

Ejercicios Declaración de clases

Ejercicio nº 1,

Deseamos crear una clase llamada **Persona** que siga las siguientes condiciones:

- Sus atributos son: nombre, edad, DNI, sexo (H hombre, M mujer, peso y altura. No queremos que se accedan directamente a ellos. Piensa que modificador es el más adecuado, también su tipo. Si quieres añadir algún atributo puedes hacerlo
- Por defecto, todos los atributos menos el DNI serán valores por defecto según su tipo (0 números, cadena vacía para String, etc.). Sexo será hombre por defecto, usa una constante para ello.
- Se implantarán varios constructores:
 - Un constructor por defecto
 - Un constructor con el nombre, edad y sexo, el resto por defecto
 - Un constructor con todos los atributos como parámetros.
- Los métodos que se implementarán son:
 - **calcularIMC()**: calculará si la persona está en su peso ideal (peso en kg/(altura ^ 2 en m)), si esta fórmula devuelve un valor menor que 20, la función devuelve un -1, si devuelve un número entre 20 y 25 (incluidos), significa que está por debajo de su peso ideal la función devuelve un 0 y si devuelve un valor mayor que 25 significa que tiene sobrepeso, la función devuelve un 1. Se recomienda utilizar constantes para devolver estos valores.
 - **esMayorDeEdad()**: indica si es mayor de edad, devuelve un booleano.
 - **comprobarSexo(char sexo)**: comprueba que el sexo introducido es correcto. Si no es correcto, será H. No será visible al exterior.
 - **toString()**: devuelve toda la información del objeto.
 - **generarDNI()**: genera un número aleatorio de 8 cifras, genera a partir de este su número de letra correspondiente. Este método será invocado cuando se construya el objeto. Puedes dividir el método para que te sea más fácil. No será visible al exterior.
 - Métodos set de cada parámetro, excepto de DNI
 - Métodos get de cada parámetro.

Ahora, crea una clase ejecutable que haga lo siguiente:

- Pide por teclado el nombre, la edad, sexo, peso y altura.
- Crea 3 objetos de la clase anterior, el primer objeto obtendrá las variables anteriores pedidas por teclado, el segundo objeto obtendrá los valores anteriores excepto el peso y la altura y el último por defecto, para este último utiliza los métodos set para darle a los atributos un valor.
- Para cada objeto, deberás comprobar si está en su peso ideal, tiene sobrepeso por debajo de su peso ideal con un mensaje.
- Indicar para cada objeto si es mayor de edad o no.
- Por último, mostrar la información de cada objeto.

Ejercicio nº 2,

Deseamos crear una clase llamada **Password** que siga las siguientes condiciones:

- Que tenga los atributos **longitud** y **contraseña**. Por defecto, la longitud será de 8 caracteres.
- Los constructores serán los siguientes:
 - Un constructor por defecto
 - Un constructor con la longitud que nosotros le pasemos. Generará una contraseña aleatoria con esa longitud
- Los métodos que implementa serán:
 - **esFuerte()**: devuelve un booleano si es fuerte o no, para que sea fuerte debe tener más de 2 mayúsculas, más de 1 minúscula y más de 5 números.
 - **generarPassword()**: genera la contraseña del objeto con la longitud que tenga.
 - Método get para contraseña y longitud
 - Método set para longitud

Ahora, crea una clase ejecutable que haga lo siguiente:

- Crea un array de Passwords con el tamaño que tú le indiques por teclado.
- Comprobar si los passwords generados son fuertes o no.
- Informar al usuario de los password buenos y los malos, de estos últimos indicando cuál es el motivo para desecharlo

Ejercicio nº 3,

Deseamos crear una clase llamada **Electrodoméstico** que siga las siguientes condiciones:

- Sus atributos son precio base, color, consumo energético (letras entre A y F) y peso.
- Por defecto, el color será blanco, el consumo energético será F, el precio Base es de 100 € y el peso de 5 kg. Utiliza constantes para ello.
- Los colores disponibles son blanco, negro, rojo, azul y gris. No importa si el nombre está en mayúsculas o en minúsculas.
- Los constructores que se implementarán 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 implementará serán:
 - Métodos get de todos los atributos.
 - **comprobarConsumoEnergetico(char letra)**: comprueba que la letra es correcta, sino 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, sino lo es usa el color por defecto. Se invocará al crear el objeto y no será visible.
 - **precioFinal()**: según el consumo energético, aumentará su precio, y según su tamaño, también. Los siguientes cuadros muestran estos 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 €

Ahora, crea una clase ejecutable que te permita probar el funcionamiento de la clase en los casos que veas conveniente.

