

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

9na práctica (tipo b)
Segundo Semestre 2021

Indicaciones Generales:

- Duración: 110 minutos.

Obligatoriamente los alumnos deberán mantener en todo momento el AUDIO Y VIDEO de sus computadores abierto de modo que puedan recibir los comunicados que se hagan durante el examen y la revisión de los trabajos que estén desarrollando. De tener algún problema deberán hacérselo saber de inmediato al profesor de su horario por correo. De no hacerlo, no se aceptarán reclamos alegando que no oyeron las indicaciones.

- No se pueden emplear estructuras, variables globales ni la clase 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. Tampoco podrá hacer uso de plantillas
 - 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 (una línea de código no es una línea de texto, es una instrucción que termina con un punto y coma). 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 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.
 - NO SE CALIFICARÁN aquellas funciones desarrolladas en el mismo archivo que la función main.
 - El código comentado NO SE CALIFICARÁ y esto incluye el comentar el llamado a la función que lo contiene.
 - Los proyectos deben obligatoriamente desarrollarse en NetBeans bajo el sistema operativo Windows. No se revisarán los proyectos desarrollados en otros sistemas operativos o IDEs.
 - 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%.
 - Salvo en la sobrecarga de los operadores >> y <<, no se podrán definir funciones independientes que no estén ligadas como métodos a alguna de las clases planteadas.
 - SE DESCONTARÁN 2 PUNTOS EN LA NOTA FINAL DE LA EVALUACIÓN SI EN EL PROYECTO ENTREGADO SE ENCONTRASEN BIBLIOTECAS DE FUNCIONES QUE CONTENGAN FUNCIONES QUE NO TIENEN QUE VER CON EL PROBLEMA A RESOLVER
- 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. ESTO Y EL HECHO DE ENCONTRAR CUALQUIER ARCHIVO YA SEA .cpp O .h CON FECHA U HORA DE CREACIÓN ANTERIOR A LA EVALUACIÓN SERÁ CONSIDERADO UNA FALTA DE PROBIDAD Y POR LO TANTO AMERITARÁ LA ANULACIÓN DE LA PRUEBA.

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

- Puntaje total: 20 puntos.

Cuestionario:

La finalidad principal de este laboratorio es la de reforzar los conceptos contenidos en el capítulo 7 del tema: "Polimorfismo".

Cree una carpeta denominada "Lab09_Biblioteca2021-2", en esta carpeta colocará los proyectos solicitados en las preguntas 1 y 2 respectivamente. **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁ 3 PUNTOS DE LA NOTA FINAL.**

PARTE01 (15 puntos): CREACIÓN DE LAS CLASES

Se solicita que desarrolle un proyecto "PREG01_LAB09" 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 de la Biblioteca: La clase se denominará "Libro" y deberá contener lo siguiente: 1) un atributo denominado **codigo** (**char***) definido por una cadena de caracteres, 2) un atributo denominado **titulo** (**char***) definido por una cadena de caracteres, 3) un atributo denominado **autor**(**char***) definido por una cadena de caracteres, 4) un atributo denominado **cantidad** (**int**) con la cantidad de libros que posee la biblioteca, 5) un atributo denominado **anho** (**int**) con el año del libro, 6) un atributo denominado **precio** (**double**), con el valor del libro.
- Para manejar las copias de los libros: la clase se denominará "Copia" y deberá contener lo siguiente: 1) un atributo denominado **numero** (**int**) con el número de copia del libro, 2) un atributo denominado **estado** (**char***) definido por una cadena de caracteres que indicará el estado del libro ("Disponible", "Prestamo" o "Perdida"), 3) un atributo denominado **Olibro**, este atributo es un puntero de la clase **Libro**,
- Para manejar los libros disponibles: La clase se llamará "Disponible" y deberá contener lo siguiente: 1) un atributo denominado **zona** (**int**) que almacenará la zona donde está almacenado el libro disponible. Esta clase posee datos heredados de la clase **Libro**.
- Para manejar los libros prestados: La clase se llamará "Prestamo" y deberá contener lo siguiente: 1) un atributo denominado **dias** (**int**) que almacenará la cantidad de días que está en préstamo un libro desde la fecha ingresada, 2) un atributo denominado **fecha** (**int**) que almacena la fecha en que el libro fue prestado al usuario (**aaaammdd**). Esta clase posee datos heredados de la clase **Libro**.
- Para manejar los libros perdidos: La clase se llamará "Perdida" y deberá contener lo siguiente: 1) un atributo denominado **precio** (**double**) que almacena el precio del libro perdido. Esta clase posee datos heredados de la clase **Libro**.
- Para manejar la biblioteca: La clase se denominará "Biblioteca" y deberá contener lo siguiente: 1) un atributo denominado **llibros**, este atributo es un arreglo estático de la clase **Copia**, donde se guardarán las copias de todos los libros que tiene la biblioteca.

