

## Unidad 3: Programación Orientada a Objetos 1

### Ejercicios boletín 2

**NOTA:** Todas las operaciones se realizarán mediante métodos.

#### Arrays de objetos

1. Escriba una clase Producto, caracterizada por 5 atributos (uno de ellos debe ser precio de fábrica). En esta clase se debe tener el método para calcular el precio de coste, que será el precio de fábrica más un porcentaje pasado como parámetro que será el coste en el transporte.

En una clase Tienda, con un array de productos como atributo se podrá:

- Listar datos de todos los productos.
- Comprobar si un producto del array es “frágil” o no, mediante un método que devuelva un tipo boolean.
- Añadir un producto a la lista pasando como parámetro un producto.
- Calcular la cantidad invertida por la tienda en comprar todos los productos, es decir, la suma de todos los precios de fábrica.
- Calcular el PVP, Precio de Venta al Público, que será un porcentaje, dado como parámetro, sumado al precio de coste. \*FÁBRICA
- Calcular las posibles ganancias cuando se hayan vendido todos los productos, es decir, la suma de todos los precios de venta al público menos la suma de todos los precios de fábrica.

Crear un main de prueba con un menú donde se pueda hacer todo esto.

2. Estoy en crisis. He pensado en alquilar las habitaciones de mi casa, pero estilo hotel para que tenga caché. Debemos gestionar mi “minihotel” y para ello, debemos crear la clase Habitación, una clase Hotel con una lista de Habitaciones y una clase principal de prueba para gestionar el hotel (realmente habría algunas clases más, como serían Cliente, Productos, etc. Pero no entraremos en eso).

Podemos poner todos los métodos en la clase Hotel. Se podrá hacer lo siguiente:

- Agregar una habitación a la lista (cuyos atributos serán tipo, precio, qué servicios extra tiene, String con el nombre de cliente asignado, número de días contratados, si está limpia, etc.)
- Ver si está ocupada o no.
- Calcular precio final según número de días, servicios extra contratados y consumo del minibar.
- Mostrar factura.
- Aquellos métodos que se quieran añadir.

Crear un main de prueba.

3. Vamos a intentar modelar a un vendedor "callejero" móviles. Los atributos de la clase Móvil serán, como mínimo, marca, modelo, vendido o no, nuevo o de segunda mano y precio unitario. En la clase Vendedor tendremos como características un array de Móviles y total vendido (se pueden añadir más atributos si se quiere o precisa). Se deben crear métodos que hagan las siguientes operaciones:

- Comprobar cuántos móviles le quedan sin vender.
- Mostrar precio final de un producto haciendo algún descuento al precio unitario si es de segunda mano.
- Podemos incluir también, si se quiere, la posibilidad de ofrecer cambio.
- Calcular cuánto dinero se debe tener en el bolsillo cuando se hayan vendido todos los móviles.

Escribir también una clase de prueba para hacer todas las operaciones.

4. Crear un programa con una clase Cliente, caracterizado por dni, nombre, apellidos, activo o no, peso y altura. En esta clase, se podrá calcular el índice de masa corporal (IMC) de un Cliente (peso en kg/altura<sup>2</sup> en m). Crear la clase Gimnasio donde tendremos un array de clientes como característica y los métodos:

- \* Calcular IMC al cliente elegido de la lista buscando el cliente.
- \* Mostrar lista.
- \* Agregar cliente a la lista.
- \* Dar de baja (dejar de estar activo).
- \* Calcular la media de IMC entre todos los clientes.

Se pueden calcular otras medias como de altura, peso, solo de los mayores de una determinada edad...

Crear un main para probar todo.

5. Crearemos una clase llamada Electrodomestico con los siguientes atributos: precio base, color, consumo energético (letras entre A y F) y peso.

En una clase Tienda, con un array de electrodomésticos como atributo, se podrá calcular el precio de un objeto elegido por el usuario y calculado como, el precio base más la cantidad de la tabla de debajo dependiendo de la letra de consumo ya que son más caros los menos eficientes.

Se podrá calcular la suma de todos los precios de todos los objetos del array y la media de precio entre todos los objetos de la lista.

comprobarConsumoEnergetico (char letra): comprueba que la letra es correcta, si no es correcta usará una letra por defecto (la A).

Letra	A	B	C	D	E	F
Precio (€)	10	20	30	40	80	100

Crea un main para poder probar todo.

6. Este ejercicio debería ser mucho más complejo, pero solo debemos hacer lo que se pide, aunque en alguna ocasión tenga menos sentido o falten cosas por hacer. Se trata de implementar un programa para gestionar un cine con una sola sala.

Cada Entrada (una de las clases) de cine tendrá un número identificativo, precio, si está libre u ocupada, número de la fila y de butaca. Todas las funcionalidades estarán en la clase Sala, que estará caracterizada por número identificador, número total de localidades, un objeto tipo Entrada y un array de localidades (tipo Entrada).

Se darán al usuario las siguientes opciones en un switch:

- Ver cartelera que mostrará el título de la película proyectada.
- Comprar una entrada comprobando si una localidad está vendida ya anteriormente. Todas las entradas valen lo mismo. En este caso, solo hay que mirar si su atributo "libre" es verdadero o falso, sin preocuparnos por nada más. Al acabar una venta habrá que pasar esa entrada a ocupada.
- Calcular ganancias totales del día. Para ello al final de cada compra habrá que sumar la cantidad en el lugar que se haya habilitado para llevar el dinero total en caja.
- Comprar varias entradas, dando por teclado el número de las mismas. Se aplicará un descuento si se compran más de 5 entradas.
- Modificar el precio de una entrada.
- Alguno más que se te ocurra.

## De repaso

7. Puedes crear ejercicios de repaso mediante una clase Modelo (POJO), una Gestión y los métodos que hagan algún cálculo sobre sus atributos, por ejemplo:

Productos, Tiendas con precios (rebajar por días para caducar, descuentos, etc.)

Trabajadores y sueldos (pago de impuestos, extras por vender mucho...)

Series, películas o redes sociales con medias de número de likes, más vistos, menos, listar por secciones (terror, comedia...)

Alumnos y notas (medias, suspensos, cálculo de nota final mediante ponderación de exámenes, trabajos...)

Transporte, billetes...

Todos ellos pueden ser solo de dicha clase modelo y el main o con más clases como, por ejemplo, Alumno-Aula-main, Trabajador-Oficina-main, Productos-Tienda-main...