Universidad Mariano Gálvez de Guatemala

Facultad de Ingeniería en Sistemas de la Información y la Computación

Campus Villa Nueva, Guatemala

Ingeniería en Sistemas -5090

Curso: PROGRAMACIÓN I

Licenciado/Ingeniero titular: Ing. Carlos Alejandro Arias

Creación de clases en C++ Lab#6

ESTUDIANTES: Carlos Eduardo García Cortez

CARNET: 5090-24-14824

FECHA: 17/02/2025

Ejercicio 1: Definición y creación del código

Imagina que estás desarrollando un sistema para una concesionaria de autos. Debes crear un conjunto de clases que representen diferentes tipos de vehículos y sus características.

Parte 1: Definir las Clases

Debes crear al menos 5 clases con los siguientes requisitos:

- Clase base: Vehiculo
- Atributos privados: marca, modelo, precio.
- Constructor: para inicializar estos valores.
- Método mostrarInfo() que imprima los detalles del vehículo.
- Encapsulamiento: Implementa métodos get y set para acceder a los atributos privados.

- Clases derivadas: Automovil, Motocicleta, Camioneta
- Cada una debe heredar de Vehiculo.
- Deben tener un atributo propio (ej. numPuertas en Automovil, cilindrada en Motocicleta, capacidadCarga en Camioneta).
- Sobrescribe el método mostrarInfo() para incluir los nuevos atributos.

- Clase adicional: Cliente
- Atributos: nombre, edad
- Constructor: para inicializar valores.
- Método comprarVehiculo(Vehiculo v) que muestre un mensaje indicando que el cliente ha comprado un vehículo.

Parte 2: Implementación en C++

Crea un programa en C++ donde definas las clases anteriores.
Instancia al menos 3 objetos de diferentes clases derivadas.
Crea un objeto de Cliente y simula la compra de un vehículo.
Muestra en pantalla la información de los vehículos y la transacción.

Ejemplo esperado de salida:

Vehículo: Toyota Corolla, Modelo: 2022, Precio: $25,000, Puertas: 4.

Vehículo: Yamaha R1, Modelo: 2021, Precio: $18,000, Cilindrada: 1000cc.

Vehículo: Ford Ranger, Modelo: 2020, Precio: $30,000, Capacidad de carga: 1.5 toneladas.

Cliente Juan ha comprado un Toyota Corolla.

Link del repositorio de Github, donde se encuentra el código en el archivo .cpp.
https://github.com/Xplod883/POO_Laboratorio.git

Parte 3: Ejecución del programa

```
Juan Perez (Edad: 30) ha comprado el siguiente vehículo:
Marca: Toyota, Modelo: Corolla, Precio: Q25000
Número de puertas: 4
----------
Juan Perez (Edad: 30) ha comprado el siguiente vehículo:
Marca: Honda, Modelo: CBR500R, Precio: Q8000
Cilindrada: 500cilíndros
----------
Juan Perez (Edad: 30) ha comprado el siguiente vehículo:
Marca: Ford, Modelo: Ranger, Precio: Q35000
Capacidad de carga: 1000 toneladas

C:\Users\carlo\OneDrive\Documentos\UMG\Tercer semestre\Programación\Lab6\x64\Debug\Lab6.exe (proceso 12040) se cerró con
 el código 0 (0x0).
Presione cualquier tecla para cerrar esta ventana. . .
```

Parte 4: Reflexión

¿Cuál fue el mayor desafío al implementar el laboratorio?
El encapsulamiento get y set, ya que, como se sabe, no se puede manipular los datos dentro de *private:*, pero si lo podemos hacer si encapsulamos las funciones a través de funciones dentro de *public:*, en este momento, usando get y set para que se pueda modificar la marca, modelo y precio usando string y void.

¿Cómo el encapsulamiento mejora la seguridad del código?
Lo mejora al tener que llamar las funciones en private, donde no se pueden modificar, y podamos modificarlo para que haya mejor seguridad y orden en el código.

¿Por qué la herencia facilita la reutilización del código?
Usa las mismas funciones para las otras clases, en automóvil, motocicleta y camioneta, logra optimizar el código y ordenarlo para que el orden del código sea legible y que gracias a eso pueda ser reutilizable.