

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

9na práctica (tipo b)

Primer Semestre 2024

Indicaciones Generales:

Duración: 110 minutos.

**NO SE PERMITE EL USO DE APUNTES DE CLASE, FOTOCOPIAS NI MATERIAL IMPRESO**

- No se pueden emplear **variables globales**, **NI OBJETOS** (con excepción de los elementos de `iostream`, `omanip` y `fstream`). **NO PUEDE UTILIZAR LA CLASE `string`**. Tampoco se podrán emplear las funciones de C que gestionen memoria como `malloc`, `realloc`, `memset`, `strdup`, `strtok` o similares, igualmente no se puede emplear cualquier función contenida en las bibliotecas `stdio.h`, `cstdio` o similares y que puedan estar también definidas en otras bibliotecas. **NO PODRÁ EMPLEAR PLANTILLAS EN ESTE LABORATORIO**
- Deberá modular correctamente el proyecto en archivos independientes. LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN ESTRICTO DISEÑO DESCENDENTE. **Cada función NO debe sobrepasar las 20 líneas de código aproximadamente**. El archivo `main.cpp` solo podrá contener la función `main` de cada proyecto y el código contenido en él solo podrá estar conformado por tareas implementadas como funciones. En el archivo `main.cpp` deberá colocar un comentario en el que coloque claramente su nombre y código, **de no hacerlo se le descontará 0.5 puntos en la nota final**.
- El código comentado NO SE CALIFICARÁ. De igual manera NO SE CALIFICARÁ el código de una función si esta función no es llamada en ninguna parte del proyecto o su llamado está comentado.
- Los programas que presenten errores de sintaxis o de concepto se calificarán en base al 40% de puntaje de la pregunta. Los que no muestren resultados o que estos no sean coherentes en base al 60%.
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.
- Deberá mantener en todo momento el encapsulamiento de todos los atributos de las clases, así como guardar los estándares en la definición y uso de todas las clases desarrolladas. No está permitido el casteo de objetos. No se considerará en la nota las clases que violen esto.
- Salvo en la sobrecarga de los operadores `>>` y `<<`, no se podrán definir funciones (ni plantillas de funciones) independientes que no estén ligadas como métodos a alguna de las clases planteadas. Tampoco se podrá emplear la cláusula `protected` ni la cláusula `friend`, solo puede utilizarse en los casos indicados explícitamente.

**SE LES RECUERDA QUE, DE ACUERDO AL REGLAMENTO DISCIPLINARIO DE NUESTRA INSTITUCIÓN, CONSTITUYE UNA FALTA GRAVE COPIAR DEL TRABAJO REALIZADO POR OTRA PERSONA O COMETER PLAGIO.**

**NO SE HARÁN EXCEPCIONES ANTE CUALQUIER TRASGRESIÓN DE LAS INDICACIONES DADAS EN LA PRUEBA**

- **Puntaje total:** 20 puntos.

**INDICACIONES INICIALES**

Cree un proyecto de C++ en NetBeans siguiendo estrictamente las indicaciones que a continuación se detallan:

- La unidad de trabajo será **t:\** (Si lo coloca en otra unidad, no se calificará su laboratorio y se le asignará como nota cero)
- Cree allí una carpeta con el nombre **"CO\_PA\_PN\_Lab09\_2024\_1"** donde **CO** indica: Código del alumno, **PA** indica: Primer Apellido del alumno y **PN** primer nombre (de no colocar este requerimiento se le descontarán 3 puntos de la nota final). **Allí colocará los proyectos solicitados en la prueba.**

**Cuestionario:**

Se solicita que desarrolle un proyecto **"LAB09\_ABBSTL"** dentro de la carpeta correspondiente, **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 2 PUNTOS DE LA NOTA FINAL**, en la cual se declaren las clases descritas con las relaciones necesarias, que permitan manipularlas empleando herencia:

- **Para manejar los libros:** La clase se denominará **"Libro"** y deberá contener lo siguiente: 1) un atributo denominado **nombre** (**char\***) definido por una cadena dinámica de caracteres que representa el título del

libro, 2) un atributo denominado **paginas** (**int**) que representa la cantidad de páginas que tiene el libro, 3) un atributo denominado **peso** (**double**) que representa el peso del libro.

