

Ejercicios

Ejercicios de programación funcional

1. Implementar una función que calcule el producto de dos matrices de enteros.

Claves: bucles for

2. Escribir un programa que nos de los tamaños de los tipos elementales del C++.

Claves: Operador sizeof.

3. Realizar una función que cuente el número de ocurrencias en una cadena de las cinco vocales en 5 variables distintas.

Claves: Manejo de cadenas y bucles.

4. Implementar una función que compare dos cadenas.

Claves: Manejo de cadenas y bucles.

5. Definir las estructuras necesarias para implementar una lista doblemente enlazada de enteros.

Claves: Estructuras y punteros.

6. Implementar una función que calcule el factorial de forma recursiva.

Claves: Funciones recursivas.

7. Definir las estructuras necesarias para representar una pila de enteros y definir las funciones pop(), push(), top() y is_empty().

Claves: Estructuras, punteros, funciones y TAD's.

8. Implementar las operaciones de suma, resta, producto y módulo para números complejos. Definir una estructura adecuada para su representación.

Claves: Estructuras, funciones y TAD's.

9. Realizar un programa que acepte dos enteros como parámetros desde la línea de comandos y nos saque por pantalla su suma, resta, producto y división.

Claves: La función main, conversiones de cadenas a enteros.

10. Implementar una macro que nos de el valor absoluto de un número.

Clave: Uso de macros

Ejercicio de programación orientada a objetos

Planteamiento general

Programa de gestión de un almacén, debemos crear una base de datos que contenga la información necesaria para saber que cosas tenemos almacenadas. Lo haremos definiendo unas fichas para cada producto en las que tendremos el nombre, la marca, el modelo, el número de unidades almacenadas y el precio de por unidad.

Con nuestro programa podremos crear, consultar, modificar y eliminar fichas de los objetos almacenados.

Además deberemos poder almacenar y recuperar las fichas del almacén en y desde fichero.

Aspectos de diseño

Debemos separar la gestión de la base de datos de la interface con el usuario. Definiremos unos mecanismos para gestión de la base de datos en memoria y en fichero y por otro lado la gestión de la E/S de usuario.

Clases a definir

Gestión de la BD en memoria:

Definiremos la clase de las fichas con sus métodos y atributos.

Definiremos una clase almacén que empleará algún tipo de clase contenedor para almacenar las fichas. El almacén debe proporcionar los métodos de insertar, borrar y obtener las fichas (buscando por claves, necesitaremos algún tipo de gestión de claves, por ejemplo definiendo una clase clave amiga de almacén y de ficha). La consulta y modificación de las fichas se hace con los métodos de estas.

Gestión de la BD en fichero:

Usaremos la E/S estándar del C++, incluyendo métodos en la clase almacén para guardar o recuperar una BD almacenada en fichero.

Gestión E/S usuario:

Definiremos una clase menú sobre la que se podrán inicializar las cadenas de descripción del menú y los códigos de cada opción, un menú se imprime y se pide una opción, nos retorna un código y no hace nada más.

La Salida de las fichas la pueden incorporar ellas mismas, pero no deben usar cout, sino un ostream. Si cambiamos la salida, sabemos que el ostream se puede manipular para que por pantalla salga su contenido como queramos.