

RELACIÓN DE EJERCICIOS DE CLASES

1. Crear una clase “Fecha” que nos ayude a gestionar las fechas. Para ello se debe seguir con lo siguiente:
 - Constructor que tiene como argumento de entrada (anio, mes, dia).
 - Los métodos setter y getter necesarios.
 - Tendrá los siguientes métodos:
 - Método “to_string” que tiene no tiene argumentos de entrada, ya que se invoca desde el propio objeto y que devuelve una cadena de caracteres con el siguiente formato “dd/mm/aaaa”.
 - Métodos mágicos de la representación normal, legible y aquella que sea necesaria para comparar.
 - Método “edad”, que se invoca desde el propio objeto y que devuelve un entero de los años que tienen en función de la fecha actual según el sistema (no es requisito tener en cuenta los años bisiestos).
2. Crea una clase llamada **Cuenta** que tendrá los siguientes atributos: titular (que es una persona) y cantidad (puede tener decimales). El titular será obligatorio y la cantidad es opcional. Construye los siguientes métodos para la clase:
 - Un constructor.
 - Realiza la representación normal y la legible para los usuarios.
 - Los setters y getters para cada uno de los atributos. El atributo no se puede modificar directamente, sólo ingresando o retirando dinero.
 - **mostrar()**: Muestra los datos de la cuenta.
 - **ingresar(cantidad)**: se ingresa una cantidad a la cuenta, si la cantidad introducida es negativa, no se hará nada.
 - **retirar(cantidad)**: se retira una cantidad a la cuenta. La cuenta puede estar en números rojos y la cantidad ha de ser positiva.
3. Vamos a definir ahora una “Cuenta Joven”, para ello vamos a crear una nueva clase **CuentaJoven** que deriva de la anterior. En este caso se ha de introducir un nuevo argumento denominado “nacimiento” para validar si el usuario cumple con los requisitos de una cuenta joven o no, este nuevo argumento es de tipo “Fecha” (utilizando la clase del ejercicio 1). Construye los siguientes métodos para la clase:
 - Un constructor.
 - Los setters y getters necesarios.
 - Realiza la representación normal y la legible para los usuarios.
 - En esta ocasión los titulares de este tipo de cuenta tienen que ser mayor de edad, por lo tanto hay que crear un método **esTitularValido()** que devuelve verdadero si el titular es mayor de edad pero menor de 25 años y falso en caso contrario. Este método ha de consultarse antes de realizar una instancia de esta clase.

Piensa los métodos heredados de la clase madre que hay que reescribir.

4. Crea una clase que tenga como objetivo almacenar sus propias instancias para poder gestionarlas a posteriori, es decir un stock interno.

La clase en cuestión se ha de llamar “Vehiculo” y tiene que tener como argumentos de entrada, la matrícula y marca.

Se han de crear los siguientes métodos:

- a) Los Setter y Getter.
 - b) El constructor, que además ha de validar si la instancia en cuestión existe o no de forma interna y en caso de error, mostrarlo.
 - c) La representación normal y legible para el ser humano.
 - d) El método estático imprimir, que ha de mostrar el total de vehículos dados de alta representados en su forma legible.
 - e) El método estático baja, para eliminar del stock interno un vehículo determinado.
 - f) El método mágico hash, teniendo en cuenta que ha de ser una identificación única para cada instancia.
 - g) Agregar al constructor una nueva funcionalidad, que agregue un nuevo atributo al objeto llamada “id_entrada” que se identifica con un número de entrada para el sistema. Este número ha de ser consecutivo y en el caso de haber un hueco por la baja de un vehículo, se tiene que utilizar este número ya que está disponible.
 - *Nota:* para agregar este atributo, se debe tener en cuenta que se ha de agregar en aquellos métodos donde sea necesario.
5. Utilizando la clase anterior, se quiere crear una nueva clase llamada Coche que sea subclase de “Vehiculo”, y que tenga un constructor que añada los argumentos de, modelo y potencia.
- Se deben modificar los métodos de la representación normal y la legible para el usuario, además de agregar los setter y getter necesarios.