

Практична робота №5

Тема: Практичне застосування поліморфізму та віртуальних методів.

Мета: Закріпити знання про поліморфізм та віртуальні методи. Набути навички створення та використання поліморфних класів.

Хід роботи

1. У Git-репозиторію із попередніх практичних робіт створюю нову гілку «PR5» і переходжу в неї для виконання даної практичної роботи.
2. У базовому класі роблю один метод суто віртуальним.
3. Додаю у похідних класах реалізацію суто віртуального методу.
4. Роблю коміт проєкту із повідомленням «add virtual method».
5. Додаю функцію для виводу меню, за допомогою якого користувач зможе вибирати, який об'єкти похідного класу потрібно створити.
6. У функції main() оголошую масив вказівників на базовий клас розміром 5 елементів.
7. Створюю цикл, у якому викликаю меню і запитую користувача, який об'єкт створити, після чого створюю його функцією new(), після чого користувач має заповнити всі поля нового об'єкта. Заповнюю таким чином увесь масив.
8. Після заповнення, виводжу дані створених об'єктів у консоль.
9. За допомогою циклу for викликаю для кожного об'єкта масиву визначений суто віртуальний метод.
10. Роблю коміт проєкту із повідомленням «done practical work №4».
11. Зливаю гілку PR5 у гілку main (merge) і надсилаю зміни у гілці main у віддалений репозиторій (git push).
12. Оформлюю звіт.

Вихідний код

Main.cpp

```
#include <iostream>
#include "Car.h"
#include "Bus.h"
#include "Vehicle.h"

int main()
{
    Vehicle* vehicles[5];
    int choice;
```

```

    for (short x = 0; x < 5; x++) {
        cout << "1. Car\n2. Bus\nChoose what the object do you want to create:
"; cin >> choice;

        if (choice == 1) {
            vehicles[x] = new Car;
            vehicles[x]->input();
        }
        else {
            vehicles[x] = new Bus;
            vehicles[x]->input();
        }

        cout << endl;
    }

    for (int x = 0; x < 5; x++) {
        vehicles[x]->output();
    }

    for (int x = 0; x < 5; x++) {
        vehicles[x]->beep();
    }
}

```

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();
    virtual void beep() = 0;
    friend istream& operator >> (istream& in, Vehicle& bus);
    friend ostream& operator << (ostream& out, Vehicle& bus);

```

```
virtual bool operator ==(const Vehicle& other) const;
};
```

Скріншот виконання програми:

```
Id: 273895798
Model: Audi
Price: 25000
Registration number: IO5498UH
Vin code: e98gj5h8j
Number of seats: 2
Number of doors: 2

Id: 23958390
Model: Wolkswagen
Price: 15000
Registration number: OI48957UH
Vin code: g0i45h90
Number of seats: 50
Has the bus seats for disabled people: Yes

Car beeps!
Bus beeps!
Car beeps!
Car beeps!
Bus beeps!
```

Висновок: під час виконання даної практичної роботи, закріплено знання про поліморфізм та віртуальні методи. Набуто навички створення та використання поліморфних класів.