

Universidad Mariano Gálvez De Guatemala Ingeniería
En Sistemas De Información Y Ciencias De La
Computación
Campus Jutiapa

Programación I

Catedrático:

Ing. Ruldin Ayala



Estudiante:

José Mario Rosales Palma

Carnet:

0905-24-17488

2025

Clase autocombustión:

Punto 1: Derive la clase autocombustión de la clase vehículo

Punto 2: Implemente las propiedades

Punto 3: Encapsule las tres propiedades:

NivelDeAceite: se puede aumentar o disminuir el nivel de aceite usando el método cargaraceite o acelerar(int cuanto) y el método verNivelDeAceite permite consultar el nivel actual de aceite sin acceder directamente a la propiedad privada.

Espejos: mostrara la cantidad de espejos

NivelLiquidoDeFrenos: es permitirá consultar el nivel del liquido de frenos

Punto 4: Sobrescribi el método override esto va permitir que la función original se modifique.

Clase motocicleta:

Punto 1: Derive la clase Motocicleta de la clase vehículo.

Punto 2: Implemente las propiedades.

Punto 3: Use los métodos para interactuar con las propiedades:

Método para consultar el tipo de freno (vertipoDeFreno) va a permitir imprimir en consola el tipo de freno

Método para consultar el método de encendido (verencendido) va devolver el valor de la propiedad encendido

Método para alternar el estado del freno de mano(ponerFrenosDemano) Cambia el valor de la propiedad frenoDemano de false a true y viceversa

Punto 4: Sobrescribir los métodos de la clase base vehículo: La clase motocicleta sobrescribe los dos métodos (acelerar y Frenar) de la clase base vehículo.

Clase camión:

Punto 1: Derive la clase camión de la clase vehículo.

Punto 2: Implemente las propiedades.

Punto 3: Use los métodos para interactuar con las propiedades

Punto 4: existen funcionalidades específicas del camión: cargar, descargar, ver capacidad carga.

Resumen General y funcionalidad del proyecto:

Cada tipo de vehículo tiene características únicas, pero comparte atributos y comportamientos básicos gracias a la herencia desde la clase base vehículo. Esto busca encapsular datos, aplicando la herencia y sobrescribiendo métodos para personalizar comportamientos específicos de cada clase.

La funcionalidad de este proyecto:

comienza de la clase base vehículo que es la que define atributos como año, modelo y color, incluyendo métodos generales que son los que se comparten por todos los vehículos, como acelerar y frenar.

Después funciona con las clases derivadas:

camión: modelo de un vehículo de carga. Agrega atributos como la capacidad de carga y métodos para descargar el camión.

Autocombustión: esta clase representa un auto que utiliza combustible, incluyen las propiedades relacionadas como el nivel de aceite.

Motocicleta: modelo una moto, con detalles como el método de encendido y un sistema de frenos.

de ahí funciona con la personalización con sobreescritura:

Las clases derivadas pueden ajustar la funcionalidad base de vehículo como por ejemplo la motocicleta acelera más rápido y tiene un sistema de frenado específico.

Y la encapsulación que son las propiedades privadas de cada clase como nivelDeAceite, tipoDeFreno o cargaActual que están protegidas mediante los métodos que permiten acceder o modificar esos valores de forma controlada.