➤ **Para manejar los libros tipo enciclopedia:** La clase se denominará "**Enciclopedia**" y deberá contener lo siguiente: 1) un atributo denominado **sku** (**int**) que representa el código de la enciclopedia, 2) un atributo denominado **anho** (**int**) que representa el año que fue publicada la enciclopedia. Esta clase posee datos heredados de la clase **Libro**.

➤ **Para manejar los libros tipo novela:** La clase se denominará "**Novela**" y deberá contener lo siguiente: 1) un atributo denominado **autor** (**char\***) definido por una cadena dinámica de caracteres que representa el nombre del autor del libro. Esta clase posee datos heredados de la clase **Libro**.

➤ **Para manejar los libros tipo revista:** La clase se denominará "**Revista**" y deberá contener lo siguiente: 1) un atributo denominado **ISSN** (**int**) que representa el código ISSN de la publicación, 2) un atributo denominado **anho** (**int**) que representa el año que fue publicada la revista, 3) un atributo denominado **numero** (**int**) que representa de la revista. Esta clase posee datos heredados de la clase **Libro**.

➤ **Para manejar los libros dentro del STL-Vector:** La clase se denominará "**NLibro**" y deberá contener lo siguiente: 1) un atributo denominado **plibro**, este es un puntero de la clase **Libro**.

➤ **Para manejar los estantes:** La clase se denominará "**NEstante**" y deberá contener lo siguiente: 1) un atributo denominado **id** (**int**), este atributo guardará el identificador del estante, 2) un atributo denominado **capacidad** (**double**), este atributo almacena el peso máximo que puede soportar el estante, 3) un atributo denominado **disponible** (**double**), este atributo almacena el peso que aún puede soportar el estante, inicialmente tendrá el mismo valor que **capacidad** pero cada vez que se le asigna un libro al estante se debe disminuir el peso del libro colocado, 4) un atributo denominado **vlibros**, este atributo es un STL-Vector de la clase **NLibro**, y sirve para guardar todos los libros que se asignan al estante, 5) un atributo denominado **izq**, este atributo es un puntero a la clase **NEstante** (autoreferenciado), 6) un atributo denominado **der**, este atributo es un puntero a la clase **NEstante** (autoreferenciado).

➤ **Para manejar el ABB de estantes:** La clase se denominará "**Arbol**" y deberá contener lo siguiente: 1) un atributo denominado **raiz**, este atributo es un puntero de clase **NEstante**. Este árbol esta ordenado por el atributo **disponibilidad** de cada estante. Puede revisar el proceso de llenado del árbol en el detalle del método **carga**.

➤ **Para manejar la biblioteca:** La clase se denominará "**Biblioteca**" y deberá contener lo siguiente: 1) un atributo denominado **AEstante** dado por un objeto de la clase **Arbol**, donde se guardarán los estantes y dentro de cada estante los libros.

**"DEBE EMPLEAR OBLIGATORIAMENTE LOS NOMBRES DE LAS CLASES Y SUS ATRIBUTOS"**

#### **Consideraciones:**

Para el desarrollo de la pregunta debe considerar el siguiente código:

```
#include "Biblioteca.h"

using namespace std;

int main(int argc, char** argv) {
    Biblioteca bli;

    bli.carga();
    bli.muestra();
    bli.prueba(8);

    return 0;
}
```

**NO PUEDE  
CAMBIAR  
ESTE CÓDIGO**

Con las clases indicadas debe realizar las siguientes operaciones:

