

Enunciado - Muebleria

La mueblería "Muebles Elegantes" necesita desarrollar un sistema para gestionar su catálogo de muebles. Se requiere implementar un programa en Java que utilice conceptos de clases, herencia, interfaces y manejo de excepciones para administrar los diferentes tipos de muebles que ofrece la tienda.

Crea una clase base llamada "**Mueble**" con los siguientes atributos privados:

- nombre (String): el nombre del mueble.
- precio (double): el precio del mueble.
- stock (int): la cantidad de muebles en stock.

Esta clase debe tener un constructor que reciba los valores para los atributos y los inicialice. Además, implementa los siguientes métodos:

- getters y setters
- mostrarInformacion(): muestra en pantalla el nombre, precio y stock del mueble.

Crea tres clases derivadas de "Mueble": "**Sofa**", "**Mesa**" y "**Silla**". Cada clase derivada debe tener atributos adicionales que sean específicos del tipo de mueble. Por ejemplo, "Sofa" puede tener un atributo "material", "Mesa" puede tener un atributo "dimensiones", y "Silla" puede tener un atributo "color". Asegúrate de que cada clase derivada tenga un constructor que reciba los valores para los atributos y los inicialice correctamente. Además, cada clase derivada debe sobrescribir el método "mostrarInformacion()" para mostrar la información específica del tipo de mueble.

Crear una **clase "Mueblería"** que declare una colección (ArrayList) llamada "catalogo" para almacenar los muebles disponibles en stock. Se debe incluir el método void agregarMueble(Mueble mueble), también debe incluir un método "mostrarCatalogo()" que muestre la información de todos los muebles en el catálogo utilizando un bucle "for each".

Agrega un método llamado ordenarPorPrecio() en la clase Muebleria que ordene la colección catálogo en base al precio de los muebles, de forma ascendente.

(Nota:tenga en cuenta que las clases envoltorio implementan compareTo)

Crear una **interfaz Oferta** que incluya el método **void calcularDescuento(int porcentaje)** que setea el descuento en precio según el porcentaje. Hacer que los muebles tipo Silla implementen esta interfaz.

Tenga en cuenta la siguiente **clase Main**

```

public class Main {
    public static Mueble pedirDatos(String tipo){ //propagar excepciones
        Mueble mueble=null;
        Scanner leer=new Scanner (System.in);
        System.out.println("Ingrese nombre: ");
        String nombre=leer.nextLine();
        System.out.println("Ingrese precio: ");
        double precio=leer.nextDouble();
        leer.nextLine();
        System.out.println("Ingrese stock: ");
        int stock=leer.nextInt();
        leer.nextLine();
        switch (tipo){
            case "Sofa": System.out.println("Ingrese material: ");
                String material=leer.nextLine();
                mueble=new Sofa(nombre,precio,stock,material);
                break;
            case "Mesa": System.out.println("Ingrese dimension: ");
                double dimension=leer.nextDouble();
                leer.nextLine();
                mueble=new mesa(nombre,precio,stock,dimension);
                break;

            case "Silla": System.out.println("Ingrese color: ");
                String color=leer.nextLine();
                mueble=new Silla(nombre,precio,stock,color);
                break;

        }
        return mueble;
    }
    public static void main(String[] args) {
        Muebleria muebleria = new Muebleria();
        //controlar excepciones como dice el enunciado
        Mueble sofa = pedirDatos("Sofa");
        //agregar sofa a la coleccion
        Mueble mesa = pedirDatos("Mesa");
        //agregar mesa a la coleccion
        Mueble silla = pedirDatos("Silla");
        //agregar silla a la coleccion
        //calcular 15% de descuento a silla
        //ordenar coleccion
        //mostrar coleccion
    }
}

```

se pide modificar la clase Main para que:

- Modificar el método `static Mueble pedirDatos(String tipo)` para que **propague excepciones** del tipo `InputMismatchException` y `Exception`.

1- Dentro del método `main()`, se debe agregar un bloque `try-catch-finally` al crear un nuevo Mueble. en caso de que se produzca una excepción y proporcionar información adicional sobre el problema:

-Si ocurre una excepción tipo `InputMismatchException` se debe mostrar el mensaje "Error de tipo de datos", si ocurre una excepción `Exception`, se debe mostrar un cartel "Error general". El bloque `finally` debe mostrar un cartel "En bloque finally"

2- agregar los muebles creados a la colección.

3- calcular descuento del objeto silla.

3- ordenar la colección.

4- Mostrar la información de cada mueble usando el método de la clase `Muebleria`.