Documentación del proceso

Punto 1: creación de las clases derivadas a partir de vehículo

Primero, creé la clase base llamada vehículo, que contiene propiedades y métodos comunes para todos los vehículos. Esta clase incluye:

Propiedades:

- ***Int velocidad: guarda la velocidad actual del vehículo.***
- ***String marca: almacena la marca del vehículo.***

Métodos:

- ***Acelerar(int cuanto): aumenta la velocidad del vehículo.***
- ***Frenar(): reduce la velocidad del vehículo.***
- ***Encender() y apagar(): métodos para encender o apagar el vehículo.***

Luego, creé tres clases derivadas de la clase vehículo:

Autodecombustión: representa un vehículo que utiliza gasolina.

Motocicleta: representa un vehículo de dos ruedas.

Camión: representa un vehículo de carga pesada.

Cada una de estas clases derivadas hereda los métodos y propiedades de la clase vehículo, pero también tiene sus propias características y comportamientos específicos.

Punto 2: encapsulación de propiedades en las clases derivadas

Para proteger los datos internos de cada vehículo y evitar que sean modificados directamente desde fuera de la clase, utilicé el concepto de encapsulación. Esto se hizo declarando ciertas propiedades como privadas, para que solo se puedan acceder o modificar a través de métodos públicos.

Por ejemplo, en la clase autodecombustión, encapsulé la propiedad tanquedecombustible como privada para protegerla de cambios no

controlados. Luego, creé un método público llamado cargargasolina(int cantidad) que permite modificar esta propiedad de manera controlada.

También hice lo mismo en la motocicleta y en el camión, encapsulando propiedades como cilindraje y cargamaxima para garantizar que los datos sean protegidos y solo puedan ser modificados de forma controlada.

Punto 3: sobrescritura de métodos en las clases derivadas

Una vez que las clases derivadas heredaron de la clase base vehículo, necesitaba que cada vehículo tuviera su comportamiento específico para ciertos métodos, como acelerar y frenar. Para ello, utilicé la técnica de sobrescritura de métodos.

En el autodecombustión, al acelerar, no solo se aumenta la velocidad, sino que también disminuye el nivel de gasolina.

En la motocicleta, la aceleración es más rápida que en un auto, así que el incremento de velocidad es mayor.

En el camión, la aceleración es más lenta debido al peso y la carga que transporta.

Punto 4: resumen general de la implementación

Herencia:

> ***Las clases autodecombustión, motocicleta y camión heredan de la clase base vehículo. Esto me permitió reutilizar el código común en todos los vehículos y, al mismo tiempo, agregar comportamientos específicos a cada tipo de vehículo.***

Polimorfismo:

> ***Al sobrescribir los métodos acelerar y frenar, logré que cada vehículo tuviera su propia versión de estos métodos. Este es un ejemplo de polimorfismo, ya que aunque el nombre de los métodos es el mismo, su comportamiento cambia según el tipo de vehículo.***

Encapsulación:

> ***Utilicé encapsulación para proteger las propiedades de cada vehículo (como tanquedecombustible, cilindraje, y cargamaxima). Esto evita que otras clases modifiquen directamente estos valores y***

asegura que el acceso a estos datos se haga solo de manera controlada.

Conclusión

El proyecto simula cómo funcionan distintos tipos de vehículos. Cada vehículo tiene su comportamiento único:

El autodecombustión consume gasolina cada vez que acelera.

La motocicleta acelera más rápido que los autos.

El camión tiene una aceleración más lenta debido a su tamaño y carga.

Este proyecto me permitió aplicar los conceptos de herencia, polimorfismo y encapsulación en un escenario práctico. La herencia me ayudó a crear una estructura común para todos los vehículos, mientras que el polimorfismo permitió que cada vehículo tuviera un comportamiento único. Además, con la encapsulación, pude proteger los datos internos de cada clase y asegurar que solo se accediera a ellos de manera controlada.