- (2 puntos) Implementar todas las clases indicadas en el enunciado anterior.
- (10 puntos) En la clase **Biblioteca** implementar el método **carga**, que se encarga de la lectura del archivo "**Estantes3.csv**" y generar un objeto de la clase **NEstante** que representa al estante respectivo. Antes de ubicar el estante en el árbol **AEstante** debe leer el archivo "**Libros31.csv**" y cargar 10 libros en el estante, recuerde que los libros van en el STL-Vector **vlibros**, el archivo contiene la información exacta para el llenado de 10 libros para cada estante, sin sobrepasar su capacidad. Debe leer el archivo una sola vez para llenar los estantes. Para la lectura de los datos correspondiente de cada libro debe utilizar el método polimórfico **lee**, ya que los mismos pueden ser novelas, enciclopedias o revistas. Para diferenciar cada tipo de libro en el archivo correspondiente las novelas se representan con la letra "N", las enciclopedias con la "E" y las revistas con la "R". Con el objeto de la clase **NEstante** cargado de 10 libros y el atributo **disponible** actualizado, debe insertarlo en **Arbol** recordando que se trate de un ABB ordenado por el atributo **disponible**. Para esta pregunta puede emplear un puntero de la clase **NEstante** como parámetro de los métodos relacionados a su almacenamiento.
- (4 puntos) En la clase **Biblioteca** implementar el método **muestra**, que se encarga de realizar la impresión de un archivo debidamente tabulado (**sin usar el carácter '\t'**), que muestre los datos de cada estante y los libros asignados de acuerdo con su tipo correspondiente. Ya que se trata de un ABB la impresión de los estantes debe ser **ENORDEN** de acuerdo con el atributo **disponible**. Para la impresión de los libros debe usar el método polimórfico **imprime**. El reporte debe tener el siguiente formato:

```
Estante: 4
Capacidad: 10      Disponible:0.15
=====
Titulo:Boletin de Arqueologia PUCP
Peso: 0.75
ISSN:10292004      Año: 2023

Titulo:Boletin de Arqueologia PUCP
Peso: 0.75
ISSN:10292004      Año: 2023

Titulo:LA ENCICLOPEDIA DE LOS DINOSAURIOS
Peso: 2
SKU:36225      Año: 2018
...

Estante: 2
Capacidad: 10      Disponible:2.30
=====
Titulo:Educacion
Peso: 0.2
ISSN:10199403      Año: 2022

Titulo:Educacion
Peso: 0.2
ISSN:10199403      Año: 2022
...
```

- (4 puntos) En la clase **Biblioteca** implementar el método **prueba**, que recibe como parámetro un número **double** que representa el peso de un libro, con el fin de saber si el peso recibido puede ser asignado en alguno de los estantes. Si es posible colocarlo en un estante, ya que el valor del número (peso) es menor o igual a la disponibilidad del estante, debe mostrar un mensaje indicando los datos del estante y su contenido, para los libros puede emplear el polimórfico **imprime**. Si no es posible colocarlo en ningún estante, debe emitir un mensaje indicando "**No es posible colocar el libro**". El mensaje debe generarse en un archivo de texto.

```
Si hay espacio
Estante: 3
Capacidad: 15      Disponible:8.15
=====
Titulo:Historica
Peso: 0.5
ISSN:2223375      Año: 2022

Titulo:Histórica
...
```

Se recomienda revisar los archivos que servirán para la lectura de datos, los cuales se describen a continuación:

Estantes3.csv
1,20
2,10
...
Identificador, Peso o Capacidad que soporta

Libros31.csv
N,Diamantes y pedernales,120,0.7,Jose Maria Arguedas
E,ENCICLOPEDIA SALVAT,2000,12,78323,2015
N,Confesiones de una mascara,80,0.8,Yukio Mishima
R,Educacion,48,0.2,10199403,2024,6
...
Tipo, Título, Páginas, Peso, Autor/ISSN/SKU, Año, Número

Recuerde que si no usa polimorfismo la respuesta no será válida, así mismo siempre debe mantener el encapsulamiento por lo cual debe realizar cada operación donde le corresponde, por ejemplo, si va a leer o imprimir un libro esta operación la debe realizar en la clase que le corresponde. La cláusula *friend* solo se puede emplear para acceder desde la TAD al nodo, en este caso desde el árbol a los nodos.

Al finalizar la práctica, comprima la carpeta dada en las indicaciones iniciales empleando el programa Zip que viene por defecto en el Windows, no se aceptarán los trabajos compactados con otros programas como RAR, WinRAR, 7zip o similares.

Profesores del curso:

Rony Cueva  
Erick Huiza  
Miguel Guanira

Erasmus Gómez  
Heider Sánchez

San Miguel, 21 de junio del 2024.