Міністерство освіти і науки України
Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

ЗВІТ
ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ №3
з дисципліни «Об’єктно-орієнтоване програмування»

Тема:
«Успадкування, агрегація та композиція»

Виконав: студент групи КБ-24мб
Воробйов Д.С.
Перевірив:
Козірова Н.Л.

Кропивницький 2024

Мета: ознайомитись з основними поняттями успадкування, агрегація та композиція в ООП та навчитись їх програмно реалізовувати мовою С++.

Завдання:

Завдання 1:

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, «Спортивний Автомобіль») з додатковими властивостями та методами;

7. Розширте функціональність, додавши додаткові методи та властивості до класів за власним бажанням.

Завдання 2

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 лабораторної роботи №1, або

класи які ви створили за допомогою композиції. За бажанням можете зобразити схематично як ваші класи залежать один від одного.

Результати (обробка даних, графіки, діаграми, таблиці)

Завдання 1:

Лістинг main.cpp

```
#include <iostream>
#include <string>

using namespace std;

class Engine {
public:
    Engine(string type, double volume){
        this->type = type;
        this->volume = volume;
    }

    string getEngineType() {
        return type;
    }

    double getEngineVolume() {
       return volume;
    }

private:
    string type;
    double volume;
};

class Interior {
public:
    Interior(int seats, string type){
        this->seats = seats;
        this->type = type;
    }
```

```
        int getInteriorSeats() {
            return seats;
        }

        string getInteriorType() {
            return type;
        }

    private:
        int seats;
        string type;
    };

    class Car {
    public:
         Car(string brand, string model, Engine engine, Interior
interior) : engine(engine), interior(interior){
            this->brand = brand;
            this->model = model;
        }

        void carInfo() {
            cout << "Car: " << brand << " " << model << endl;

            cout << "Engine type: " << engine.getEngineType() << endl;
             cout << "Engine volume: " << engine.getEngineVolume() <<
endl;

            cout << "Interior seats: " << interior.getInteriorSeats()
<< endl;
            cout << "Interior type: " << interior.getInteriorType() <<
endl;
        }

    private:
        string brand;
        string model;
```

```
        Engine engine;
        Interior interior;
    };

    class SportsCar : public Car {
    public:
        SportsCar(string brand, string model, Engine engine, Interior
interior, double maxSpeed) : Car(brand, model, engine, interior){
            this->maxSpeed = maxSpeed;
        }

        void carInfo() {
            Car::carInfo();
            cout << "Maximum speed: " << maxSpeed << endl;
        }

    private:
        double maxSpeed;
    };

    int main() {
        Engine engineCar("Diesel", 2.0);
        Interior interiorCar(5, "Leather");

        Car car("Volkswagen", "golf", engineCar, interiorCar);
        car.carInfo();

        cout << endl;

        Engine engineSportsCar("Diesel", 5.0);
        Interior interiorSportsCar(4, "Leather");

          SportsCar  sportsCar("Ford",  "Mustang",  engineSportsCar,
interiorSportsCar, 250.4);
        sportsCar.carInfo();

        return 0;}
```

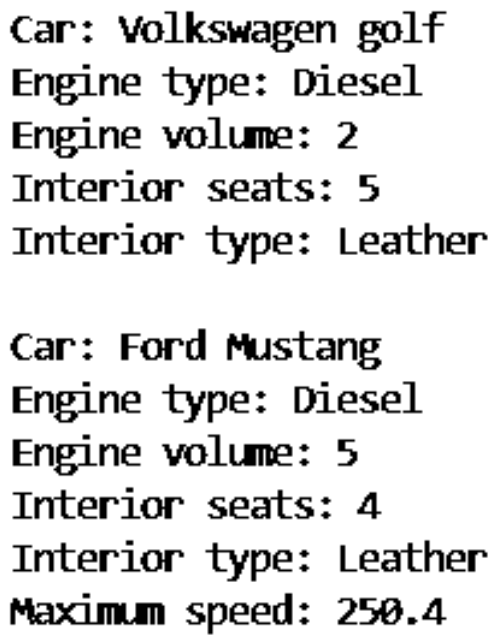

Рисунок 1 – Результат роботи програми

Завдання 2

Лістинг City.cpp

```
#include "City.h"

City::City(std::string name, int population) {
    this->name = name;
    this->population = population;
}

std::string City::getName() {
    return name;
}

int City::getPopulation() {
    return population;
}
```

Лістинг City.h

```
#ifndef CITY_H
#define CITY_H
#include <string>

class City {
public:
```

```
        City(std::string name, int population);

        std::string getName();
        int getPopulation();

    private:
        std::string name;
        int population;
    };

    #endif
```

