



Programación Orientada a Objetos

Academia Main()

Ejercicio 1 (2 puntos)

Implementar la clase Pieza en los ficheros pieza.cc y pieza.h. Una pieza tiene una cadena nombre, un entero código, entero cantidad y double precio.

- Constructor vacío con los valores por defecto nombre("Sin Nombre") y el resto a cero.
- Constructor parametrizado cuyos únicos parámetros obligatorios son el código y el nombre (en ese orden) el resto son opcionales tomando como valores por defecto cero.
- Constructor de copia.
- Observadores y modificadores.
Nota: el observador de la variable nombre se debe implementar por referencia.
- Método getPieza. Dicho método retorna una cadena, por ejemplo:
"Hay X unidades de la pieza Y."
Siendo X la cantidad e Y el nombre de la pieza

Ejercicio 2 (1 puntos)

Codificar la clase Refpieza, en los ficheros, refpieza.h y refpieza.cc, que deriva de la clase Pieza. Esta clase tiene además un código de referencia de tipo string.

- Constructor parametrizado cuyos únicos parámetros, todos obligatorios, son el código de referencia, el código de pieza, y el nombre de pieza, en dicho orden.
- Constructor de copia.
- Observadores y modificadores.

Ejercicio 3 (3 puntos)

Implementar la clase CPieza que dispone de una lista de objetos de tipo Pieza.

- Constructor vacío. Debe asegurarse que la lista esta vacía.
- Constructor de copia.
- Observador get() que retorna la lista.
- Método size() de la clase CPieza, que retorna el numero de piezas de la lista.
- Método addPieza. Recibe un objeto Pieza y lo inserta en la lista si este no existe ya. Para buscar la pieza se utiliza el campo código. Retorna true si la pieza se ha insertado correctamente y false en caso contrario.
- Método deletePieza. Recibe un objeto Pieza y lo elimina de la lista. Si se ha eliminado el método retorna true y false en caso contrario. Para buscar la pieza se utiliza el campo código, como en el apartado anterior.
- Método write(), el cual recibe un string que será el nombre de un fichero, y escribe el contenido de la lista en dicho fichero con la siguiente estructura:

*codigo nombre
cantidad precio*

- Método read(), que recibe un string que representa el nombre de un fichero y carga las piezas leidas en la lista. Los datos se encuentran respetando la estructura anterior.



Ejercicio 4 (2 puntos)

Copie los ficheros del ejercicio anterior con el nombre cpieza2.cc y cpieza2.h.

- Operador[]]. Este recibe un entero y retorna la pieza que se encuentra en la posición de la lista que dice el entero.
- Operador=. Efectúa la asignación de un objeto CPieza en otro.

Ejercicio 5 (2 puntos)

Codifica la sobrecarga de la función intercambia() de forma que intercambie el valor de dos variables recibidas como parámetros. Codifica una versión que intercambie dos variables de tipo int, otra que intercambie dos variables de tipo float y otra dos variables de tipo string.
Escribe el código en el fichero intercambia.cc

Ejercicio 6 (2 puntos)

Codifica la clase Usuario con un campo “id” de tipo int. La clase Usuario además tendrá una variable estática “n” de tipo int inicializada a 0 que se encarga de informar del número de usuarios creados en el sistema.
Codifica esta clase en los ficheros usuario.h y usuario.cc con la siguiente funcionalidad:

- El constructor de la clase recibe como parámetro el id del usuario e incrementa el valor de la variable estática “n” en una unidad.
- Observador getId().
- Observador getN() que devuelve el valor de la variable estática n.