

LISTA DE EJERCICIOS

1. Crear una clase llamada Hora que tenga como datos de tipo int para horas, minutos y segundos. Un constructor inicializará esos datos a 0, y otro lo inicializará a valores dados por usuario. Un método de instancia deberá visualizar la hora en formato HH:MM:SS. Otro método de instancia sumará dos objetos de tipo Hora pasados como argumentos. Por último, visualizar el valor resultante. Realizar la aplicación la cual crea dos objetos inicializados y uno que no está inicializado. Sumar los dos valores inicializados y dejar el resultado en el objeto no inicializado.
2. Crear una clase llamada Empleado que contenga como datos el nombre y el número de empleado, y como métodos de instancia leerDatos() y verDatos() que lean los datos del teclado y los visualice en pantalla, respectivamente.
Escribir un programa que utilice la clase, creando un arreglo de tipo Empleado y luego llenándolo con datos correspondientes a 50 empleados. Una vez llenado el arreglo, visualizar los datos de todos los empleados.
3. Diseñar una clase Vector3D que permita manipular vectores de 3 componentes (de tipo float) y que contenga un constructor con valores por defecto (0) y los métodos de instancia suma (suma de dos vectores), productoEscalar (producto escalar de dos vectores-: $v1=x1, y1, z1, v2=x2, y2, z2; v1*v2=x1*x2+y1*y2+z1*z2$).
4. Realizar una clase Complejo que permita la gestión de números complejos. Las operaciones a implementar son las siguientes.
 - Un método establecer() permite inicializar un objeto de tipo Complejo a partir de dos componentes double.
 - Un método imprimir() realiza la visualización formateada de un Complejo.
 - Dos métodos agregar() (sobrecargadas) que permiten añadir uno o dos componentes double a un Complejo.
5. Implementar una clase Hora. Cada objeto de esta clase representa una hora específica de un día, almacenando las horas, minutos y segundos como enteros. Incluir un constructor, métodos de acceso, un método avanzar() para avanzar (adelantar) la hora actual de un objeto existente, un método ponerACero para poner a cero la hora actual de un objeto existente y un método visualizar().
6. Implementar una clase Cadena. Cada objeto de esta clase representa una cadena de caracteres. Los datos son: la longitud de la cadena y la cadena de caracteres actual. Además, se deben añadir constructores, métodos de acceso y de visualizar, así como incluir un método carácter(int i) que devuelva el carácter de la cadena representado por el parámetro i que representa el índice o lugar del carácter en la cadena o -1 en caso de que i sea mayor que la longitud de la cadena.

7. Diseñar una jerarquía de clases que utilice como clase base o raíz una clase LAN.
Las subclases derivadas deben representar diferentes topologías, como estrella, anillo, bus y hub. Los datos deben representar propiedades tales como soporte de transmisión, control de acceso, formato del marco de datos, estándares, velocidad de transmisión, etc. Se desea simular la actividad de los nodos de tal LAN. La red consta de nodos, que pueden ser dispositivos tales como computadoras personales, estaciones de trabajo, máquinas FAX, etc. Una tarea principal de LAN es soportar comunicaciones de datos entre nodos. El usuario del proceso de simulación debe, como mínimo, poder:
- Enumerar los nodos actuales de la red LAN.
 - Añadir un nuevo nodo a la red LAN.
 - Quitar un nodo de la red LAN.
 - Configurar la red, proporcionándole una topología de estrella o bus.
 - Especificar el tamaño del paquete, que es el tamaño en bytes del mensaje que va de un nodo a otro.
 - Enviar un paquete de un nodo especificado a otro.
 - Difundir el paquete desde un nodo a todos los demás de la red.
 - Realizar estadísticas de la LAN, tales como tiempo medio que emplea un paquete.
8. Implementar una clase Automóvil de una jerarquía de herencia. Considere que, además de ser un Vehículo, un medio de transporte, etc. Automóvil debe tener al menos una herencia de tres clases y al menos tres clases derivadas.
9. Escribir una clase FigGeometrica que represente figura geométricas tales como punto, línea, rectángulo, triángulo y similares. Debe proporcionar métodos que permitan dibujar, ampliar, mover y destruir tales objetos. La jerarquía debe constar al menos de una ocho de clases.
10. Diseñar la siguiente jerarquía de clases:

Persona			
nombre			
edad			
visualizar()			
Estudiante		Profesor	
nombre	heredado	nombre	heredado
edad	heredado	edad	heredado
id	definido	salario	definido
visualizar()	redefinido	visualizar	heredada

Escribir un programa que manipule la jerarquía de clases, lea un objeto de cada clase y lo visualice.

- a) Sin utilizar funciones abstractas.
- b) Utilizando funciones abstractas.

11. Implementar una jerarquía de herencia de animales tal que contenga al menos cuatro niveles de derivación y diez clases.
12. Implementar una jerarquía Empleado de cualquier tipo de empresa que le sea familiar. La jerarquía debe tener al menos tres niveles, con herencia de dato, y métodos. Los métodos deben poder calcular salarios, despidos, promoción, dar de alta, jubilación, etc. Los métodos deben permitir también calcular aumentos salariales y primas para Empleados de acuerdo con su categoría y productividad. La jerarquía de herencia debe poder ser utilizada para proporcionar diferentes tipos de acceso a Empleados. Por ejemplo, el tipo de acceso garantizado a al público diferirá del tipo de acceso proporcionado a un supervisor de empleado, al departamento de nominas, o al Ministerio de Hacienda. Utilice la herencia para distinguir entre al menos cuatro tipos diferentes de acceso a la información de Empleado.
13. Realizar una clase que calcule el menor valor de 2 datos dados.
14. Realizar una clase que calcule el mayor valor de 2 datos dados.