МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ ТА КОМП’ЮТЕРНИХ ТЕХНОЛОГІЙ

**ЗВІТ**

лабораторної роботи №5

з дисципліни «Об’єктно орієнтоване програмування»

на тему «Множинне успадкування. Проблема ромба у С++. Віртуальне успадкування»

**ВИКОНАВ:**

Студент 1-го курсу

Групи ФЕП-14

Іванський Остап Юрійович

**ПЕРЕВІРИВ:**

Асистент кафедри РКТ

Левуш Павло Назарович

Львів 2025

**МЕТА**: Проробити на практиці можливі варіанти множинного успадкування та методи уникнення проблем із ним.

**ПОСИЛАННЯ**: https://github.com/Ostap-Ivanskyi/OOP/tree/main/Lab6

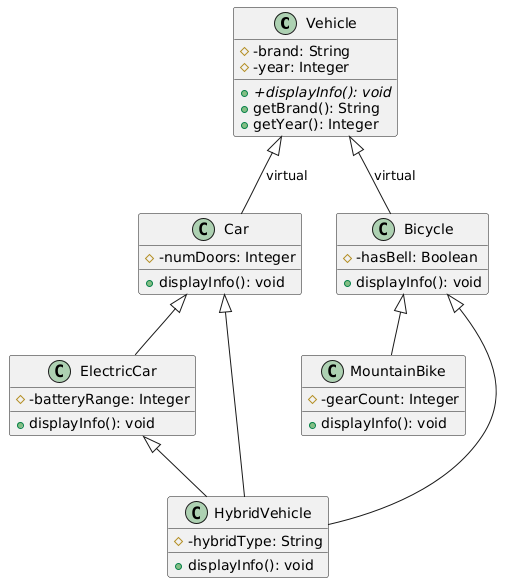
**ЗАВДАННЯ:**

До створеної ієрархії класів у лаб. 4. добавити класи таким чином, щоб отримати множинне успадкування. Переробити класи так, щоб отримати ромбовидне успадкування. Класи мають мати окрім конструкторів і звичайних методів ще й деструктори. Навчитись керувати порядком виклику конструкторів та деструкторів класів.

Реалізувати клас, який успадковується від трьох батьківських класів. Дослідити у цьому випадку порядок викликів конструкторів та деструкторів, як на це впливає віртуальне успадкування?

**ХІД РОБОТИ**

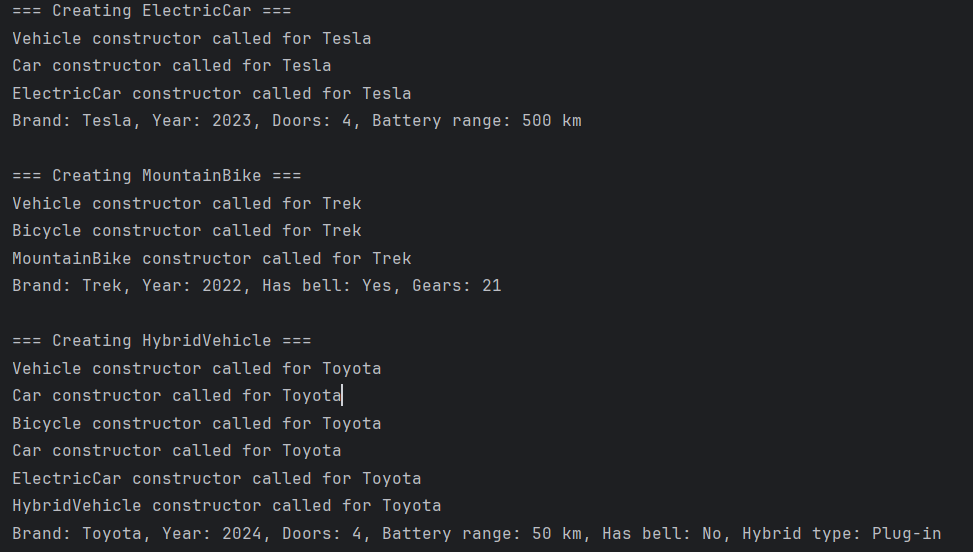
До створеної ієрархії класів додаємо класи щоб отримати множинне успадкування.



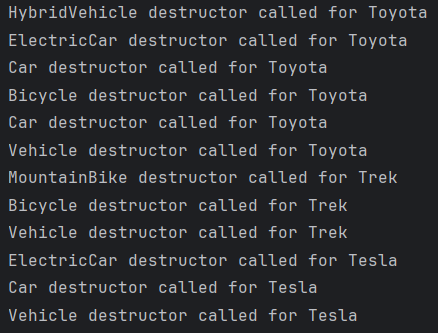
Створюємо три екземпляри класів

int main() {  
 cout << "=== Creating ElectricCar ===" << endl;  
 ElectricCar tesla("Tesla", 2023, 4, 500);  
 tesla.displayInfo();  
 cout << endl << endl;  
  
 cout << "=== Creating MountainBike ===" << endl;  
 MountainBike bike("Trek", 2022, true, 21);  
 bike.displayInfo();  
 cout << endl << endl;  
  
 cout << "=== Creating HybridVehicle ===" << endl;  
 HybridVehicle hybrid("Toyota", 2024, 4, false, 50, "Plug-in");  
 hybrid.displayInfo();  
 cout << endl << endl;  
  
 cout << "=== Objects going out of scope ===" << endl;  
 return 0;  
}

Результат виконання програми що показує порядок виклику конструкторів



Та деструкторів



Класи Car і Bicycle успадковуються від Vehicle віртуально, що дозволяє уникнути дублювання базового класу при множинному успадкуванні та вирішують проблему ромбовидного наслідування.

**ВИСНОВОК**

Під час виконання лабораторної роботи ми дослідили множинне наслідування та можливі проблеми з ним а саме проблему ромба. Також використали віртуальне наслідування яке гарантує, що базовий клас (Vehicle) буде створений лише один раз, навіть якщо він є батьківським для кількох класів у ієрархії. Без віртуального успадкування ми б отримали дві копії класу Vehicle у HybridVehicle, що призвело б до неоднозначності.