1. Ejercicio: Sistema de Reservas de Vuelos

A diagram of a business

Description automatically generated

Crea un sistema de reservas de vuelos para una aerolínea. La aerolínea tiene diferentes tipos de vuelos: Vuelos regulares y Vuelos charter.

- Crea una clase abstracta Vuelo con atributos como numeroVuelo, origen, destino, y fecha. La clase debe tener un método abstracto calcularPrecio().

- Crea dos clases VueloRegular y VueloCharter que hereden de la clase Vuelo.

- La clase VueloRegular debe tener un atributo adicional numeroAsientos y el método calcularPrecio() debe devolver un precio base por asiento multiplicado por el número de asientos.

- La clase VueloCharter debe tener un atributo adicional precioTotal y el método calcularPrecio() debe devolver el precio total del vuelo charter.

- Crea una interfaz Promocionable con un método aplicarPromocion().

- Modifica las clases VueloRegular y VueloCharter para que implementen la interfaz Promocionable. El método aplicarPromocion() debe aplicar una promoción al precio del vuelo.

- Crea una clase Reservas que tenga un array de objetos Vuelo y métodos para agregar vuelos, calcular el precio total de las reservas, aplicar promociones a los vuelos promocionables y mostrar la información de todos los vuelos.

Crea la clase SistemaReservas con un main para probar el sistema.

2. Ejercicio: Sistema de Pagos

A diagram of a system

Description automatically generated

Crea un sistema de pagos para una tienda en línea. La tienda acepta diferentes métodos de pago: Tarjeta de crédito y PayPal.

- Crea una clase abstracta MetodoPago con atributos como titular y numero. La clase debe tener un método abstracto realizarPago().

- Crea dos clases TarjetaCredito y PayPal que hereden de la clase MetodoPago.

- La clase TarjetaCredito debe tener atributos adicionales como fechaExpiracion y codigoSeguridad y el método realizarPago() debe simular el proceso de pago con tarjeta de crédito.

- La clase PayPal debe tener un atributo adicional correoElectronico y el método realizarPago() debe simular el proceso de pago con PayPal.

- Crea una interfaz Cancelable con un método cancelarPago().

- Modifica las clases TarjetaCredito y PayPal para que implementen la interfaz Cancelable. El método cancelarPago() debe cancelar el pago si es posible.

- Crea una clase Pagos que tenga un array de objetos MetodoPago y métodos para agregar métodos de pago, realizar pagos, cancelar pagos y mostrar la información de todos los métodos de pago.

Crea la clase SistemadePagos y un main para probar el sistema

3. Ejercicio: Sistema de Notificaciones

A diagram of a computer

Description automatically generated

Crea un sistema de notificaciones para una aplicación móvil. La aplicación envía notificaciones a través de diferentes canales: Correo electrónico y Mensajes de texto.

- Crea una clase abstracta Canal Notificación con atributos como usuario y mensaje. La clase debe tener un método abstracto enviarNotificacion().

- Crea dos clases CorreoElectronico y MensajeTexto que hereden de la clase CanalNotificacion.

- La clase CorreoElectronico debe tener un atributo adicional direccionCorreo y el método enviarNotificacion() debe simular el proceso de envío de correo electrónico.

- La clase MensajeTexto debe tener un atributo adicional numeroTelefono y el método enviarNotificacion() debe simular el proceso de envío de mensaje de texto.

- Crea una interfaz Personalizable con un método personalizarMensaje().

- Modifica las clases CorreoElectronico y MensajeTexto para que implementen la interfaz Personalizable. El método personalizarMensaje() debe permitir personalizar el mensaje de notificación.

- Crea una clase Notificaciones que tenga un array de objetos CanalNotificacion y métodos para agregar canales de notificación, enviar notificaciones, personalizar mensajes y mostrar la información de todos los canales de notificación.

Crea una clase SistemaNotificaciones con un main para probar el sistema