libro.h.
2. **Como usuarios** del TAD Libro, definir una serie de operaciones auxiliares que, aunque no aparezcan en la especificación del TAD, pueden resultar útiles para los usuarios del mismo. Dichas operaciones son las siguientes: leer del teclado los datos de un libro, mostrar por pantalla los datos de un libro y realizar una copia de los datos de un libro. Para implementar dichas operaciones no puedes utilizar la implementación concreta del TAD Libro, sino que sólo puedes utilizar las operaciones dadas en la especificación.

    Implementa estas operaciones adicionales en los ficheros UsaLibro.h y UsaLibro.cpp.
3. Para el TAD Bibliografía, cuyo género será `tbibliografia`:
    1. Definir el TAD con la **especificación** dada en la última página.

2. Dar una **implementación dinámica** del TAD pensando en una implementación lo más eficiente posible. Para dicha implementación solo puedes utilizar las operaciones definidas en la especificación del TAD Libro no su implementación concreta.

Para implementar el TAD Bibliografía, crea los ficheros Bibliografia.cpp y Bibliografia.h.

4. **Como usuarios** del TAD Bibliografía, definir una serie de operaciones auxiliares que, aunque no aparezcan en la especificación del TAD, pueden resultar útiles para los usuarios del mismo. Dichas operaciones son las siguientes: crear una copia de una bibliografía, mostrar por pantalla los libros de una bibliografía por orden creciente de su ISBN, y mostrar por pantalla los libros de una bibliografía por orden decreciente de año de publicación. Para implementar dichas operaciones no puedes utilizar la implementación concreta del TAD Bibliografía, sino que sólo puedes utilizar las operaciones dadas en su especificación.

   Implementa estas operaciones adicionales usando los ficheros UsaBibliografia.h y UsaBibliografia.cpp.

5. El fichero Principal.cpp se encargará de mostrar el menú indicado en la primera página de la práctica y de gestionar cada una de sus operaciones. Después de ejecutar la operación seleccionada por el usuario, el programa volverá a mostrar el menú. Las operaciones implementadas en este fichero no podrán usar la implementación concreta de los TADs Bibliografía y Libro, solo las operaciones definidas en su especificación, o en los ficheros UsaLibro y UsaBibliografia.

Nota: para comparar lexicográficamente dos cadenas de caracteres `<cad1>` y `<cad2>` en C++ se utiliza la función `strcmp(<cad1>, <cad2>)` que devuelve un entero: positivo si `<cad1>` es mayor que `<cad2>`,0 si son iguales y un entero negativo si `<cad1>` es menor que `<cad2>`.

```
void crear(tbibliografia &b);
{Pre: }
{Post: Inicia b como una bibliografía sin libros.}

void anadir(tbibliografia &b, tlibro l);
{Pre: }
{Post: Añade en orden alfabético, basándose en el ISBN, el libro l
a la bibliografía b.}

void eliminar(tbibliografia &b, char ISBN[]);
{Pre: ISBN es un ISBN de los libros de la bibliografía b}
{Post: Elimina el libro cuyo ISBN es ISBN de la bibliografía b.}

void extraer(tbibliografia b, char ISBN[], tlibro & l);
{Pre: ISBN es un ISBN de los libros de la bibliografía b}
{Post: Devuelve de la bibliografía b los datos del libro cuyo ISBN
es ISBN.}
```

```
void extraerPosicion(tbibliografia b, int pos, tlibro & l);
{Pre: pos es un entero menor que el número de libros de la
bibliografía b}
{Post: Devuelve de la bibliografía b los datos del libro que ocupa
la posición pos, empezando a contar en 0, en la bibliografía.}

bool existe (tbibliografia b, char ISBN[]);
{Pre: }
{Post: Devuelve true si en la bibliografía b hay un libro cuyo ISBN
es ISBN, y devuelve false en caso contrario.}

void modificarAnioLibro(tbibliografia &b, char ISBN[], float anio);
{Pre: ISBN es un ISBN de los libros de la bibliografía b}
{Post: Modifica el año de publicación del libro de la bibliografía
b cuyo ISBN es ISBN poniendo como nuevo año de publicación anio.}

void masNuevo(tbibliografia b, tlibro & l);
{Pre: la bibliografía b no debe estar vacía}
{Post: l es el libro cuyo año de publicación es el mayor de la
bibliografía b. En caso de haber varios libros con el mismo año de
publicación, solo devuelve uno.}

bool bibliografiaSinLibros (tbibliografia b);
{Pre: }
{Post: Devuelve true si la bibliografía no tiene libros y false en
caso contrario.}

int numeroLibros (tbibliografia b);
{Pre: }
{Post: Devuelve el número de libros de la bibliografía b.}
```