Ejercicio nº 4.

Con el ejercicio nº 3 del apartado de declaración de clases, vamos a realizar a continuación las siguientes tareas:

➤ 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
 - Un constructor con el precio y el peso. El resto por defecto
 - Un constructor con la carga y el resto de atributos heredados. Recuerda que deben llamar al constructor de la clase padre.
- Los métodos que se implementarán serán:
 - Método get de carga.
 - precioFinal(): si tiene una carga mayor de 30 Kg. aumentará el precio 50 €, sino es así el precio no se incrementará. Llamar al método padre y añadir el código necesario. Recuerda que las condiciones que hemos visto en la Clase Electrodoméstico 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 será de 20 pulgadas y el sintonizador será false
- Los constructores que se implementarán serán:
 - Un constructor por defecto
 - Un constructor con el precio y el 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 implementarán serán:
 - Método get de resolución y sintonizador TDT.
 - PrecioFinal(). Si tiene una resolución mayor de 40 pulgadas, se incrementará el precio un 30% y si tiene un sintonizador TDT incorporado, aumentará 50 €. Recuerda que las condiciones que hemos visto en la clase Electrodoméstico también deben afectar al precio.

➤ Ahora crea una clase ejecutable que realice lo siguiente:

- Crear un array de Electrodomésticos con 10 posiciones.
- Asigna a cada posición un objeto de las clases anteriores con los valores que desees
- Ahora, recorrer este array y ejecutar 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 Electrodomésticos (puedes crear objetos Electrodoméstico, pero recuerda que Televisión y Lavadora también son electrodomésticos), Recuerda el usos del operador instanceof

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

Ejercicio nº 5.

Crearemos una clase llamada **Serie** con las siguientes características:

- Sus atributos son **título, número de temporadas, entregado, género y creador**.
- Por defecto, el número de temporadas es de 3 temporadas y entregado false. El resto de atributos serán valores por defecto según el tipo de atributo.
- Los constructores por se implementarán serán:
 - Un constructor por defecto
 - Un constructor con el título y creador. El resto por defecto
 - Un constructor con todos los atributos, excepto de entregado.
- Los métodos que se implementarán serán:
 - Métodos get de todos los atributos, excepto de entregado.
 - Métodos set de todos los atributos, excepto de entregado.
 - Sobrescribe los métodos toString.

➤ Crearemos una clase **VideoJuego** con las siguientes características:

- Sus atributos son título, horas estimadas, entregado, género y compañía.
- Por defecto, las horas estimadas serán de 10 horas y entregado false. El resto de atributos serán valores por defecto según el tipo de atributo.
- Los constructores que se implementarán serán:
 - Un constructor por defecto.
 - Un constructor con el título y horas estimadas. El resto por defecto
 - Un constructor con todos los atributos, excepto de entregado.
- Los métodos que se implementarán serán:
 - Métodos get de todos los atributos, excepto de entregado
 - Métodos set de todos los atributos, excepto de entregado.
 - Sobrescribe los métodos toString.

Como vemos, en principio, las clases anteriores no son padre-hija, pero si tienen en común, por eso vamos a hacer un interfaz llamada Entregable con los siguientes métodos:

- **entregar()**: cambia el atributo prestado a true.
- **devolver()**: cambia el atributo prestado a false
- **isEntregado()**: devuelve el estado del atributo prestado.
- Método **compareTo(Objeto a)**, compara las horas estimadas en los videojuegos y en las series el número de temporadas. Como parámetro que tenga un objeto, no es necesario implementar la interfaz Comparable. Recuerda el uso del casting de objetos.