

CLASES (II)

1. Crea una clase Persona para guardar nombre, apellidos, DNI y edad de una persona.
 - a. Define un constructor, donde se puedan indicar los datos iniciales (pueden estar vacíos).
 - b. Para cada atributo define sus correspondientes setter (para poder validar el valor de entrada: nombre, apellidos y DNI no pueden ser cadenas vacías y se guardará en mayúsculas. Edad debe ser un valor entero positivo).
 - c. Para cada atributo define sus correspondientes getter.
 - d. Añade el método mostrar(), que muestra los datos de la persona (nombre, apellidos, DNI edad).
 - e. Añade el método mayorDeEdad(), que indica si la persona es mayor de edad o no.
2. Crea una clase Cuenta, que tendrá los siguientes atributos: titular (que es una persona) y cantidad (puede tener decimales).
 - a. Define un constructor, teniendo en cuenta que el titular es obligatorio y la cantidad es opcional.
 - b. Define los getter y setter, teniendo en cuenta que la cantidad no se puede modificar directamente sino realizando ingresos o retiradas de dinero.
 - c. Define ingresar(cantidad), que ingresa la cantidad indicada (hay que comprobar que la cantidad es positiva).
 - d. Define retirar(cantidad), que retira la cantidad indicada (hay que comprobar que la cantidad es positiva). La cuenta se puede quedar en números rojos.
 - e. Define mostrar, que muestra todos los datos de la cuenta.
3. Crea una clase CuentaJoven, para clientes menores de 25 años. Hereda de la clase Cuenta, a la que se añade:
 - a. Un atributo bonificacion, que guarda el porcentaje de bonificación que se le da al cliente a final de año. Al tratarse de un porcentaje, debe ser un valor entre 0 y 100.
 - b. Adapta mostrar para que se vea toda la información de la cuenta joven.
4. Crea un programa que haga uso de las tres clases para crear una cuenta normal y una cuenta joven.