Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

Об’єктно-орієнтоване програмування

**Лабораторна робота №3**

**Тема:** «Успадкування, агрегація та композиція»

|  |
| --- |
| Виконав: ст. гр. КН-23 |
| Кобзар М. В. |
| Перевірив: Козірова Н. Л. |
|  |

Кропивницький

2024

**Тема:** Успадкування, агрегація та композиція

**Мета:** ознайомитись з основними поняттями успадкування, агрегація та

композиція в ООП та навчитись їх програмно реалізовувати мовою С++.

**Завдання:**

*Завдання* ***1***

Створіть використовуючи композицію два класи, які матимуть свої

властивості, перший реалізуйте за допомогою включення (composition), а

другий за допомогою посилання (reference), контейнером для цих класів

використайте клас з лабораторної роботи №1, створіть також третій –

похідний клас, використовуючи наслідування, базовим класом може бути або

клас з лр№1, або класи які ви створили за допомогою композиції. За

бажанням можете зобразити схематично як ваші класи залежать один від

одного.

*Завдання* ***2***

Варіант **8**

1. Створіть клас "Автомобіль", який містить такі властивості:

• Марка автомобіля

• Модель автомобіля

• Двигун (використовуйте композицію)

• Салон (використовуйте композицію)

2. Створіть клас "Двигун", який містить такі властивості:

• Тип двигуна (бензиновий, дизельний і т.д.)

• Об'єм двигуна

3. Створіть клас "Салон", який містить такі властивості:

• Кількість місць в салоні

• Тип салону (тканинний, шкіряний і т.д.)

4. Реалізуйте методи для встановлення та отримання інформації про

автомобіль, двигун та салон.

5. Напишіть демонстраційну функцію main(), в якій створюються об'єкти

класів "Автомобіль", "Двигун" та "Салон". Встановіть значення для

властивостей та виведіть інформацію про автомобіль.

6. Продемонструйте успадкування, створивши похідний клас від

"Автомобіль" (наприклад, "СпортивнийАвтомобіль") з додатковими

властивостями та методами.

7. Розширте функціональність, додавши додаткові методи та властивості

до класів за власним бажанням.

Не забудьте додати коментарі та пояснення до коду, щоб роз'яснити його

функціональність та зрозумілість.

**Варіант 8**

**Лістинг :**

Car.h

#pragma once

#include <iostream>

using namespace std;

// Клас Engine (Двигун)

class Engine {

private:

string engineType;

double engineVolume;

public:

// Конструктор

Engine(const string& type = "Unknown", double volume = 0.0)

: engineType(type), engineVolume(volume) {}

void setEngineType(const string& type) {

engineType = type;

}

string getEngineType() const {

return engineType;

}

void setEngineVolume(double volume) {

engineVolume = volume;

}

double getEngineVolume() const {

return engineVolume;

}

void showEngineInfo() const {

cout << "Engine Type: " << engineType << "\nEngine Volume: " << engineVolume << "L" << endl;

}

};

// Клас Interior (Салон)

class Interior {

private:

int seatCount;

string interiorType;

public:

// Конструктор

Interior(int seats = 0, const string& type = "Unknown")

: seatCount(seats), interiorType(type) {}

void setSeatCount(int seats) {

seatCount = seats;

}

int getSeatCount() const {

return seatCount;

}

void setInteriorType(const string& type) {

interiorType = type;

}

string getInteriorType() const {

return interiorType;

}

void showInteriorInfo() const {

cout << "Seat Count: " << seatCount << "\nInterior Type: " << interiorType << endl;

}

};

// Клас Car (Автомобіль)

class Car {

private:

string brand;

string model;

Engine engine; // Композиція: клас Engine

Interior interior; // Композиція: клас Interior

public:

// Конструктор

Car(const string& carBrand = "Unknown", const string& carModel = "Unknown",

const Engine& carEngine = Engine(), const Interior& carInterior = Interior())

: brand(carBrand), model(carModel), engine(carEngine), interior(carInterior) {}

void setBrand(const string& carBrand) {

brand = carBrand;

}

string getBrand() const {

return brand;

}

void setModel(const string& carModel) {

model = carModel;

}

string getModel() const {

return model;

}

void showCarInfo() const {

cout << "Brand: " << brand << "\nModel: " << model << endl;

engine.showEngineInfo();

interior.showInteriorInfo();

}

};

// Похідний клас SportsCar (СпортивнийАвтомобіль)

class SportsCar : public Car {

private:

double maxSpeed; // Додаткове поле

public:

// Конструктор

SportsCar(const string& carBrand, const string& carModel, const Engine& carEngine,

const Interior& carInterior, double speed)

: Car(carBrand, carModel, carEngine, carInterior), maxSpeed(speed) {}

void setMaxSpeed(double speed) {

maxSpeed = speed;

}

double getMaxSpeed() const {

return maxSpeed;

}

void showSportsCarInfo() const {

showCarInfo();

cout << "Max Speed: " << maxSpeed << " km/h" << endl;

}

};

Main.cpp

#include <iostream>

#include "Car.h"

int main() {

// Створюємо об'єкт Engine

Engine myEngine("Gasoline", 2.5);

// Створюємо об'єкт Interior

Interior myInterior(5, "Leather");

// Створюємо об'єкт Car (Автомобіль)

Car myCar("Toyota", "Camry", myEngine, myInterior);

cout << "Car Info:" << endl;

myCar.showCarInfo();

// Створюємо об'єкт SportsCar (СпортивнийАвтомобіль)

SportsCar mySportsCar("Ferrari", "488", Engine("Gasoline", 3.9), Interior(2, "Leather"), 340);

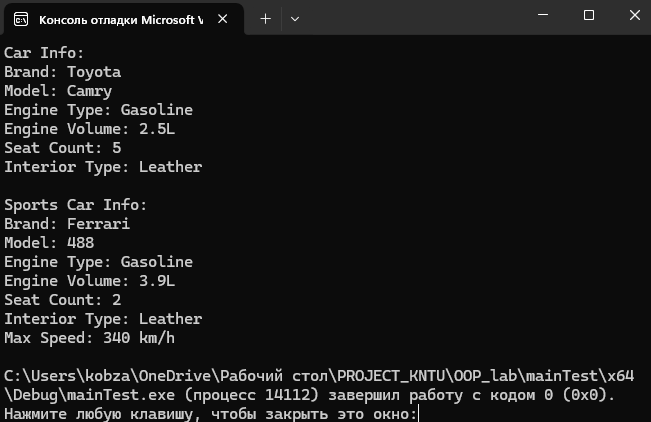
cout << "\nSports Car Info:" << endl;

mySportsCar.showSportsCarInfo();

return 0;

}

**Результат:**



**Висновок :** У ході роботи було створено три класи: “Engine”, “Interior”, та “Car”, де класи двигуна і салону реалізовані через композицію в класі автомобіля. Також створено похідний клас “SportsCar”, який успадковує властивості від класу автомобіля та додає нову властивість для максимальної швидкості. Основна функція демонструє роботу з об'єктами цих класів, їх властивості та методи, з можливістю виводу інформації про автомобіль і спортивний автомобіль.