

Ejercicios POO Bloque 01

Ejercicio 1:

Vamos a crear una clase llamada Persona. Sus atributos son: nombre, edad y DNI. Construye los siguientes métodos para la clase:

- Un constructor, donde los datos pueden estar vacíos.
- Los setters y getters para cada uno de los atributos. Hay que validar las entradas de datos.
- mostrar(): Muestra los datos de la persona.
- esMayorDeEdad(): Devuelve un valor lógico indicando si es mayor de edad.

Ejercicio 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, donde los datos pueden estar vacíos.
- 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.

Ejercicio 3:

Realizar un programa que conste de una clase llamada Alumno que tenga como atributos el nombre y la nota del alumno. Definir los métodos para inicializar sus atributos, imprimirlos y mostrar un mensaje con el resultado de la nota y si ha aprobado o no.

Ejercicio 4:

Desarrollar un programa que cargue los datos de un triángulo. Implementar una clase con los métodos para inicializar los atributos, imprimir el valor del lado con un tamaño mayor y el tipo de triángulo que es (equilátero, isósceles o escaleno).

Ejercicio 5:

Realizar un programa en el cual se declaren dos valores enteros por teclado utilizando el método `__init__`. Calcular después la suma, resta, multiplicación y división. Utilizar un método para cada una e imprimir los resultados obtenidos. Llamar a la clase Calculadora.

Ejercicio 6:

Realizar una clase que administre una agenda. Se debe almacenar para cada contacto el nombre, el teléfono y el email. Además deberá mostrar un menú con las siguientes opciones

- Añadir contacto
- Lista de contactos
- Buscar contacto
- Editar contacto
- Cerrar agenda

Ejercicio 7:

En un banco tienen clientes que pueden hacer depósitos y extracciones de dinero. El banco requiere también al final del día calcular la cantidad de dinero que se ha depositado.

Se deberán crear dos clases, la clase cliente y la clase banco. La clase cliente tendrá los atributos nombre y cantidad y los métodos `__init__`, `depositar`, `extraer`, `mostrar_total`.

La clase banco tendrá como atributos 3 objetos de la clase cliente y los métodos `__init__`, `operar` y `deposito_total`.

Ejercicio 8:

Desarrollar un programa que conste de una clase padre Cuenta y dos subclases PlazoFijo y CajaAhorro. Definir los atributos titular y cantidad y un método para imprimir los datos en la clase Cuenta. La clase CajaAhorro tendrá un atributo llamado número de cuenta, y un método para mostrar la información.

La clase PlazoFijo tendrá dos atributos propios, plazo e interés. Tendrá un método para obtener el importe del interés ($\text{cantidad} \times \text{interés} / 100$) y otro método para mostrar la información, datos del titular plazo, interés y total de interés.

Crear al menos un objeto de cada subclase.

Ejercicio 9:

Basándonos en el ejercicio 2. Vamos a definir ahora una “Cuenta Joven”, para ello vamos a crear una nueva clase CuantaJoven que deriva de la anterior. Cuando se crea esta nueva clase, además del titular y la cantidad se debe guardar una bonificación que estará expresada en tanto por ciento. Construye los siguientes métodos para la clase:

- Un constructor.
- Los setters y getters para el nuevo atributo.
- 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.
- Además la retirada de dinero sólo se podrá hacer si el titular es válido.
- El método `mostrar()` debe devolver el mensaje de “Cuenta Joven” y la bonificación de la cuenta.

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