Лістинг Country.cpp

```
    #include "Country.h"

    Country::Country(std::string name, std::string capital, int population, City city, President* president) : city(city), president(president) {
            this->name = name;
            this->capital = capital;
            this->population = population;
        }

    void Country::setName(std::string name) {
        this->name = name;
    }

    std::string Country::getName() {
        return name;
    }

    void Country::setCapital(std::string capital) {
        this->capital = capital;
    }

    std::string Country::getCapital() {
        return capital;
    }
```

```
void Country::setPopulation(int population) {
    this->population = population;
}

int Country::getPopulation() {
    return population;
}

City Country::getCity() {
    return city;
}

President* Country::getPresident() {
    return president;
}
```

Лістинг Country.h

```
#ifndef COUNTRY_H
#define COUNTRY_H
#include <string>
#include "City.h"
#include "President.h"

class Country {
public:
    Country(std::string name, std::string capital, int population,
City city, President* president);

    void setName(std::string name);
    std::string getName();

    void setCapital(std::string capital);
    std::string getCapital();

    void setPopulation(int population);
    int getPopulation();
```

```
        City getCity();
        President* getPresident();

    private:
        std::string name;
        std::string capital;
        int population;
        City city;
        President* president;
    };

    #endif
```

Лістинг DevelopedCountry.cpp

```
    #include "DevelopedCountry.h"

    DevelopedCountry::DevelopedCountry(std::string name, std::string
capital, int population, City city, President* president, double gdp) :
Country(name, capital, population, city, president){
        this->gdp = gdp;
    }

    double DevelopedCountry::getGDP() {
        return gdp;
    }
```

Лістинг DevelopedCountry.h

```
    #ifndef DEVELOPEDCOUNTRY_H
    #define DEVELOPEDCOUNTRY_H
    #include "Country.h"

    class DevelopedCountry : public Country {
    public:
         DevelopedCountry(std::string name, std::string capital, int
population, City city, President* president, double gdp);

        double getGDP();

    private:
```

```
    double gdp;
};

#endif
```

Лістинг President.cpp

```
#include "President.h"

President::President(std::string name, int age) {
    this->name = name;
    this->age = age;
}

std::string President::getName() {
    return name;
}

int President::getAge() {
    return age;
}
```

Лістинг President.h

```
#ifndef PRESIDENT_H
#define PRESIDENT_H
#include <string>

class President {
public:
    President(std::string name, int age);

    std::string getName();
    int getAge();

private:
    std::string name;
    int age;
};

#endif
```

Лістинг main.cpp

```
#include <iostream>
#include "Country.h"
#include "City.h"
#include "President.h"
#include "DevelopedCountry.h"

using namespace std;

int main() {
    City kyiv("Kyiv", 2967000);
    President president_of_ukraine("Volodymyr Zelensky", 45);

        Country ukraine("Ukraine", "Kyiv", 41258478, kyiv,
&president_of_ukraine);

    cout << "Country: " << ukraine.getName() << endl;
    cout << "Capital: " << ukraine.getCapital() << endl;
    cout << "Population: " << ukraine.getPopulation() << endl;
    cout << "City: " << ukraine.getCity().getName() << endl;
            cout << "City Population: " <<
ukraine.getCity().getPopulation() << endl;
    cout << "President: " << ukraine.getPresident()->getName() <<
endl;
    cout << "President Age: " << ukraine.getPresident()->getAge()
<< endl << endl;

    City berlin("Berlin", 3769000);
    President president_of_germany("Frank-Walter Steinmeier", 68);
        DevelopedCountry germany("Germany", "Berlin", 83149300,
berlin, &president_of_germany, 3846.4);
    cout << "Developed Country: " << germany.getName() << endl;
    cout << "Capital: " << germany.getCapital() << endl;
    cout << "Population: " << germany.getPopulation() << endl;
    cout << "City: " << germany.getCity().getName() << endl;
            cout << "City Population: " <<
germany.getCity().getPopulation() << endl;
```

```
        cout << "President: " << germany.getPresident()->getName() <<
endl;
        cout << "President Age: " << germany.getPresident()->getAge()
<< endl;
        cout << "GDP: " << germany.getGDP() << " billion USD" << endl;

        return 0;
    }
```

```
Country: Ukraine
Capital: Kyiv
Population: 41258478
City: Kyiv
City Population: 2967000
President: Volodymyr Zelensky
President Age: 45

Developed Country: Germany
Capital: Berlin
Population: 83149300
City: Berlin
City Population: 3769000
President: Frank-Walter Steinmeier
President Age: 68
GDP: 3846.4 billion USD
```

Рисунок 2 – Результат роботи програми

Висновки: в лабораторній роботі було вивчено основні концепції успадкування, агрегація та композиція в ООП, а також їх реалізацію на C++.