PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1
6ta 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, iomanip y fstream). **NO PUEDE UTILIZAR LA CLASE string**. Tampoco se podrán emplear las funciones de C que gesten 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 muestres 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_Lab06_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 capítulos 6 y 7 del tema: "Programación orientada a objetos" y "Operadores Sobrecargados".

PARTE01 (14 puntos): CREACIÓN DE LAS CLASES

Se solicita que desarrolle un proyecto **"LAB06_PREG01_CLASES"** 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 libros y sus estantes. Así

como una serie de operadores sobrecargados que permitan manejar estas clases. 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 **colocado**(`bool`).

- **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 o 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 **libros** definido por un arreglo estático de la clase Libro, 4) un atributo denominado **espacios**, definido por un **arreglo dinámico de una sola dimensión** de la clase `Espacio`, 5) un atributo denominado **cantidad_libros**(`int`). 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 la biblioteca:** La clase se denominará "**Biblioteca**" y deberá contener lo siguiente: 1) un atributo denominado **estantes**, definido por un arreglo estático de la clase `Estante`, 2) un atributo denominado **cantidad_estantes** (`int`), 3) un atributo denominado **libros**, definido por un arreglo estático de la clase `Libro`, 4) un atributo denominado **cantidad_libros** (`int`).

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

Las operaciones que se permitirá realizar a través de sobrecargas de operadores se definen a continuación:

- **Lectura**:
  - Sobrecargando el operador >> de modo que permita leer un libro de un archivo de CSV. La operación (**arch >> libro;**) involucraría un archivo de csv ("*libros.csv*") y una variable de tipo "class Libro".

```
ABC123,EL LIBRO DE LA SELVA,2,5
(CÓDIGO, NOMBRE, ANCHO y ALTO del libro)
El resto de campos deben inicializarse adecuadamente
```

  - Sobrecargando el operador >> de modo que permita leer un estante de un archivo de CSV. La operación **(arch >> estante;)** involucraría un archivo csv ("*estantes.csv*") y una variable de tipo "`class Estante`". Una línea de archivo tendrá la siguiente forma:

```
AAA123,10,5
(CÓDIGO, ALTURA y ANCHURA del estante)
El resto de campos deben inicializarse adecuadamente
```

  Estas sobrecargas deben ser definidas como parte del archivo de la clase que manejan.

- **Agregación:**
  - Sobrecargando el operador += de modo que permita agregar un libro a un estante. Para usar la operación (**estante** += **libro**;) deberá verificar que se pueda agregar dicho libro al **espacio físico del estante**. La sobrecarga deberá devolver un valor de tipo `bool`, **true** si el libro se pudo agregar al estante y **false** si no se puede. De ser el caso debe incrementar la cantidad de libros en el estante y el valor **colocado** del libro.
    - 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.
    - 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 4x6 (anchura x altura), el libro rojo (1x3) se insertó correctamente ya que entra dentro de las dimensiones del **estante** , lo mismo para el libro verde (2x2) y el azul (1x4). Cada libro debe ser ubicado en los espacios contenidos en algún estante de ser posible.

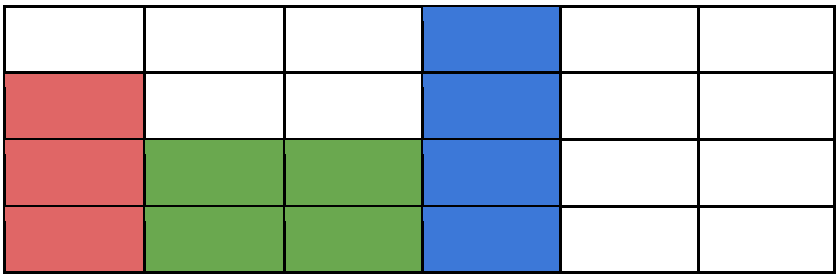

Figura 1. Representación Física de un Estante

- **Impresión:**

  - Sobrecargando el operador << de modo que permita imprimir la información de un espacio.
    La operación (**arch << espacio**) permitirá imprimir en un archivo de textos el contenido como un "[*]" , de estar ocupado o un espacio vacío "[ ]", de estar libre.
  - Sobrecargando el operador << de modo que permita imprimir la información de **un** libro.
    La operación (**arch << libro;**) permitirá imprimir en un archivo de textos los datos contenidos en una variable de tipo **"class Cliente"**. El formato será el siguiente:

```
ABC123  LIBRO DE LA SELVA  1  3
```

