

Guia #3

Diseño e Interfaces

1. Implementar un sistema básico para la gestión de diferentes tipos de transporte urbano. Utilizar interfaces para definir comportamientos comunes y enumeradores para representar diferentes tipos de transporte.

Consignas:

a. Enumerador TipoTransporte:

- Define los tipos de transporte: AUTOBUS, TRANVIA, TREN, BICICLETA.

b. Interfaz Transporte:

- Define los métodos arrancar(), detener(), y obtenerCapacidad().

c. Clase Autobus:

- Implementa la interfaz Transporte.
- Atributos: numeroLinea (entero), capacidad (entero), tipo (TipoTransporte).
- Métodos: Implementa los métodos definidos en la interfaz y un método adicional para anunciar la próxima parada (anunciarParada(String parada)).

d. Clase Tranvía:

- Implementa la interfaz Transporte.
- Atributos: ruta (cadena de texto), capacidad (entero), tipo (TipoTransporte).
- Métodos: Implementa los métodos definidos en la interfaz y un método adicional para cambiar de vía (cambiarVia(int nuevaVia)).

e. Clase Bicicleta:

- Implementa la interfaz Transporte.
- Atributos: numeroSerie (cadena de texto), tipo (TipoTransporte).
- Métodos: Implementa los métodos definidos en la interfaz y un método adicional para ajustar la altura del asiento (ajustarAsiento(int altura)).
- obtenerCapacidad() devuelve 1 (porque solo tiene capacidad para una persona).

En la clase Main:

1. Crea instancias de cada clase (Autobus, Tranvía, Bicicleta).
2. Simula el arranque, la detención y la obtención de la capacidad de cada transporte.
3. Muestra el tipo de transporte y el estado de cada uno en la consola.

2. Hemos tenido una entrevista con un cliente que desea que diseñemos un videojuego para su empresa y nos ha brindado la siguiente información: “En un lejano reino, existe un bosque mágico habitado por criaturas muy especiales. Cada animal tiene habilidades únicas, y algunos incluso poseen más de una destreza sorprendente. Tu misión es adentrarte en este reino y modelar las clases necesarias para representar a estos fascinantes seres.

En el juego, el personaje principal conocerá a un pato, un murciélago y un pez. El pato, por ejemplo, no solo camina como cualquier otro animal, sino que también puede volar sobre el agua y nadar bajo ella. El murciélago, por su parte, vuela en la oscuridad y se cuelga boca abajo en cuevas. El pez, aunque no puede volar ni caminar, se mueve con agilidad en las profundidades del océano.

Cada una de estas criaturas realiza actividades esenciales como comer y dormir, pero también despliegan sus habilidades únicas dependiendo de la situación en la que se encuentran.

Además, te encontrarás con algunas decisiones interesantes. ¿Cómo se comunican estos animales? ¿Qué sonidos emiten cuando están felices o en peligro? Y lo más importante, ¿cómo representarías estas criaturas en un programa de manera que puedan mostrar sus habilidades únicas cuando se les llame a la acción?

Tu tarea es crear una simulación en la que puedas modelar a estos animales, permitiéndoles comer, dormir, y desplegar sus habilidades especiales como volar, nadar, o cualquier otra que puedas imaginar. Además, considera cómo podrías organizar tu código para que sea fácil de entender y modificar en el futuro.”

3. Imagina que trabajas en una empresa que se especializa en la gestión de una flota de vehículos, y te han encomendado la tarea de desarrollar una pequeña aplicación en Java para gestionar diferentes tipos de vehículos.

En tu sistema, todos los vehículos tienen características básicas como la marca, el modelo y la velocidad máxima. Además, algunos vehículos tienen capacidades específicas, como transportar pasajeros o llevar carga.

Para modelar esta situación, deberás crear una interfaz llamada `Vehiculo` que defina los métodos básicos que todos los vehículos deben tener. Luego, define otras dos interfaces que extiendan de `Vehiculo`: `VehiculoDePasajeros` para los vehículos que transportan pasajeros y `VehiculoDeCarga` para los vehículos que transportan carga.

Tu objetivo es implementar esta estructura en un programa donde se puedan crear diferentes tipos de vehículos, como automóviles, camiones, y autobuses. Cada uno de estos tipos de vehículos debe implementar la interfaz adecuada y definir comportamientos específicos.

Por ejemplo, un automóvil será un vehículo de pasajeros y debe implementar un método para establecer la cantidad de pasajeros que puede llevar. Un camión será un vehículo de carga y debe implementar un método para definir la capacidad de carga en toneladas. Pero una bicicleta, ¿Qué interfaz debe implementar?

Finalmente, crea un programa que permita agregar estos vehículos a un sistema de gestión y los muestre por pantalla, junto con sus características y capacidades.