

INFORMACIÓN GENERAL

Prueba práctica de programación mediante ordenador. Está permitido consultar apuntes, código propio e información disponible en Internet. No está permitido comunicarse con otras personas.

EJERCICIO 1 (6 PUNTOS)

En *La Tienda del Barrio S.A.* necesitan un programa en lenguaje Java que les ayude a gestionar la información sobre los artículos que tienen en la tienda, así como realizar ventas a clientes y compras a proveedores. Crea un proyecto Java con las clases descritas abajo. Entrega todo el proyecto comprimido en ZIP.

Clase 'Artículo' (3 puntos)

- Un artículo tiene un identificador (número entero), un nombre, un precio de venta a cliente, un precio de compra a proveedor, un IVA (%) y un stock (representa la cantidad de ese artículo disponible en tienda). El nombre y el identificador de un artículo no pueden cambiar. El identificador ha de establecerse automáticamente y ser único. El IVA es el mismo para todos los artículos (un 21%). Tanto el precio de compra como el precio de venta han de ser valores superiores a cero. El precio de venta ha de ser superior al precio de compra. El stock ha de ser superior o igual a cero.
- Deberá tener un único constructor así como todos los getters y setters que sea posible.
- Deberá tener al menos dos métodos públicos, uno para vender (a cliente) y otro para comprar (a proveedor). En ambos casos deberá pasarse un único argumento con la cantidad de unidades a vender o comprar, y se devolverá un valor indicando si fue posible realizar la operación o no.
- Deberá tener un método público que devuelva una cadena de texto de una sola línea con la información sobre dicho artículo: id, nombre, precios, IVA y stock.
- En los casos en los que sea necesario, los métodos públicos lanzarán una excepción si no es posible realizar la operación solicitada.
- Pueden implementarse otros métodos si se considera necesario.
- Es obligatorio que esta clase no realice ningún tipo de entrada por teclado ni salida por pantalla.

Clase 'Tienda' (2 puntos)

- Contendrá la función 'main' del programa.
- Al iniciar el programa se mostrará por pantalla un menú principal con las siguientes opciones:
 1. Mostrar artículos.
 2. Venta a cliente.
 3. Compra a proveedor.
 4. Gestionar artículos.
 5. Salir.
- La opción 1 mostrará por pantalla la descripción de todos los artículos de la tienda.
- La opción 2 permitirá realizar una venta. Pedirá los identificadores y cantidades de los artículos deseados además del nombre del cliente. Mostrará el precio total y pedirá confirmar la venta.
- La opción 3 permitirá realizar una compra. Pedirá los identificadores y cantidades de los artículos deseados además del nombre del proveedor. Mostrará el precio total y pedirá confirmar la compra.
- La opción 4 mostrará un submenú con cuatro opciones: 'Añadir artículo', 'Editar artículo', 'Eliminar artículo' y 'Volver'. Las tres primeras opciones pedirán introducir la información necesaria y realizarán la operación si es posible. La cuarta opción volverá al menú principal.
- La opción 5 termina el programa.

- El menú principal y el submenú se volverán a mostrar tras cada operación hasta que el usuario elija 'Salir' o 'Volver' según el caso.
- Toda interacción con el usuario deberá realizarse por entrada y salida estándar (teclado y pantalla).
- Los artículos deberán almacenarse en memoria en alguna estructura de datos. No es necesario que haya persistencia de datos (no se precisa utilizar ficheros).
- Deberán manejarse los posibles errores y excepciones que puedan producirse.
- No programes todo directamente en la función main. Implementa distintas funciones adicionales para que el código sea lo más modular.
- Es obligatorio utilizar la clase 'Articulo'.

Clase 'Calculadora' (1 punto)

- Interfaz gráfica tipo JFrame que ayude a un empleado de la tienda a calcular el precio de venta de un artículo. Deberá contener los componentes necesarios para introducir 4 datos: Precio de un artículo, unidades de dicho artículo, descuento (%) e IVA (%). También tendrá un botón 'Calcular' que al presionarlo mostrará en un JLabel el precio final tras aplicar el descuento y luego el IVA.
- Comprueba que los datos introducidos sean válidos y maneja las posibles excepciones. En caso de error avisa al usuario mediante un cuadro de diálogo.
- La interfaz debe ser sencilla pero fácil de entender y de utilizar.
- Esta clase es independiente de las dos anteriores (no es necesario utilizar 'Artículo' ni 'Tienda').

EJERCICIO 2 (4 PUNTOS)

El *Agencia Estatal de Meteorología* (AEMET) necesita un software de análisis de datos meteorológicos para un estudio sobre el cambio climático que están realizando. A modo de ejemplo, te proporcionan el histórico de temperaturas diarias máximas y mínimas de Santa Cruz de Tenerife desde el año 2000 hasta el 2019 (descomprime el archivo **tenerife.zip**). Crea un programa Java con una sola clase que haga lo siguiente:

- Al iniciar el programa se mostrará por pantalla un menú con dos opciones:
 1. VOLCAR FICHEROS A BD
 2. PROCESAR DATOS
- La opción 1 leerá los datos históricos almacenados en los ficheros y los guardará en la tabla 'temperaturas' de la base de datos 'AEMET' almacenada en tu servidor MySQL en localhost. La base de datos y la tabla tendrás que haberlas creado previamente (puedes hacerlo desde phpMyAdmin, NetBeans o cualquier otro medio). La tabla deberá tener seis campos: Ciudad, Año, Mes, Día, Máxima y Mínima (del tipo de dato más apropiado).
- La opción 2 accederá a la información de la base de datos (no a los ficheros) y mostrará por pantalla, para cada año, la siguiente información:
 - Temperatura más alta y más baja del año.
 - Temperatura máxima media del año (suma de máximas / días).
 - Temperatura mínima media del año (suma de mínimas / días).
 - Temperatura intermedia de las dos medias anteriores (suma de ambas / 2)
 - Mayor diferencia de temperatura en un mismo día.
- Tras realizar cualquiera de las dos opciones se mostrará por pantalla un mensaje indicando si la operación fue satisfactoria y el programa terminará.

No escribas todo el código directamente en la función 'main', implementa funciones adicionales para evitar repetir código y que el programa sea modular. Entrega todo el proyecto comprimido en ZIP. También la base de datos AEMET exportada en un archivo sql.