Ejercicios Herencia – Encapsulación

1. **Crea una clase Animal con atributos protegidos nombre y edad. Añade un método info() que muestre sus atributos. Luego crea clases que hereden de Animal (Perro, Gato) y añade un método específico en cada una (ladrar() en Perro, maullar() en Gato). Instancia objetos y llama a todos los métodos.**
2. **Crea una clase Vehiculo con atributos protegidos marca y modelo. Incluye un método mostrar_info(). Deriva clases Automovil y Motocicleta, agregando atributos específicos y sobreescribiendo mostrar_info() para incluir todos los datos. Instancia uno de cada y muestra la info.**
3. **Diseña una clase Empleado con atributos nombre y salario. Hereda a clases Gerente y Vendedor. Añade métodos específicos: subir_subsidio() en Gerente, vender() en Vendedor. Usa herencia múltiple si quieres agregar más funcionalidades.**
4. **Crea una clase InstrumentoMusical con método sonar(). Las clases hijas (Guitarra, Piano) sobreescriben sonar() para emitir sonidos diferentes. Implementa una función que reciba una lista de instrumentos y los haga sonar según su clase.**
5. **Implementa una jerarquía de clases Cuenta con métodos depositar() y retirar(). Hereda clases CuentaAhorros y CuentaCorriente. Cada subclase debe añadir restricciones distintas (ejemplo, un máximo de retiro, o intereses). Ejecuta operaciones en varias cuentas.**

Problemas de Encapsulamiento (nivel intermedio):

1. **Crea una clase CuentaBancaria con atributos privados __saldo, __titular. Implementa getter y setter solo para saldo. Añade un método transferir() para mover saldo entre dos objetos de CuentaBancaria. Crea dos cuentas y realiza transferencias, mostrando los saldos.**
2. **Diseña una clase Producto con atributos privados __precio y __cantidad. Usa métodos para aumentar y reducir stock, y doble control en el setter del precio para no aceptar valores negativos. Crea un producto e realiza varias operaciones.**
3. **Haz una clase Poligono con atributos protegidos lados (lista). Implementa un método area() que se sobreescriba en clases hijas Triangulo, Rectangulo. Usa encapsulamiento para mantener los lados protegidos y calcular áreas en las subclases.**
4. **Crea una clase Alumno con atributos privados __notas (lista). Incluye métodos para agregar notas, calcular promedio y obtener notas. Limita el acceso directo a las notas y obliga a usar los métodos. Testea con datos reales.**