

Ejercicios Introducción a Programación Orientada a Objetos (POO)

Ejercicio 1: Clase Producto

1. Crea una clase llamada `Producto` con los siguientes atributos privados:
 - `nombre` (tipo `String`).
 - `precio` (tipo `double`).
 - `cantidad` (tipo `int`).
- Implementa un constructor para inicializar los atributos.
- Crea métodos `getter` y `setter` para todos los atributos.
- Agrega un método llamado `mostrarDetalles` que imprima la información del producto en la consola.
- En el programa principal, crea un objeto de la clase `Producto`, establece sus atributos con los `setters`, y muestra los detalles con el método `mostrarDetalles`.

Ejercicio 2: Validación en los Setters

1. Modifica la clase `Producto` del ejercicio anterior para añadir validación en los `setters`:
 - El `precio` no puede ser negativo; si se intenta establecer un valor negativo, debe mostrarse un mensaje de error.
 - La `cantidad` no puede ser negativa; si se intenta establecer un valor negativo, debe asignarse un valor por defecto de 0.
- Crea un objeto de la clase `Producto` en el programa principal, intenta asignar valores negativos para el `precio` y la `cantidad` y verifica que la validación funciona correctamente.

Ejercicio 3: Clase Cuenta Bancaria

1. Crea una clase llamada `CuentaBancaria` con los siguientes atributos privados:
 - `numeroCuenta` (tipo `String`).
 - `saldo` (tipo `double`).
- Implementa un constructor para inicializar los atributos.

- Crea métodos getter para ambos atributos, pero no crees un setter para numeroCuenta (debe ser inmutable una vez creada la cuenta).
- Añade los siguientes métodos públicos:
 - depositar(double cantidad): Incrementa el saldo en la cantidad especificada si es positiva.
 - retirar(double cantidad): Disminuye el saldo en la cantidad especificada si es positiva y no excede el saldo actual.
- En el programa principal, crea una instancia de CuentaBancaria, realiza algunas operaciones de depósito y retiro, e imprime el saldo final.