

Ejercicio 1

Se le solicita construir una clase de nombre **CuentaCorriente** que permita almacenar los datos asociados a la cuenta bancaria de un cliente, e interactuar con ellos.

Esta clase tendrá las siguientes propiedades, métodos y constructores:

1. Propiedades privadas:
 - DNI, nombre, apellidos, distrito donde vive, teléfono: todas de tipo string.
 - Saldo: de tipo real.
2. Constructores:
 - Constructor que por defecto inicializa las propiedades de la clase.
 - Constructor al que se le pasan como argumentos todas las propiedades que tiene la clase.
3. Métodos públicos:
 - set() y get() para todas las propiedades de la clase.
 - retirarDinero() : resta al saldo una cantidad de dinero pasada como argumento.
 - ingresarDinero() : añade al saldo una cantidad de dinero.
 - consultarCuenta() : visualizará los datos de la cuenta.
 - saldoNegativo() : devolverá un valor lógico indicando si la cuenta está o no en números rojos.

El programa en Python debe hacer lo siguiente

- Ingresar los datos de un conjunto de cuentas corrientes que serán almacenadas en una lista. El ingreso de datos finalizará cuando se ingrese como DNI una cadena vacía. Verifique que el DNI debe tener una longitud de 8.
- Realizar una transferencia de dinero de una cuenta a otra, para lo cual debe ingresar los DNI de la persona que retira y hacia quien se le deposita.
- Determinar y mostrar el saldo promedio de las personas que viven en determinado distrito.
- Determinar y mostrar los datos de las personas cuyo apellido empieza con una determinada letra.
- Indicar el o los datos de las personas que tienen el mayor saldo.



Ejercicio 2

Se le solicita construir una clase de nombre **Hora** que permita almacenar la hora, así como los métodos para manipularla. Tendrá las siguientes propiedades y métodos:

1. Propiedades privadas:

- hora: de tipo entero (00 a 23)
- minutos: de tipo entero (00 a 59)
- segundos: de tipo entero (00 a 59)

2. Constructores:

- Constructor que, por defecto, inicialice las propiedades de la clase a 0.
- Constructor al que se le pasen como argumentos tres enteros y se los asigne a las propiedades de la clase. Si la cantidad recibida no satisface las restricciones de los valores impuestos a horas, minutos y segundos, el valor que se fija es 0.

3. Métodos públicos:

- setHora() : recibe como argumentos tres enteros y se los asigna a las propiedades de la clase. Utiliza el mismo nombre en las variables que reciben los argumentos y en las propiedades de la clase. Incluir una condición: si los valores recibidos no satisfacen las restricciones de los valores impuestos a horas, minutos y segundos, el valor que se establece es 0.
- getHora() : devuelve la hora como una lista de la forma [horas, minutos, segundos] o como un string de la forma "horas:minutos:segundos".
- imprmirHora() que muestra la hora en formato string de la forma "horas:minutos:segundos".
- Métodos set() y get() para todas las propiedades.

El programa en Python debe hacer lo siguiente

- Ingresar los datos de un conjunto de horas que serán almacenadas en una lista. El ingreso de datos finalizará cuando se ingrese como hora el valor de -1.
- Incrementar en un segundo todas las horas de la lista. Luego mostrar las nuevas horas.
- Determinar y mostrar para cada una de las horas de la lista cuanto tiempo (horas, minutos y segundos) les falta para llegar al final del día.
- Determinar y mostrar el tiempo (hora, minutos y segundos) existente entre la menor y mayor hora registrada en la lista.