

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

10ma práctica (tipo b)

Primer Semestre 2023

Indicaciones Generales:

Duración: 110 minutos.

SOLO ESTÁ PERMITIDO EL USO DE APUNTES DE CLASE. NO PUEDE UTILIZAR FOTOCOPIAS NI MATERIAL IMPRESO, TAMPOCO PODRÁ EMPLEAR HOJAS SUELTAS.

- No se pueden emplear **variables globales, ni estructuras**. No puede utilizar la clase (o el tipo de datos) **string**. Tampoco se podrán emplear las funciones **malloc, realloc, strdup o strtok**, 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 ERICTO 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%.
- 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 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.
- La cláusula **friend** solo se podrá emplear en el caso de clases auto referenciadas para ligar el nodo con la clase inmediata que encapsula la lista, **en ningún caso adicional**. No se considerará en la nota las clases que violen esto. Tampoco se podrá emplear la cláusula **protected**.
- Se tomará en cuenta en la calificación el uso de comentarios relevantes.

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_Lab10_2023_1"** donde **CO** indica: Código del alumno, **PA** indica: Primer Apellido del alumno y **PN** primer nombre (de no colocar este requerimiento se le descontará 3 puntos de la nota final). **Allí colocará los proyectos solicitados en la prueba.**

Cuestionario:

La finalidad principal de este laboratorio es la de reforzar los conceptos contenidos en el capítulo 9 del tema: "Clases autoreferenciadas".

PARTE01 (12 puntos): CREACIÓN DE LAS CLASES

Se solicita que desarrolle un proyecto "LAB10_PREG01" 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 alumnos:** La clase se denominará "Alumno" y deberá contener lo siguiente: 1) un atributo denominado **codigo** (**int**), 2) un atributo denominado **nombre** definido por una cadena de caracteres dinámica, 3) un atributo denominado **escala** (**int**), 4) un atributo denominado **creditos** (**double**), 5) un atributo denominado **total** (**double**).
- **Para manejar las escalas:** La clase se denominará "Escala" y deberá contener lo siguiente: 1) un atributo denominado **codigo** (**int**), 2) un campo denominado **precio** (**double**).
- **Para manejar a los alumnos presenciales:** La clase se denominará "Presencial" y deberá contener lo siguiente: 1) un atributo denominado **recargo** (**double**), este campo almacena el porcentaje de recargo que tiene un alumno sobre su boleta, 2) un atributo denominado **total** (**double**), este campo almacena el monto correspondiente al recargo que se aplicará a la boleta. Además, esta clase posee datos heredados de la clase **Alumno**.
- **Para manejar a los alumnos semipresenciales:** La clase se denominará "Semipresencial" y deberá contener lo siguiente: 1) un atributo denominado **descuento** (**double**), este campo almacena el porcentaje de descuento que tiene un alumno sobre su boleta, 2) un atributo denominado **total** (**double**), este campo almacena el monto correspondiente al descuento que se aplicará a la boleta. Además, esta clase posee datos heredados de la clase **Alumno**.
- **Para manejar a los alumnos virtuales:** La clase se denominará "Virtual" y deberá contener lo siguiente: 1) un atributo denominado **licencia** definido por una cadena de caracteres dinámica, 2) un atributo denominado **total** (**double**), este campo almacena el monto correspondiente a la licencia, inicialmente en 0. Además, esta clase posee datos heredados de la clase **Alumno**.
- ☐ **Para manejar una boleta:** la clase se denominará "Boleta" y deberá contener lo siguiente: 1) un atributo denominado **pboleta**, este atributo es un puntero de la clase **Alumno**, y servirá para registrar los atributos del alumno que representan, de acuerdo con el tipo que tiene asignado.
- **Para manejar los Nodos:** La clase se denominará "Nodo" y deberá contener lo siguiente: 1) un atributo denominado **dboleta** de la clase **Boleta**, 2) un atributo denominado **izq**, este atributo es un puntero a la clase **Nodo** (autoreferenciado), 3) un atributo denominado **der**, este atributo es un puntero a la clase **Nodo** (autoreferenciado).
- **Para manejar el árbol:** La clase se denominará "Arbol" y deberá contener lo siguiente: 1) un atributo denominado **raiz**, este atributo es un puntero de clase **Nodo**. Esta estructura se trata de un ABB ordenado por el código del alumno (ascendente), 2) un atributo denominado **lescala**, este atributo es un arreglo estático de la clase **Escala**, aquí se guardarán todas las escalas existentes.
- **Para manejar todas las boletas:** La clase se denominará "Tesoreria" y deberá contener lo siguiente: 1) un atributo denominado **aboleta**, este atributo es un objeto de la clase **Arbol**, donde se guardarán todas las boletas de los alumnos a cargar.

