

Ejercicio1

Escribe un programa para una biblioteca que contenga libros y revistas.

- Las características comunes que se almacenan tanto para las revistas como para los libros son el código, el título, y el año de publicación. Estas tres características se pasan por parámetro en el momento de crear los objetos.
- Los libros tienen además un atributo prestado. Los libros, cuando se crean, no están prestados.
- Las revistas tienen un número. En el momento de crear. En el momento de crear las revistas se pasa el número por parámetro.
- Tanto las revistas como los libros deben tener (aparte de los constructores) un método toString() que devuelve el valor de todos los atributos en una cadena de caracteres. También tienen un método que devuelve el año de publicación, y otro el código.
- Para prevenir posibles cambios en el programa **se tiene que implementar una interfaz Prestable con los métodos prestar(), devolver() y prestado**. La clase Libro implementa esta interface

Ayuda:

1. **implementar una superclase de Libro y Revista** con sus características comunes, que se llama Publicación.
2. Se implementan las clases Libro y Revista que añaden sus nuevos atributos.
3. Se escriben constructores, que llaman al constructor de la clase padre.
4. Se **sobreescribe el método toString()**, que también llama al método toString() de la superclase.
5. La interfaz Prestable declara los métodos indicados sin implementarlos. La clase Libro la implementa.

Ejercicio2

Crearemos una supeclase 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 sera 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. No importa si el nombre esta en mayúsculas o en minúsculas.
- Los constructores que se implementaran serán
- Un constructor por defecto.
- Un constructor con el precio y peso. El resto por defecto.
- Un constructor con todos los atributos.
- Los métodos que implementara serán:
- Métodos get de todos los atributos.
- **comprobarConsumoEnergetico(char letra)**: comprueba que la letra es correcta, sino es correcta usara la letra por defecto. Se invocara al crear el objeto y no sera visible.
- **comprobarColor(String color)**: comprueba que el color es correcto, sino lo es usa el color por defecto. Se invocara al crear el objeto y no sera visible.
- **precioFinal()**: según el consumo energético, aumentara su precio, y según su tamaño, también.

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 implementaran serán:
- Un constructor por defecto.
- Un constructor con el precio y peso. El resto por defecto.
- Un constructor con la carga y el resto de atributos heredados. Recuerda que debes llamar al constructor de la clase padre.
- Los métodos que se implementara serán:
- Método get de carga.
- **precioFinal()**: si tiene una carga mayor de 30 kg, aumentara 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 **Televisión** con las siguientes características:

- Sus atributos son **resolución** (en pulgadas) y **sintonizador TDT** (booleano), además de los atributos heredados.
- Por defecto, la resolución sera de 20 pulgadas y el sintonizador sera false.
- Los constructores que se implementaran serán:
- Un constructor por defecto.
- Un constructor con el precio y peso. El resto por defecto.
- Un constructor con la resolución, sintonizador TDT y el resto de atributos heredados. Recuerda que debes llamar al constructor de la clase padre.
- Los métodos que se implementara serán:
- Método get de resolución y sintonizador TDT.
- **precioFinal()**: 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 array de Electrodomesticos de 10 posiciones.
- Asigna a cada posición un objeto de las clases anteriores con los valores que desees.
- Ahora, recorre este array y ejecuta el método precioFinal().
- Deberás mostrar el precio de cada clase, es decir, el precio de todas las televisiones por un lado, el de las lavadoras por otro y la suma de los Electrodomesticos (puedes crear objetos Electrodomestico, pero recuerda que Television y Lavadora también son electrodomésticos). Recuerda el uso operador instanceof.

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

Ejercicio 3-Se pretende realizar un programa para gestionar la lista de participaciones en una competición de salto de longitud. El número de plazas disponible es de 10. Sus datos se irán introduciendo en el mismo orden que vayan inscribiéndose los atletas, **por lo que el n° de dorsal se corresponde con el n° de inscripción**

Crea la clase Atleta con los atributos dorsal,Nombre, un array de 3 elementos para consignar la mejor marca del 2021, mejor marca del 2020 y mejor marca del 2019 de ese atleta

Diseñar el programa que muestre las siguientes opciones:

- 1- Inscribir un participante.
- 2- Mostrar listado de datos.
- 3- Mostrar listado por marcas.
- 4- Finalizar el programa.

Si se selecciona 1, se introducirán los datos de uno de los participantes: Nombre, **la mejor marca del 2021, mejor marca del 2020 y mejor marca del 2019 de ese atleta**

Si se elige la opción 2, se debe mostrar un listado por número de dorsal.

La opción 3 mostrará un listado ordenado por la marca del 2021, de mayor a menor.

Tras procesar cada opción, se debe mostrar de nuevo el menú inicial, hasta que se seleccione la opción 4, que terminará el programa.

Ten en cuenta que aunque el array es de 10 posiciones, en un momento dado, puede haber inscritos solo 3 atletas por ej y que una vez que haya inscritos 10 atletas ya no se podrán inscribir más atletas.

Piensa si es necesario crear la clase Competicion, que tendrá como atributo un array de Atletas (de 10 máximo) y un contador para saber el n.º de atletas inscritos hasta el momento en la competición.