

Práctica uno

- Haz una clase llamada Persona que siga las condiciones expuestas a continuación. Intenta hacer el ejercicio de dos maneras diferentes (con class y sin class).
 - Sus atributos son: nombre, edad, DNI, sexo (H hombre, M mujer), peso y altura.
 - Vamos a querer crea objetos de tres maneras diferentes:
 1. Con un constructor vacío.
 2. Con el nombre, edad y sexo, y el resto con ceros o blancos.
 3. Un constructor con todos los atributos como parámetros.
 - Los métodos que se implementarán son:
 - **calcularIMC()**: Este método calculará el índice de masa corporal ($\text{peso} / (\text{altura} * \text{altura})$). Si ésta fórmula devuelve un valor menor que 20, la función **devuelve un -1**. Si se obtiene un número entre 20 y 25 (ambos incluidos), significa que está por debajo de su peso ideal así que la función **devuelve un 0**. Si se obtiene un valor mayor que 25 significa que tiene sobrepeso, la función **devuelve un 1**.
 - **esMayorDeEdad()**: indica si es mayor de edad devolviendo un valor booleano.
 - **comprobarSexo(char sexo)**: comprueba que el sexo introducido es correcto. Si no es correcto, sera H.
 - **toString()**: devuelve toda la información del objeto.
 - **generaDNI()**: genera un número aleatorio de 8 cifras, genera también la letra correspondiente. Este método será utilizado cuando se construya el objeto.
 - Métodos set de cada parámetro, excepto de DNI.

Ahora a través de un formulario html solicita el nombre, la edad, sexo, peso y altura y crea objetos de manera diferente.

Para cada objeto, comprobar si está en su peso ideal, tiene sobrepeso o está por debajo de su peso ideal y muestra un mensaje. Indicar para cada objeto si es mayor de edad. Por último, mostrar todos los datos.

■ Ejercicio dos

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 tres y entregado false. El resto de atributos serán valores por defecto según el tipo de dato (0 o blancos).

Los constructores que se implementaran serán:

- Un constructor por defecto.
- Un constructor con el título y creador. El resto por defecto.
- Un constructor con todos los atributos, excepto el entregado.

Los métodos que se implementara serán:

- Métodos get de todos los atributos, excepto entregado.
- Métodos set de todos los atributos, excepto entregado.

Crearemos una **clase Videojuego** con las siguientes características:

Sus atributos son título, horas estimadas, entregado, género y compañía.

Por defecto, las horas estimadas serán diez y entregado false. El resto de atributos serán valores por defecto según el tipo del atributo.

Los constructores que se implementaran 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 hay que implementar:

- Métodos get de todos los atributos, excepto de entregado.
- Métodos set de todos los atributos, excepto de entregado.

Como vemos, en principio, las clases anteriores no son padre-hija, pero si tienen elementos en común, por eso vamos a hacer otra **clase 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.

compareTo (Object a), compara las horas estimadas en los videojuegos y en las series el número de temporadas.

Relaciona las tres clases.

Ahora crea una aplicación y realiza lo siguiente:

- Crea dos arrays, uno de Series y otro de Videojuegos, de 5 posiciones cada uno.
- Crea un objeto en cada posición del array, con los valores que desees, debes usar distintos constructores.
- Entrega algunos Videojuegos y Series con el método entregar().
- Cuenta cuántas Series y Videojuegos hay entregados.
- Por último, indica el Videojuego que tiene más horas estimadas y la serie con más temporadas. Muestra en pantalla toda su información.