

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

7ma 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 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 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_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:

La finalidad principal de este laboratorio es la de reforzar los conceptos contenidos en los capítulo 7, Herencia.

PARTE01 (20 puntos): CREACIÓN DE LAS CLASES

Se solicita que desarrolle un proyecto **"LAB07_BIBLIOTECA_HERENCIA"** dentro de la carpeta correspondiente, **DE NO COLOCAR ESTE REQUERIMIENTO SE LE DESCONTARÁN 3 PUNTOS DE LA**

NOTA FINAL, en la cual se definirán clases para el manejo de los datos de una Biblioteca, sus los libros y sus respectivos estantes. A continuación, se definen las clases que serán necesarias:

- **Para manejar los libros:** La clase se denominará "**Libro**" y deberá contener lo siguiente: 1) un atributo denominado **codigo** (**char***) 2) un atributo denominado **nombre** (**char***), 3) un atributo denominado **ancho** (**int**), 4) un atributo denominado **alto**(**int**), 5) un atributo denominado **peso**(**double**), 5) un atributo denominado **colocado**(**bool**).
- **Para manejar las enciclopedias:** La clase se denominará "**Enciclopedia**" y deberá contener lo siguiente: 1) un atributo denominado **volumen** (**int**) 2) un atributo denominado **peso** (**double**). Además, esta clase posee datos heredados de la clase **Libro**.
- **Para manejar las novelas:** La clase se denominará "**Novela**" y deberá contener lo siguiente: 1) un atributo denominado **tipo** (**char***), 2) un atributo denominado **peso** (**double**). Además, esta clase posee datos heredados de la clase **Libro**.
- **Para manejar los espacios en los estantes:** La clase se denominará "**Espacio**" y deberá contener lo siguiente: 1) un atributo **contenido** (**char**), 2) un atributo denominado **posx** (**int**) 3) un atributo denominado **posy** (**int**). El atributo contenido es un carácter que representa si el espacio está vacío u ocupado.
- **Para manejar los estantes:** La clase se denominará "**Estante**" y deberá contener lo siguiente: 1) un atributo denominado **codigo** (**char***), 2) un atributo denominado **anchura** (**int**), 3) un atributo denominado **altura** (**int**), 4) un atributo denominado **espacios**, definido por un **arreglo dinámico de una sola dimensión** de la clase **Espacio**, 5) un atributo denominado **pesoSoportado** (**double**), 6) un atributo denominado **pesoActual** (**double**). El atributo espacios, es la representación física de un estante, es decir un arreglo de dos dimensiones **representado por una sola dimensión**. Ver Figura 1.
- **Para manejar los estantes verticales:** La clase se denominará "**EstanteVertical**" y deberá contener lo siguiente: 1) un atributo denominado **enciclopedias** definido por un arreglo estático de la clase **Enciclopedia**, 2) un atributo denominado **cantidad_enciclopedias**(**int**)
- **Para manejar los estantes horizontales:** La clase se denominará "**EstanteHorizontal**" y deberá contener lo siguiente: 1) un atributo denominado **novelas** definido por un arreglo estático de la clase **Novela**, 2) un atributo denominado **cantidad_novelas**(**int**)
- **Para manejar la biblioteca:** La clase se denominará "**Biblioteca**" y deberá contener lo siguiente: 1) un atributo denominado **estantesVerticales**, definido por un arreglo estático de la clase **EstanteVertical**, 2) un atributo denominado **cantidad_estantes_verticales** (**int**), 3) un atributo denominado **estantesHorizontales**, definido por un arreglo estático de la clase **EstanteHorizontal**, 4) un atributo denominado **cantidad_estantes_horizontales** (**int**), 5) un atributo denominado **enciclopedias**, definido por un arreglo estático de la clase **Enciclopedia**, 6) un atributo denominado **cantidad_total_enciclopedias** (**int**). 7) un atributo denominado **novelas**, definido por un arreglo estático de la clase **Novela**, 8) un atributo denominado **cantidad_total_novelas** (**int**).

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

Con las clases indicadas debe realizar las siguientes operaciones:

- En la clase **Biblioteca** debe implementar el método **cargar_libros**, que se encarga de la lectura del archivo "libros.csv" y cargar los libros en los arreglos **enciclopedias** o **novelas**, de acuerdo con el tipo de pedido que indique el primer campo de cada fila del archivo leído. (E: Enciclopedia, N: Novela).
- En la clase **Biblioteca** debe implementar el método **cargar_estantes**, que se encarga de la lectura del archivo "estantes.csv" y cargar los estantes en los arreglos **estantesVerticales** o **EstantesHorizontales**, de acuerdo con el tipo de estante que indique el primer campo de cada fila del archivo leído. (H: Horizontal, V: Vertical).

