

EJERCICIO 2 –TIPOS DE HERENCIA

Imagina que estás construyendo un sistema de gestión de vehículos en una empresa de transporte. En este sistema, hay distintos tipos de vehículos, cada uno con diferentes características. Utiliza herencia en C++ para modelar la relación entre estos tipos de vehículos.

Instrucciones:

1. **Crea una clase base** llamada `Vehiculo`. Esta clase debe tener:
 - o Atributos `marca`, `modelo` (atributos públicos) y `año` (poner `anho`) (atributo privado).
 - o Método `mostrarInformacion()` que imprime la información básica del vehículo.
2. **Public Furgoneta:**
 - o Crea una clase `Furgoneta` que hereda de `Vehiculo`.
 - o Agrega un atributo `numeroPuertas` y un método `tocarBocina()` que imprima algo como "¡Bip bip!".
3. **Private Motocicleta:**
 - o Crea una clase `Motocicleta` que herede de `Vehiculo` de manera privada.
 - o Añade un atributo específico como `tipoManillar`.
 - o Redefine `mostrarInformacion()` para que sólo muestre los atributos específicos de `Motocicleta` y no los de `Vehiculo`.
4. **Protected Camión:**
 - o Crea una clase `Camion` que herede de `Vehiculo` de manera protegida.
 - o Agrega un atributo específico como `capacidadCarga` y un método `mostrarCapacidad()` que imprima la capacidad de carga.
 - o Redefine `mostrarInformacion()` para que pueda ser usado en esta clase.
5. **Furgomoto:**
 - o Crea una clase `Furgomoto` que herede de `Furgoneta` y `Motocicleta`.
 - o Asegúrate de evitar los problemas de ambigüedad de herencia y define los métodos de forma clara para que esta clase use características tanto de `Furgoneta` como de `Motocicleta`.

Objetivo:

Instancia objetos de cada una de estas clases (`Furgoneta`, `Motocicleta`, `Camion`, y `Furgomoto`) y prueba sus métodos para verificar que la herencia funciona correctamente.