Programación
UD10 – Apuntes. Utilización avanzada de clases
Ejercicios 1

Ficheros

1. Se necesita desarrollar un sistema de gestión de empleados para una empresa. La empresa tiene diferentes tipos de empleados, entre ellos empleados comunes y gerentes. Todos los empleados tienen un nombre y un salario base.

   Los gerentes, además del salario base, reciben un bono adicional por su cargo. Se debe crear una jerarquía de clases utilizando el concepto de herencia, en donde la clase Empleado sea la clase base, y Gerente sea una subclase que hereda de Empleado.

   Se requiere implementar:

   - Un constructor en la clase Empleado que inicialice el nombre y el salario.
   - Un método mostrarInfo() en Empleado que muestre el nombre y el salario del empleado.
   - Una clase Gerente que herede de Empleado e incluya un atributo adicional llamado bono.
   - Un constructor en Gerente que inicialice el nombre, el salario y el bono.
   - Sobreescribir el método mostrarInfo() en Gerente para incluir el bono.
   - Un programa principal que cree objetos de ambas clases y muestre su información.

2. Se requiere modelar un sistema para gestionar vehículos en un estacionamiento. Existen diferentes tipos de vehículos como autos, motocicletas y camiones, cada uno con características propias.

   El sistema debe contar con una clase base llamada Vehiculo que tenga un método describirVehiculo(), el cual será sobrescrito en cada subclase para describir el tipo de vehículo de manera diferente.

   Se debe implementar:

- Una clase Vehiculo con un método describirVehiculo() que imprima un mensaje genérico.
- Tres clases que hereden de Vehiculo: Auto, Motocicleta y Camion.
- Cada subclase debe sobrescribir describirVehiculo() con un mensaje específico sobre su tipo.
- En el programa principal, se deben crear instancias de cada tipo de vehículo y almacenarlas en un arreglo de tipo Vehiculo, demostrando el polimorfismo al llamar describirVehiculo() en cada una.

3. Se debe diseñar un sistema de pago electrónico en el que se puedan realizar pagos mediante diferentes métodos: tarjeta de crédito, PayPal y transferencia bancaria.

El sistema debe utilizar una clase base abstracta MetodoPago que defina un método procesarPago(), el cual será implementado de manera diferente en cada clase concreta.

Se debe implementar:

- Una clase abstracta MetodoPago con un método abstracto procesarPago(double monto).
- Tres clases concretas (TarjetaCredito, PayPal, TransferenciaBancaria) que hereden de MetodoPago y sobrescriban procesarPago() con una implementación específica.
- En el programa principal, se deben crear objetos de cada tipo de pago y procesar pagos con diferentes montos, demostrando la abstracción.

4. Se necesita desarrollar una aplicación para gestionar cuentas bancarias. Cada cuenta tiene un número de cuenta, un titular y un saldo. Para garantizar la seguridad de la información, se debe aplicar el concepto de encapsulación.

   Se debe implementar:

- Una clase CuentaBancaria con atributos privados: numeroCuenta, titular y saldo.
- Métodos públicos depositar(double monto) y retirar(double monto) que modifiquen el saldo de manera segura.
- Un método mostrarSaldo() que muestre el saldo actual.
- Métodos getTitular() y setTitular() para acceder y modificar el titular.
- En el programa principal, se debe crear una cuenta bancaria, realizar depósitos y retiros, y mostrar el saldo actualizado.

5. Se quiere desarrollar un sistema de seguridad para el manejo de usuarios en una aplicación. Cada usuario tiene un nombre de usuario y una contraseña, pero la contraseña debe estar oculta y no ser accesible desde fuera de la clase.

Se debe implementar:

- Una clase Usuario con atributos privados nombreUsuario y contrasena.
- Un constructor que permita inicializar ambos valores.
- Un método verificarContrasena(String ingreso) que compare la contraseña ingresada con la almacenada y devuelva true si coinciden, o false en caso contrario.
- Un método cambiarContrasena(String nuevaContrasena) que permita cambiar la contraseña solo si cumple con ciertos requisitos (Los vistos en el examen de expresiones regulares. Al menos una mayúscula, al menos un dígito y como mínimo 8 caracteres).
- En el programa principal, se debe crear un usuario, intentar verificar la contraseña y cambiarla, demostrando el concepto de ocultación de información.