PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1
7ma práctica (tipo b)
Segundo 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 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 muestres 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.** Tampoco se podrá emplear la cláusula ***protected*** ni la cláusula ***friend***, de hacerlo se no se le calificarán las clases involucradas.
- 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_Lab07_2023_2"** 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 7 de los temas: "Herencia" y "Polimorfismo".

PARTE 01 (14 puntos): CREACIÓN DE LAS CLASES

Se solicita que desarrolle un proyecto **"PREG01_LAB07"** 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 y polimorfismo:

- **Para manejar los pedidos:** La clase se denominará "**Pedido**" y deberá contener lo siguiente: 1) un atributo denominado **codigo** (`char*`), 2) un atributo denominado **dni_cliente** (`int`), 3) un atributo denominado **subtotal** (`double`), 4) un atributo denominado **fecha** (`int`), 5) un atributo denominado **estado** (`char*`), 6) un atributo denominado **total** (`double`).
- **Para manejar a los pedidos de prioridad Alta:** La clase se denominará "**PrioridadAlta**" y deberá contener lo siguiente: 1) un atributo denominado **recargo** (`double`), este campo almacena el porcentaje de recargo que tiene el pedido sobre el precio final, 2) un atributo denominado **total** (`double`), este campo almacena el monto correspondiente al recargo que se aplicará al precio final. Además, esta clase posee datos heredados de la clase **Pedido.**
- **Para manejar a los pedidos de prioridad Media:** La clase se denominará "**PrioridadMedia**" y deberá contener lo siguiente: 1) un atributo denominado **descripcion** (`char*`), donde se almacena la razón del posible retraso del pedido. 2) un atributo denominado **nueva_fecha_entrega** (`int`), Además, esta clase posee datos heredados de la clase **Pedido.**
- **Para manejar a los pedidos de prioridad Baja:** La clase se denominará "**PrioridadBaja**" y deberá contener lo siguiente: 1) un atributo denominado **dias_espera** (`int`), 2) un atributo denominado **nueva_fecha_entrega** (`int`), este campo almacena la nueva fecha probable de atención del pedido, inicialmente en 0. Además, esta clase posee datos heredados de la clase **Pedido.**
- **Para manejar una orden de venta**: la clase se denominará "**OrdenVenta**" y deberá contener lo siguiente: 1) un atributo denominado **ptr_orden**, este atributo es un puntero de la clase **Pedido**, y servirá para registrar los atributos del pedido que representan, de acuerdo con el tipo que tiene asignado.
- **Para manejar todas las órdenes de venta**: La clase se denominará "**Almacen**" y deberá contener lo siguiente: 1) un atributo denominado **ordenes**, este atributo es un arreglo estático de la clase **OrdenVenta**, donde se guardarán todos los pedidos, 2) un atributo **cantidad_ordenes** (int)

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

Con las clases indicas debe realizar las siguientes operaciones:

- En la clase **Almacen** debe implementar el método **cargar_pedidos,** que se encarga de la lectura del archivo "pedidos.csv" y cargar los pedidos en el arreglo **ordenes**, de acuerdo con el tipo de pedido que indique el primer campo de cada fila del archivo leído. (A: Alta, M: Media, B: Baja). Los atributos **total** y **estado** de todas las clases debe considerarse en 0. Para este paso debe utilizar el método polimórfico **lee**.
- En la clase **Almacen** implementar el método **imprimir_ordenes_venta,** que se encargue de realizar la impresión de un archivo de prueba debidamente tabulado (**sin usar el carácter '\t'**), que muestre el contenido del arreglo **ordenes** correspondientes a los pedidos. 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 pedidos de acuerdo con su tipo.
- **imprime**: para la impresión de los datos de cada pedido de acuerdo con su tipo.

Consideraciones:

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

```
#include "Almacen.hpp"

using namespace std;

int main(int argc, char** argv) {
   Almacen almacen;

   almacen.cargar_pedidos();
   almacen.actualizar_pedidos();
   almacen.imprimir_ordenes_venta();

   return 0;
}
```

NO PUEDE CAMBIAR ESTE CÓDIGO

PARTE 2 (6 puntos): Proceso final.

Desarrolle un proyecto denominado **"PREG02_LAB07"** en el cual se utilizará obligatoriamente las clases desarrolladas en la pregunta anterior. El proyecto ejecutará las tareas descritas a continuación:

- Cargar el arreglo **ordenes** de acuerdo con lo indicado a la pregunta anterior.
- En la clase **Almacen** debe implementar el método **actualizar_orden**, que se encargará de calcular el valor total de la orden. Luego se deben aplicar los cargos o actualizaciones respectivas. Para el caso de los pedidos de prioridad Alta el atributo **total** a considerar en la clase derivada tiene un recargo porcentual adicional. Para el caso de los pedidos con prioridad Media, se actualizará el estado (0 Inactivo, 1 Activo) y la nueva fecha de entrega para el siguiente día. Para el caso de los pedidos con prioridad Baja, solo se actualizará la nueva fecha de entrega. Finalmente se debe actualizar en el atributo **total**, **estado** y **fecha** de la clase base **Pedido**, considerando las observaciones antes mostradas. Para esta tarea debe usar un método polimórfico denominado **actualiza**.
-

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

pedidos.csv

```
B,JXD-139,50375303,64.82,2023-12-01,5
M,CRU-009,50375303,49.38,2023-11-26,1
A,YYK-309,22777006,69.37,2023-11-11,7
…
```

Tipo de pedido, código de pedido, código de cliente, subtotal, fecha, recargo/estado/días en espera

Recuerde que si no usa polimorfismo y herencia la respuesta no será válida.

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
Erasmo Gómez
Miguel Guanira

San Miguel, 03 de noviembre del 2023.