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

Con las clases indicas debe realizar las siguientes operaciones:

- En la clase **Biblioteca** debe implementar el método **GeneraCopias**, que se encarga de la lectura del archivo "libros2.csv" y cargar la información en el arreglo **llibros**, considerando una posición en el arreglo por cada copia que posee un libro, por ejemplo, si un libro tiene 10 copias deberá almacenarse en el arreglo las copias numeradas de 1 a 10, además de los datos propios del libro para cada copia, a continuación, debe considerarse las copias del siguiente libro. Inicialmente todas las copias tienen el atributo **Olibro** en estado disponible.
- En la clase **Biblioteca** debe implementar el método **ActualizaLibros**, que se encarga de la lectura del archivo "RegistroDePrestamos2.csv" y de acuerdo con una fecha ingresada por el usuario determinará el número de días que la copia el libro está en préstamo (puede probar con la fecha 20211115). Para este cálculo debe usar la biblioteca **FuncionesFecha**. Si la copia tiene menos de 60 días se considera que está en préstamo, por tal motivo debe actualizar el atributo **Olibro**, debido a que el libro ya no está disponible. Si la copia tiene igual o más de 60 días se considera como perdida, por tal motivo debe actualizar el atributo **Olibro**. Para esta operación se debe utilizar al método polimórfico **Actualiza**. Recuerde que el proceso de actualización debe liberar de forma apropiada la memoria correspondiente al estado anterior.
- Finalmente, en la clase **Biblioteca** implementar el método **imprimeCopias**, que se encargue de realizar la impresión de un **archivo de prueba debidamente tabulado (sin usar el carácter '\t')**,

que muestre cada libro con sus respectivas copias, además de los datos de sus atributos correspondientes. Para esta operación debe utilizar el método polimórfico **Imprimir**.

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

- **Actualiza**: para la actualización de los estados de las copias de los libros.
- **Imprimir**: para la impresión de los datos de cada una de las copias según su estado.

Consideraciones:

Debido a que se leen y almacenan todos los libros existentes, es posible que un libro se repita varias veces. Para el desarrollo de ambas preguntas debe considerar el siguiente código:

```
#include "Biblioteca.h"

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

    bib.GeneraCopias();
    bib.ActualizaLibros();
    bib.ImprimeCopias();

    return 0;
}
```

**NO PUEDE
CAMBIAR
ESTE CÓDIGO**

PARTE 2(5 puntos): Prueba final.

Desarrolle un proyecto denominado **"PREG02_LAB09"** en el cual se utilizará obligatoriamente las clases desarrolladas en la pregunta anterior. El proyecto ejecutará las tareas descritas a continuación:

- Cargar el arreglo **libros** de acuerdo con lo indicado a la pregunta anterior.
- Actualizar la información de las copias de los libros utilizando los métodos polimórficos.
- Finalmente imprimir el siguiente reporte:

```
Codigo:KC34197
Titulo:Las_demoliciones_de_Paris
Autor: Zkenia/Chung/Zuniga
Cantidad: 15
-----
1 Perdida          Precio: 294.26
2 Disponible       Zona:      3
3 Disponible       Zona:      3
4 Disponible       Zona:      3
5 Disponible       Zona:      3
6 Prestamo Dias:   54
7 Prestamo Dias:   58
...
-----
Codigo:FY25192
Titulo:La_hermosa_jardinera
Autor: Humberto/Amezquita/Contreras
Cantidad: 15
...
```

Se recomienda revisar los archivos que servirán para la lectura de datos. **Recuerde que si no usa polimorfismo 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**. Luego súbalo a la tarea programa en Paideia para este laboratorio.

Profesores del curso: Miguel Guanira
Rony Cueva

San Miguel, 26 de noviembre del 2021.