- En la clase Biblioteca implementar el método **posicionar_libros**, que se encarga de insertar física y digitalmente los libros en sus respectivos estantes, siempre y cuando cumplan con las siguientes condiciones:
 - Para verificar que un libro pueda ser introducido en un estante, debe verificar si el ancho y altura del libro es suficiente para soportar los espacios **SOBRANTES** del estante. Se recomienda crear un método para conseguir los espacios restantes del estante.
 - Una enciclopedia **SOLO** puede ser introducida en un **EstanteVertical** y una novela **SOLO** puede ser introducida en un **EstanteHorizontal**. Cada inserción actualizará los valores de **pesoActual** en la clase **Estante**.
 - Cuando un libro es introducido a un estante deberá no solo actualizar la información de cantidad de libros en el estante y su estado, si no también actualizar el estante **FÍSICAMENTE**. Deberá marcar los **espacios** con el **contenido** que corresponda.
 - En la figura 1 se muestra la representación física de un estante horizontal 6x4 (anchura x altura), la **novela roja** (1x3) se insertó correctamente ya que entra dentro de las dimensiones del **estante horizontal**, lo mismo para la **novela verde** (2x2) y la **novela azul** (1x4). Cada libro debe ser ubicado en los espacios contenidos en algún estante de ser posible. En la figura 2 se muestra la representación física del **estante vertical** (4x5), la **enciclopedia roja** (2x4) se inserta de manera horizontal, luego la **enciclopedia verde** (2x3) se inserta ya que aún sobran espacios verticales.
- En la clase **Biblioteca**, implementar el método **mostrar_datos**, que se encargue de realizar la impresión de un archivo de prueba debidamente tabulado (**sin usar el carácter '\t'**), que muestre el contenido de los arreglos de **estantes** correspondientes a los libros posicionados.

Figura 1. Representación Física de un Estante Horizontal

| | | | | | |
|--|--|--|--|--|--|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

Figura 2. Representación Física de un Estante Vertical

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

De acuerdo con lo solicitado, la función "main" del proyecto estará compuesto por el siguiente código:

```
#include "Biblioteca.hpp"
using namespace std;
int main(int argc, char** argv) {
    Biblioteca biblioteca;
    biblioteca.cargar_libros();
    biblioteca.cargar_estantes();
    biblioteca.posicionar_libros();
    biblioteca.mostrar_datos();

    return 0;
}
```

NO PUEDE CAMBIAR ESTE
CÓDIGO

El reporte debe lucir de la siguiente manera:

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

| libros2.csv |
|--|
| N, ABC123, El libro de la Selva, 2, 5, Aventura, 0.75 E, EFG123, La enciclopedia de la Vida, 2, 4, 1, 6.30 ... |

Tipo de libro, código de libro, descripción de libro, ancho, alto, tipo/volumen, peso

| estantes2.csv |
|-------------------------------------|
| H, AAA, 8, 5 V, BBB, 4, 6 ... |

Tipo de libro, código de libro, descripción de libro, ancho, alto, tipo/volumen, peso

```

=====
Informacion del posicionamiento de Libros
en los estantes de la Biblioteca
=====
Cantidad de Estantes: 3
-----
Codigo Estante: AAA          Cantidad de Libros: 3
Anchura X Anchura : 6 x 4   Peso Máximo: 3.00 Kg
Tipo: Horizontal            Peso Total: 2.50 Kg
-----

[  ][  ][  ][ * ][  ][  ]
[ * ][  ][  ][ * ][  ][  ]
[ * ][ * ][ * ][ * ][  ][  ]
[ * ][ * ][ * ][ * ][  ][  ]
-----

CODIGO      NOMBRE                                ANCHO  ALTO  PESO
-----
ABC123      El libro de la Selva                        1      3    0.75
ABC321      El libro de la Costa                          2      2    1.00
CBA123      El libro de la Sierra                        1      4    0.75
-----
Codigo Estante: BBB          Cantidad de Libros: 2
Anchura X Anchura : 4 x 5   Peso Máximo: 10.00 Kg
Tipo: Vertical            Peso Total: 8.30 Kg
...

=====
Informacion de todos los Libros:
Cantidad de Libros Total: 10
ABC123      El libro de la Selva                        1      3    0.75
...
CCC123      NO SE PUDO COLOCAR                          11     5    2.80
=====

```

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, 31 de mayo del 2024.