Parcial Nro 1 - Tarde

Programación Orientada a Objetos - Parcial 1 - 28/09/2021 - Tema B

Ejercicio 1 (30 pts)

a) Escriba una función que reciba tres punteros indicando el comienzo, una posición intermedia y el final de un arreglo. La función debe generar y retornar dos nuevos arreglos dinámicos: uno con los elementos que en el arreglo de entrada estaban antes de la posición intermedia, y otro con los que estaban después (notar que el elemento apuntado por ese puntero intermedio no formará parte de ninguno de los dos). Como casos especiales, si se recibe como posición intermedia a la del primer o del último elemento la función deberá generar y retornar un solo arreglo (el otro estaría vacío, así que mejor no generarlo en ese caso).

b) Escriba un programa cliente para probar la función que permita al usuario ingresar un arreglo y un número de posición para generar el 2do puntero; y luego muestre los resultados.

Nota: no puede usar std::vector para este ejercicio.

Ejercicio 2 (35 pts)

a) Defina una clase Asignatura que reciba en su constructor el nombre de una materia y el número de cuatrimestre al que pertenece. La clase debe tener métodos para consultar estos datos y para gestionar una lista de "unidades" (strings).

b) Implemente, reutilizando la clase Asignatura, las clases FuPro y CompGraf para representar las materias “Fundamentos de Programación” y “Computación Gráfica” respectivamente. Estas clases deben tener un método CalcularCondicion que reciba un vector de elementos de tipo struct Evaluacion { string tipo; int nota; } con los resultados de un alumno. El método debe verificar que los tipos de evaluaciones sean los correctos (si no lo son, deberá retornar "error"), y calcular su condición final en base a sus notas la condición final del alumno.

- Para FuPro, las evaluaciones deben ser de tipos "parcial1" y "parcial2"; y la condición final será "promocionado" si el promedio llega a 70 y la nota mínima no es menor de 60, "regular" si la nota mínima no baja de 40, o "libre" en caso contrario.

- Para CompGraf, las evaluaciones deben ser de tipos "tp1" y "tp2", "tpFinal", y "parcial". La condición final será "promocionado" si todos los tps están aprobados (nota 60 o superior) y el parcial tiene 65 o más; "regular" si los tps están aprobados pero el parcial no, o "libre" si algún tp está desaprobado.

c) Escriba una función “testCalcularCondicion” para probar el método CalcularCondicion. La función debe poder recibir cualquier Materia, permitir al usuario ingresar un vector de Evaluaciones y mostrar la condición resultante.

Ejercicio 3 (35 pts)

Se desea gestionar la venta de entradas de un teatro. Una sala de teatro está compuesta por butacas y cada sala puede tener diferente número de butacas.

a) Cada butaca tiene un número, un precio y un estado (si está libre u ocupada). Codifique una clase Butaca que tenga un constructor para cargarle el precio, y métodos para modificar o consultar su estado, entre otras funcionalidades que considere necesarias.

b) Defina una clase Sala que guarde el nombre de la sala y las lista de butacas. La misma debera tener:

- Un constructor que reciba el nombre de la sala
- Un método que permita inicializar la lista de butacas
- Un método para registrar la compra de una entrada recibiendo el número de butaca. Si esa butaca está ocupada, a partir de la misma se buscará la siguiente libre del mismo precio. El método debe retornar el número de butaca asignado, o -1 si no encuentra ninguna libre.
- Un método para obtener la recaudación total de una función
- Un método para obtener el porcentaje de ocupación de la sala

c) Escriba un programa cliente para crear una sala y utilizar sus funcionalidades. La sala del ejemplo debe tener 100 plateas centrales (butacas nros 1 a 100, con un costo de $600), 200 laterales (nros 101 a 300, con un costo de $450), y 200 más en un 1er piso (nros 301 a 501, con un costo de $300). El programa debe permitir al usuario registrar N ventas, y luego mostrar la recaudación y el porcentaje de ocupación.