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

Con las clases indicadas debe realizar las siguientes operaciones:

- En la clase **Tesoreria** debe implementar el método **cargaalumnos**, que se encarga de la lectura del archivo "alumnos.csv" y cargar la información en el árbol ABB denominada **aboleta**, de acuerdo con el tipo de alumno que indique el primer campo de cada fila del archivo leído. (P: Presencial, S: Semipresencial, V: Virtual). El atributo **total** de todas las clases debe considerarse en 0. Para este paso debe utilizar el método polimórfico **lee**.

- En la clase **Tesoreria** implementar el método **imprimeboleta**, que se encargue de realizar la impresión de un archivo de prueba debidamente tabulado (**sin usar el carácter '\t'**), que muestre el contenido del árbol **aboleta** in orden. Para este paso debe utilizar el método polimórfico **imprime**.

Para esta pregunta, por lo menos debe desarrollar los siguientes métodos polimórficos:

- lee**: para la lectura de los datos correspondientes a los alumnos de acuerdo con su tipo.
- imprime**: para la impresión de los datos de cada uno alumno de acuerdo con su tipo.

Consideraciones:

Para el desarrollo de ambas preguntas debe considerar el siguiente código, con excepción del método **actualizaboleta** que solo debe estar en la pregunta 2:

```
#include "Tesoreria.h"

using namespace std;
int main(int argc, char** argv) {
    Tesoreria ABoleta;

    ABoleta.cargaalumnos();
    ABoleta.actualizaboleta();
    ABoleta.imprimeboleta();

    return 0;
}
```

**NO PUEDE
CAMBIAR
ESTE CÓDIGO**

PARTE 2 (8 puntos): Proceso final.

Desarrolle un proyecto denominado "**LAB10_PREG02**" en el cual se utilizará obligatoriamente las clases desarrolladas en la pregunta anterior. El proyecto ejecutará las tareas descritas a continuación:

- Cargar el árbol **aboleta** de acuerdo con lo indicado a la pregunta anterior.
- En la clase **Tesoreria** debe implementar el método **actualizaboleta**, que en primer lugar se encarga de la lectura del archivo "escalas.csv" y cargar la información en el arreglo **lescala**. En segundo lugar, se encargará de calcular el valor de la boleta del alumno. Este valor bruto se obtendrá de multiplicar el valor del crédito con la cantidad de créditos que lleva el alumno. Luego se deben aplicar los cargos o descuentos respectivos. Para el caso de los alumnos virtuales el atributo **total** a considerarse en la clase derivada tiene un cargo de S/.100.00 por la licencia. Para el caso de los alumnos semipresenciales y presenciales, el cálculo de los descuentos y recargas se realiza sobre el monto bruto previamente determinado, y se registra en el atributo **total** de las clases derivadas. Finalmente se debe actualizar en el atributo **total** de la clase base **Alumno**, considerando el valor bruto más o menos el cargo o descuento correspondiente. Para esta tarea debe usar un método polimórfico denominado **actualizatotat**. **Es obligatorio para esta operación utilizar polimorfismo con el fin de acceder a los alumnos de acuerdo con el tipo, en caso contrario la respuesta no tiene valor.**
- Finalmente imprimir el siguiente reporte:

Codigo	Nombre	Escala	Cred.	Licencia	Total
202111260	GORDILLO/CASTRO/RONAL	2	19.50		8470.80
202111397	ZEVALLOS/PRADO/RICARDO	2	20.00		3620.00
202111935	GUZMAN/BARROS/FELIPE	3	22.00	1984F2-2	10092.40
202112598	MORI/ZAVALETA/LUZ-LOURDES	1	21.50		4855.56
202112887	CABRERA/AGUERO/JESSICA	2	21.00		3801.00
202113179	PEREZ/PEREZ/BIANCA/PATRICIA	3	21.00	6201G3-2	9638.20
202113223	DIAZ/SEDANO/YVONNE	4	19.50		7598.96
202113758	SOMA/INGA/DIALY	1	21.00		6521.13
...					

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

alumnos.csv
S,202123703,GAMARRA/TABORI/PAUL-RONAL,5,20,30
P,202119153,MENDOZA/ARIAS/HENRY,4,19.5,10
P,202113758,SOMA/INGA/DIALY,1,21,10
P,202123373,ESCALANTE/RARAZ/BLANCA,3,18,20
...

Tipo de alumno, código, nombre, escala, créditos, descuento/recarga/licencia

escala.csv
5,666.90
4,556.70
3,454.20 ...

Escala, costo del crédito.

Recuerde que si no usa polimorfismo la respuesta no será válida. Los métodos pueden ser recursivos o iterativos según lo desee.

Al finalizar la práctica, comprima la carpeta de su proyecto 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
Erasmus Gómez
Miguel Guanira

San Miguel, 30 de junio del 2023.