Практична робота №4

Тема: Принцип успадкування.

Мета: Набути практичних навичок розробки ієрархії класів з використанням механізмів успадкування та узагальнення, а також використання успадкованих властивостей бібліотечних класів.

Хід роботи

1. У Git-репозиторію із попередніх практичних робіт створюю нову гілку «PR4».

2. Розробиляю клас згідно варіанту завдання:

8	Bus	id, модель, рік випуску, ціна, реєстраційний номер, кількість місць, наявність місць для осіб з інвалідністю(так/ні).

3. Перевіряю роботу класу у функції main() шляхом створення і використання об’єкта розробленого класу.

4. Роблю коміт проєкту із повідомленням «add Bus class».

5. Додаю новий клас, базовий для розроблених, згідно варіанту завдання даної практичної роботи і практичної роботи №1, класів. Виношу у базовий клас спільні атрибути і методи.

6. Перевіряю роботу похідних класів, створивши пару екземплярів кожного класу.

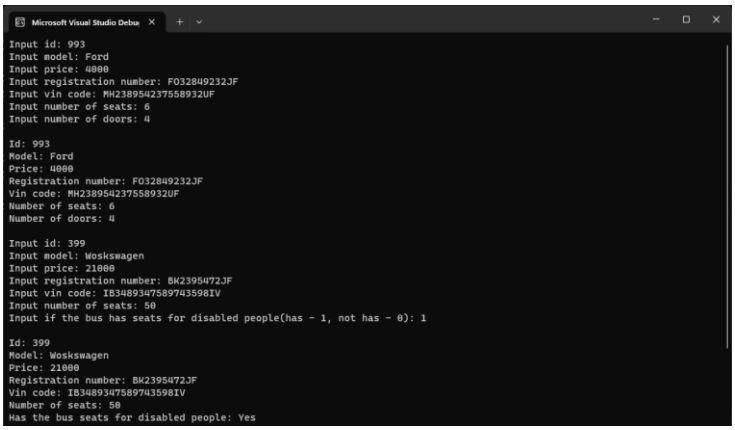
7. Роблю коміт проєкту із повідомленням «do practical work №4».

8. Зливаю гілку PR4 у гілку main (merge).

9. Відправляю зміни у гілці main у віддалений репозиторій (git push).

10. Оформлюю звіт.

Скріншот виконання програми:



Вихідний код розробленого класу:

Vehicle.h:

```
#pragma once
#include<iostream>
using namespace std;

class Vehicle
{
protected:
int id;
string model;
int price;
string registrationNumber;
string vinCode;
int numberOfSeats;
public:
Vehicle() = default;
Vehicle(int id, const string& model, int price, const string& registrationNumber, const string&
vinCode, int numberOfSeats);
Vehicle(const Vehicle& bus);
~Vehicle();
virtual void input();
virtual void output();
friend istream& operator >> (istream& in, Vehicle& bus);
friend ostream& operator << (ostream& out, Vehicle& bus);
virtual bool operator==(const Vehicle& other) const;
};
```

Vehicle.cpp:

```
#include "Vehicle.h"

Vehicle::Vehicle(
int id,
const string& model,
int price,
const string& registrationNumber,
const string& vinCode,
int numberOfSeats)
:
id(id),
model(model),
price(price),
registrationNumber(registrationNumber),
vinCode(vinCode),
numberOfSeats(numberOfSeats)
{
}

Vehicle::Vehicle(const Vehicle& vehicle)
{
this->id = vehicle.id;
this->model = vehicle.model;
this->price = vehicle.price;
this->registrationNumber = vehicle.registrationNumber;
this->vinCode = vehicle.vinCode;
this->numberOfSeats = vehicle.numberOfSeats;
}
```

```
Vehicle::~Vehicle()
{
}

void Vehicle::input()
{
cout << "Input id: "; cin >> this->id;
cout << "Input model: "; cin >> this->model;
cout << "Input price: "; cin >> this->price;
cout << "Input registration number: "; cin >> this->registrationNumber;
cout << "Input vin code: "; cin >> this->vinCode;
cout << "Input number of seats: "; cin >> this->numberOfSeats;
}

void Vehicle::output()
{
cout << "Id: " << this->id << endl;
cout << "Model: " << this->model << endl;
cout << "Price: " << this->price << endl;
cout << "Registration number: " << this->registrationNumber << endl;
cout << "Vin code: " << this->vinCode << endl;
cout << "Number of seats: " << this->numberOfSeats << endl;
}

bool Vehicle::operator==(const Vehicle& other) const
{
return id == other.id &&
model == other.model &&
price == other.price &&
registrationNumber == other.registrationNumber &&
vinCode == other.vinCode &&
numberOfSeats == other.numberOfSeats;
}

istream& operator>>(istream& in, Vehicle& vehicle)
{
vehicle.input();
return in;
}

ostream& operator<<(ostream& out, Vehicle& vehicle)
{
vehicle.output();
return out;
}
```

Висновок: під час виконання даної практичної роботи, набуто практичних навичок розробки ієрархії класів з використанням механізмів успадкування та узагальнення, а також використання успадкованих властивостей бібліотечних класів.