

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJE DE PROGRAMACIÓN 1

10ma práctica (tipo b)
(Primer Semestre 2018)

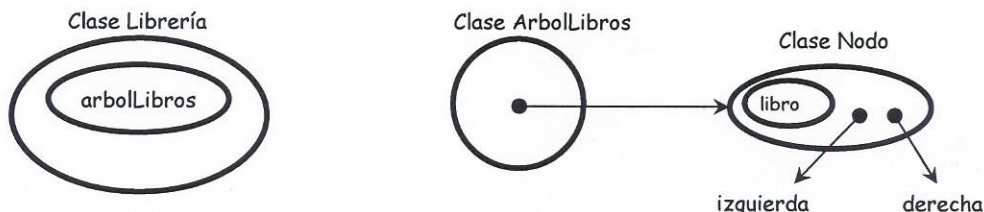
Indicaciones Generales:

- Duración: 110 minutos.
- Se podrá usar como material de consulta solo sus apuntes de clase.
- No se pueden emplear variables globales, ni estructuras. No se podrá emplear la clase string, las plantillas definidas en la STL, ni las funciones malloc, realloc, strdup, strtok, tampoco las funciones incluidas en las bibliotecas cstdio, stdio.h o similares. La cláusula "friend" solo se podrá emplear en la sobrecarga de >> y <<. No se podrán definir funciones (ni plantillas) independientes que no estén ligadas como métodos a alguna de las clases planteadas.
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE. Cada módulo NO debe sobrepasar las 20 líneas de código aproximadamente. El archivo **main.cpp** solo podrá contener la función **main** de cada proyecto. En cada archivo que implemente en los proyectos (.h y .cpp) deberá colocar un comentario en el que coloque claramente su nombre y código, de no hacerlo se le **descontará 0.5 puntos por archivo**. Se tomará en cuenta en la calificación el uso de comentarios relevantes.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% del puntaje de la pregunta. Los que no den resultados coherentes en base al 60%.
- La **presentación, la ortografía y la gramática de los trabajos influirá en la calificación.**

Puntaje total: 20 puntos

Problema

Se desea elaborar una aplicación orientada a objetos que permita gestionar los libros de una librería mediante un árbol binario de búsqueda (ABB). Para manipular la información se desea cargar los datos de los archivos con los que se trabajará en objetos cuyas estructuras serán similares a la mostrada en la Figura 1:



Donde:

- "**libro**", es un objeto de clase "Libro". Esta clase estará compuesta por los atributos: codigo (char *), titulo (char*), autor (char*), precio (double), nVend (número de libros vendidos - int), monto (monto recaudado - double). La clase definirá por lo menos su constructor por defecto, su destructor, métodos selectores para todos sus atributos, una sobrecarga del operador ++ que permita incrementar en uno el número de libros vendidos y acumular el monto recaudado y la sobrecarga de los operadores de inserción y extracción de flujo << y >> hacia y desde archivos de textos de los datos de cada libro. (6 puntos)
- "**Clase ArbolLibros/Clase Nodo**", permitirá definir un ABB. Los métodos del árbol permitirán crear el árbol, insertar un libro en el árbol de modo que quede ordenado por el código, también permitirá buscar y modificar los datos de un libro, finalmente imprimir en un archivo de textos los datos de los libros ordenados tabulados y etiquetados correctamente sin usar '\t'). (8 puntos)
- "**Clase Librería**", esta clase permitirá crear el árbol, leyendo los datos desde un archivo, determinar el número de libros vendidos y el monto recaudado, leyendo la información de otro archivo y finalmente mostrar todos los datos de los libros en un archivo de reporte. La clase tendrá un único atributo de esta clase será un objeto (no un puntero) de clase ArbolLibros y los métodos convenientes. (6 puntos)

Los datos se leerán de dos archivos de CSV con siguiente información:

Archivo 1
PQ4355, Lenguaje de programación C++, Paul Disgram, 85.75
RM6523, Métodos numéricos, Scheid DiConstanzo, 25.30
...

Archivo 2
10/05/2108,10:20:30,MG0507
10/05/2018,10:23:10,PQ355
...

En el primer archivo se encuentra la lista de libros que vende la librería, aquí aparece el código, título, autor y precio del libro. En el segundo archivo aparece las ventas que se han hecho de los libros en un periodo de tiempo, en cada línea aparece una fecha, la hora y código del libro que se vendió los libros pueden aparecer varias veces en este archivo. Los datos no se encuentran ordenados en ninguno de los archivos.

Consideraciones:

- Cada clase deberá definirse en dos módulos independientes (archivo .h y .cpp). No se podrá definir la implementación de los métodos en los archivos .h.
- Deberá respetar estrictamente el encapsulamiento de datos a todo nivel.
- Deberá definir los constructores y destructores en las clases que lo requieran.

Anotaciones finales

Al finalizar el laboratorio, comprima¹ la carpeta "Laboratorio10" en un archivo con nombre <código del alumno con 8 dígitos>.zip y súbalo a la intranet del curso, en el enlace Documentos, en la carpeta \Laboratorio7\<código del horario>\<aula>.

Profesor del curso: Miguel Guanira E.

San Miguel, 22 de junio del 2018.

¹ Para evitar problemas en la corrección de la prueba, utilice el programa de compresión que viene por defecto en el Windows (Zip) no 7z.