МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Факультет електроніки і комп’ютерних технологій

**Звіт**

Про виконання лабораторної роботи № 5

## «**Множинне успадкування. Проблема ромба у С++. Віртуальне успадкування**»

**Виконав:**

Студент групи ФЕП-14с

Грицюк Максим Олегович

**Перевірив:**

ас. Левуш П. Н.

Львів 2025

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

**Обладнання:**

Комп'ютер з встановленим програмним забезпеченням(Microsoft Visual Studio/Clion).

**Хід роботи**

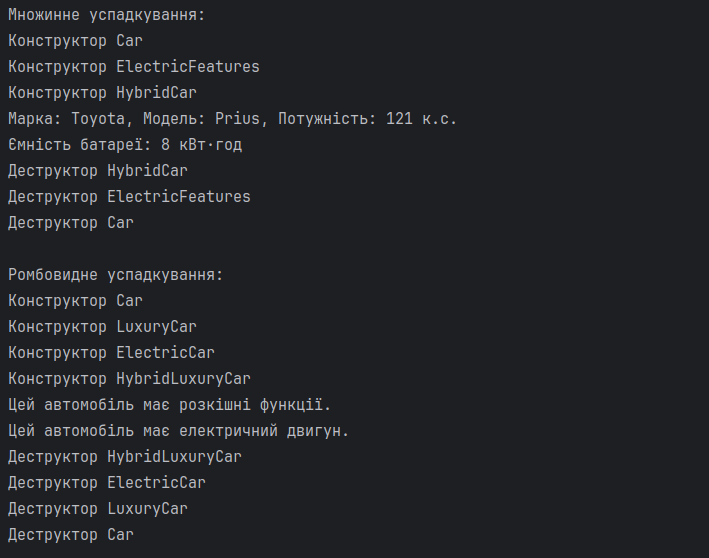
**Завдання:** До створеної ієрархії класів у лаб. 4. добавити класи таким чином, щоб отримати множинне успадкування. Переробити класи так, щоб отримати ромбовидне успадкування. Класи мають мати окрім конструкторів і звичайних методів ще й деструктори. Навчитись керувати порядком виклику конструкторів та деструкторів класів. Реалізувати клас, який успадковується від трьох батьківських класів.

**Файл Car.h:** це заголовковий файл, у якому я описав усі класи та методи, які використовуються в моїй ієрархії автомобілів. Тут є базовий клас Car, який представляє загальні характеристики автомобілів, такі як марка, модель, тип двигуна та потужність. Далі я визначив класи LuxuryCar і ElectricCar, які додають специфічні атрибути та методи для розкішних і електричних автомобілів. Клас HybridLuxuryCar демонструє ромбовидне успадкування, бо він одночасно успадковує властивості від LuxuryCar і ElectricCar. Усі методи оголошені тут, а їхня реалізація знаходиться в іншому файлі.

**Файл Car.cpp:** це файл реалізації, у якому я прописав усі методи, оголошені в Car.h. Тут я реалізував конструктори, деструктори та методи кожного класу. Наприклад, у класі Car я реалізував методи для виведення загальної інформації про автомобіль і запуску двигуна. У класі LuxuryCar додав метод для виведення розкішних характеристик, а в ElectricCar - метод для демонстрації електричних особливостей. У класі HybridLuxuryCar я перевизначив методи, щоб вони враховували як розкішні, так і електричні характеристики. Також у цьому файлі я продемонстрував, як можна уникнути дублювання базового класу через віртуальне успадкування.

**Файл main.cpp:** це основний файл програми, у якому я тестую всі створені класи. Тут я створив об'єкти для кожного класу, наприклад, для розкішного автомобіля, електромобіля та гібридного розкішного автомобіля. Для кожного об'єкта я викликав методи DisplayInfo і StartEngine, щоб показати, як працюють ці класи. Також у файлі продемонстровано роботу ромбовидного успадкування через об'єкт HybridLuxuryCar, який успадковує характеристики як від LuxuryCar, так і від ElectricCar. Цей файл показує, як застосувати принципи успадкування та поліморфізму на практиці.

**Демонстрація:**

****

**Висновок:**

У цій роботі я навчився використовувати множинне та ромбовидне успадкування, а також практично зрозумів, як працюють віртуальні методи, віртуальне успадкування та перевизначення методів. Найбільше мені сподобалося, як через віртуальне успадкування вдалося уникнути дублювання базового класу в ієрархії, що значно спрощує роботу з кодом.

**p.s** Посилання на github: https://github.com/unknownpanic/OOP\_with\_cpp