

Jerarquía de Electrodomésticos

Crearemos una superclase llamada **Electrodomestico** con las siguientes características:

Sus atributos son: precio base, color, consumo energético (letras entre A y F) y peso.

Por defecto, el color será blanco, el consumo energético sera F, el precioBase es de 100 € y el peso de 5 kg. Usa constantes para ello.

Los colores disponibles son blanco, negro, rojo, azul y gris.

Los constructores que se implementaran serán:

- Un constructor por defecto. `public Electrodomestico()`
- Un constructor con el precio y peso. El resto por defecto. `public Electrodomestico(float precioBase, float peso)`
- Un constructor con todos los atributos. `public Electrodomestico(float precioBase, String color, char consumoEnergetico, float peso)`

Los métodos que implementara serán:

- Métodos get de todos los atributos.
- **comprobarConsumoEnergetico(char letra):** comprueba que la letra es correcta, si no es correcta, usará la letra por defecto. Se invocará al crear el objeto y no será visible.
- **comprobarColor(String color):** comprueba que el color es correcto, si no lo es, usa el color por defecto. Se invocará al crear el objeto y no será visible.
- **getPrecioFinal():** según el consumo energético, aumentará su precio, y según su tamaño, también. Esta es la lista de precios:

LETRA PRECIO

A → 100 €

B → 80 €

C → 60 €

D → 50 €

E → 30 €

F → 10 €

TAMAÑO PRECIO

Entre 0 y 19 kg: 10 €

Entre 20 y 49 kg: 50 €

Entre 50 y 79 kg: 80 €

Mayor que 80 kg: 100 €

Crearemos una subclase llamada Lavadora con las siguientes características:

Su atributo es carga, además de los atributos heredados.

Por defecto, la carga es de 5 kg. Usa una constante para ello.

Los constructores que se implementarán serán:

- Un constructor por defecto. `public Lavadora()`
- Un constructor con el precio y peso. El resto por defecto. `public Lavadora(float precioBase, float peso)`
- Un constructor con la carga y el resto de atributos heredados. Recuerda que debes llamar al constructor de la clase padre. `public Lavadora(float precioBase, float peso, char consumoEnergetico, String color, int carga)` y `public Lavadora(float precioBase, float peso, int carga)`
- Un constructor con precio, peso, consumo energético y color (sin carga). `public Lavadora(float precioBase, float peso, char consumoEnergetico, String color)`

Los métodos que se implementara serán:

- Método `get` de carga.
- `getPrecioFinal()`: si tiene una carga mayor de 30 kg, aumentará el precio 50 €, sino es así no se incrementara el precio. Llama al método padre y añade el código necesario. Recuerda que las condiciones que hemos visto en la clase `Electrodomestico` también deben afectar al precio.

Crearemos una subclase llamada Television con las siguientes características:

Sus atributos son resolución (en pulgadas) y sintonizador TDT (booleano), además de los atributos heredados.

Por defecto, la resolución será de 20 pulgadas y el sintonizador será false.

Los constructores que se implementarán serán:

- Un constructor por defecto. `public Television()`
- Un constructor con el precio y peso. El resto por defecto. `public Television(float precio, float peso)`
- Un constructor con la resolución, sintonizador TDT y el resto de atributos heredados. Recuerda que debes llamar al constructor de la clase padre. `public Television(float precio, float peso, char consumo, String color, int resolucion, boolean siTDT)`

Los métodos que se implementara serán:

- Método `get` de resolución y sintonizador TDT.
- `getPrecioFinal()`: si tiene una resolución mayor de 40 pulgadas, se incrementara el precio un 30% y si tiene un sintonizador TDT incorporado, aumentara 50 €. Recuerda que las condiciones que hemos visto en la clase `Electrodomestico` también deben afectar al precio.

Ahora crea una clase ejecutable que realice lo siguiente:

Crea un `ArrayList` de `Electrodomesticos`.

Crea diferentes electrodomésticos con los valores que desees.

Deberás mostrar el precio final de cada electrodoméstico. Por ejemplo:

```
Electrodomestico [precioBase=200.0, color=Blanco, consumoEnergetico=C, peso=60.0]->
Lavadora [carga=0]
El precio final es:200.0

Electrodomestico [precioBase=150.0, color=Blanco, consumoEnergetico=F, peso=30.0]->
Lavadora [carga=5]
El precio final es:150.0

Electrodomestico [precioBase=500.0, color=Negro, consumoEnergetico=E, peso=80.0]->
Television [resolucion=0, sintonizadorTDT=false]
```

El precio final es:500.0

Electrodomestico [precioBase=250.0, color=Blanco, consumoEnergetico=F, peso=70.0]->

Television [resolucion=20, sintonizadorTDT=false]

El precio final es:250.0

Electrodomestico [precioBase=400.0, color=Blanco, consumoEnergetico=A, peso=100.0]->

Lavadora [carga=0]

El precio final es:400.0

Electrodomestico [precioBase=200.0, color=Blanco, consumoEnergetico=C, peso=60.0]->

Television [resolucion=0, sintonizadorTDT=true]

El precio final es:250.0

Electrodomestico [precioBase=50.0, color=Blanco, consumoEnergetico=F, peso=10.0]-> Television

[resolucion=20, sintonizadorTDT=false]

El precio final es:50.0

FASE II : polimorfismo

Polimorfismo: Recuerda el uso operador instanceof.

Por ejemplo, si tenemos un Electrodomestico con un precio final de 300, dos lavadoras de 200 y una televisión de 500, el resultado final sera de 1000 (300+400+500) para electrodomésticos, 400 para lavadora y 500 para televisión.

FASE III : clases enumeradas

En vez de usar los arrays para consumo energético y colores, usa clases enumeradas.