Universidad Mariano Gálvez de Guatemala Campus Jutiapa

Programación I

Ing. Ruldin Ayala

📝 Tarea: Herencia y Polimorfismo en C# 🚙⚡

Jhony Abraham de León Pérez

0905-24-22282

El Progreso, Jutiapa 21 de marzo de 2025

Jhony Abraham de León Pérez
0905-24-22282

Herencia y Polimorfismo en C#

Punto 1:

Derivé la clase AutoDeCombustión, la clase Motocicleta y la clase Camión de la clase padre Vehículo.

Punto 2:

En la sección no. 2 comencé realizando la función de frenado la cual la realicé guiándome de la función que hicimos en clase de acelerar modificándola: (velocidad -= cuanto;) donde si llamas al método Frenar con un valor de **cuanto** igual a 10, y la velocidad actual es 50, la nueva velocidad será 40.

Al agregar los dos métodos adicionales no me convencía mucho porque no sabía directamente como saber si estaba encendido o pagado, pero declaré una variable booleana que se llama encendido que es igual a false, ya realizando las funciones, las dos las realicé con un if else, la de encendido nos dice si: **!encendido** es verdadero se imprime un mensaje que dice "El vehículo ha sido encendido." Y si no "El vehículo ya está encendido.".

La de apagar es casi igual ya que primero verifica si el auto esta encendido **if (encendido)** Si el vehículo está apagado (encendido = false) muestra el mensaje "El vehículo ha sido apagado." Y si no "El vehículo ya está apagado.". Lo raro es que si al inicio sea encendido = false o true siempre da la primera opción solo cambia si niego al inicio de la función if.

En **AutoDeCombustión**, el método Frenar podría reducir la velocidad y gastar un poco de combustible, para realizar esto utilice el ejemplo que usted proporciono en clase con el carro eléctrico donde al acelerar pierde batería con el public override void pero cambie los términos para llamar a la función frenar y en vez de que disminuya la batería lo cambie a nivelGasolina--;

En **Motocicleta**, el método Acelerar podría aumentar la velocidad más rápido que en un auto, acá no me complique ya que volvía a utilizar la función override para llamar a la función acelerar y en base.Acelerar(cuanto*2) lo único que hice fue poner que al parámetro que se le agregue a la velocidad de aceleración de la moto se multiplique por dos.

Uno de los aspectos que me gusto trabajar en la clase Motocicleta es que al frenar se encienda las luces intermitentes, para esto declare un **private bool intermitentes = false**, donde las luces al ser falsa permanecían apagadas, después se realizo una función luces intermitentes donde se niega (**!**) el estado de las misma, por ultimo en la función de public override frenar que es llamada de la clase padre se agrega el **if** que identifica el **!intermitentes** e imprime el mensaje en consola "Haz frenado, las luces intermitentes están encendidas".

Jhony Abraham de León Pérez
0905-24-22282

Resumen

Lo que se realizó desde un inicio fue crear una clase llamada Vehículos de la cual se derivaron 4 clases hijas las clases: CarroElectrico, AutoDeCombustión, Camión y Motocicleta.

Se creo clase por clase heredando los atributos de la clase padre esto se realizaba colocando dos puntos (:) y escribiendo Vehículos, luego dentro de las llaves de la clase se realizó el constructor donde se identifican como public añadiendo las características necesarias, se vuelven a colocar dos puntos (:) **base** y se añaden las características del constructor padre.

Se definen los nuevos atributos como private para solo poder ser modificados dentro de la clase donde fueron definidos así mismo como cada una de las funciones que le van a dar sentido a los atributos y hacerlos funcionales en este caso cada una de las funciones no devuelven un valor por eso son void.

Las funciones que se comparte en cada una de las funciones es la de acelerar, frenar, información del vehículo, encender y apagar las cuales responde a la velocidad del vehículo.

Este es un programa sencillo con el cual estamos conociendo la herencia la cual me recordó a trabajar html con css, donde cada apartado tiene funciones necesarias y en común.

En esta tarea se crean instancias de cada clase y se llaman a sus métodos para realizar diversas acciones en este caso mostrar información de cada uno de los vehículos, en aspectos como frenar, acelerar o cargar se le debe dar el valor antes de correr el programa es decir que requieren de parámetros para realizar su tarea. Coloco aspectos que me parecen relevantes para identificar y conocer de mejor manera lo que realice.

Clases e Instancias:

carroElectrico, AutoDeCombustión, Motocicleta, y Camión son clases.

miToyota, miFord, miM53, y miCamiónJA son instancias de estas clases.

Métodos:

Las funciones como **InformacionVehiculo()**, **Acelerar()**, **Frenar()**, etc., son métodos definidos dentro de estas clases.

Encapsulamiento:

Los métodos como **verNivelBateria()**, **verNivelGasolina()** permiten acceder a los datos encapsulados dentro de las instancias de las clases.

La función de este código es que cada sección del código crea una instancia de la clase vehículo, llama a varios métodos para interactuar con el objeto y muestra información relevante en la consola:

```
Console.WriteLine("Auto de Combustion:");
AutoDeCombustión miFord = new AutoDeCombustión(2015, "Suzuki Swift", "Rojo");
miFord.InformacionVehiculo();
miFord.verAscientos();
miFord.encender();
miFord.Acelerar(20);
miFord.Frenar(10);
miFord.vernivelAceite();
Console.WriteLine("Nivel de gasolina: " + miFord.verNivelGasolina());
```