Código, Nombre de Libro, Ancho y Altura

  - Sobrecargando el operador << de modo que permita imprimir la información de **un** producto.
    La operación (**arch << estante;**) permitirá imprimir en un archivo de textos los datos contenidos en una variable de tipo **"class Estante"**. El formato será el siguiente:

```
Codigo Estante: AAA       Cantidad de Libros: 3
Anchura del Estante: 6    Altura del Estante: 4
 ------------------------------------------------

[   ][   ][   ][ * ][   ][   ]
[ * ][   ][   ][ * ][   ][   ]
[ * ][ * ][ * ][ * ][   ][   ]
[ * ][ * ][ * ][ * ][   ][   ]

CODIGO     NOMBRE                      ANCHO   ALTO
 ..................................................
ABC123     El libro de la Selva          1       3
ABC321     El libro de la Costa          2       2
CBA123     El libro de la Sierra         1       4
```

Consideraciones:
La solución debe contemplar la elaboración del proyecto de implementación, y la prueba de las sobrecargas en el main, no hay problema si para esta labor se excede en el número de líneas. Las pruebas de las sobrecargas deben ser realizadas lo más simple posible, pero que se muestre claramente que son correctas.

PARTE 2(6 puntos): Prueba final.
Desarrolle un proyecto denominado **"LAB06_PREG02_BIBLIOTECA" en el cual se utilizará obligatoriamente las clases desarrolladas en la pregunta anterior**. El proyecto ejecutará las tareas descritas a continuación utilizando las sobrecargas definidas anteriormente para popular los datos de un almacén:

a) Leer los datos de los libros contenidos en un archivo CSV como se muestra a continuación y los coloque en **libros**, atributo de la clase Biblioteca:

b) Leer los datos de los productos contenidos en un archivo CSV como se muestra a continuación y los coloque en **estantes**, atributo de la clase Biblioteca:

c) Ubique los **libros** en los **estantes** utilizando las sobrecargas realizadas en la pregunta anterior.

d) Emitir un reporte con la información completa de la Biblioteca. El reporte debe tener un título y subtítulos adecuados.

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:

```
================================================
    Informacion del posicionamiento de Libros
       en los estantes de la Biblioteca
================================================
Cantidad de Estantes: 2
------------------------------------------------
Codigo Estante: AAA       Cantidad de Libros: 3
Anchura del Estante: 6    Altura del Estante: 4
 ------------------------------------------------

[   ][   ][   ][ * ][   ][   ]
[ * ][   ][   ][ * ][   ][   ]
[ * ][ * ][ * ][ * ][   ][   ]
[ * ][ * ][ * ][ * ][   ][   ]

CODIGO     NOMBRE                      ANCHO   ALTO
 ..............................................
ABC123     El libro de la Selva          1       3
ABC321     El libro de la Costa          2       2
CBA123     El libro de la Sierra         1       4
 ------------------------------------------------
Codigo Estante: AAA       Cantidad de Libros: 3
Anchura del Estante: 6    Altura del Estante: 4
…

================================================
Informacion de todos los Libros:
Cantidad de Libros Total: 5
ABC123     El libro de la Selva          1       3
…
CCC123     NO SE PUDO COLOCAR            11      5
================================================
```

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
Erasmo Gómez
Heider Sánchez

San Miguel, 24 de mayo del 2024.