**­**

**Лабораторна робота №4**

З дисципліни «Програмування»

Тема: КЛАСИ І ОБ’ЄКТИ

Виконав: Землянський Едуард

Група: КВ-22

**Постановка задачі**

Написати програму, в якій створюються і знищуються об’єкти, визначеніо користувачем; виконати виклики конструкторів і деструкторів.

**Вказівки до виконання завдання**

1. Визначити в програмі класс відповідно до варіанта.

2. Визначити в класі наступні конструктори: без параметрів, з параметрами, копіювання.

3. Визначити в класі деструктор.

4. Визначити в класі методи для перегляду і встановлення полів даних.

5. Визначити вказівник на метод.

6. Визначити вказівник на екземпляр класа.

7. Передбачити розміщення об’єктів як в статичній, так і в динамічній пам’яті, а також створення масивів об’єктів.

8. Написать програму, в якій створюються і знищуються об’єкти користувацього класа і кожний виклик констурктора і деструктора супроводжується видачею відповідного повідомлення (про те, який об’єкт який конструктор або деструктор викликав).

9. Показати в програмі використання вказівника на об’єкт і вказівника на метод.

10. Програма мусить використовувати три файла:

• заголовочний h-файл з визначенням класа,

• cpp-файл з реалізацією класа,

• сpp-файл з демонстраційною програмою.

**Варіант завдання**

12. Автомобіль: (марка – char\*, потужність – int, вартість – float).

**Текст програми**

class.h

#ifndef CLASS\_H\_INCLUDED

#define CLASS\_H\_INCLUDED

#include <iostream>

#include <string>

#include <cstring>

using namespace std;

class Car{

    private:

        string mark;

        int power;

        float price;

    public:

        Car();

        Car(string markValue, int powerValue, float priceValue);

        void SetCar(string markValue, int powerValue, float priceValue);

        void GetCar();

        void CopyCar(const Car & refCar);

        ~Car();

};

#endif

class.cpp

#include "class.h"

Car::Car(){

    cout << "Constructor called for " << this << endl;

}

Car::Car(string markValue, int powerValue, float priceValue){

    cout << "Constructor called for " << this << endl;

    SetCar(markValue, powerValue, priceValue);

}

void Car::SetCar(string markValue, int powerValue, float priceValue){

    cout << "Setter called for " << markValue << " " << this << endl;

    mark = markValue;

    power = powerValue;

    price = priceValue;

}

void Car::GetCar(){

    cout << "Getter called for " << mark << " " << this << endl;

    cout << "Mark: " << mark << "\nPower: " << power << "\nPrice: " << price << endl;

}

void Car::CopyCar(const Car & refCar){

    cout << "Copying called for " << mark << " " << this << endl;

    mark = refCar.mark;

    power = refCar.power;

    price = refCar.price;

}

Car::~Car(){

    cout << "Destructor called for " << mark << " " << this << endl;

}

prog\_lab\_4.cpp

#include "class.h"

using namespace std;

int main(){

    Car car1, car2;

    car1.SetCar("Opel Zafira A", 190, 5400);

    car1.GetCar();

    car2.SetCar("Opel Zafira B", 130, 4200);

    car2.GetCar();

    car2.CopyCar(car1);

    car2.GetCar();

    Car \* car3 = new Car;

    Car \* car4 = new Car;

    car3->SetCar("BMW E5", 210, 5000);

    car3->GetCar();

    car4->SetCar("Audi R8", 220, 7090);

    car4->GetCar();

    delete car3;

    delete car4;

    Car &car5 = \* (new Car);

    Car &car6 = \* (new Car);

    car5.SetCar("Tavria Nova", 90, 900);

    car5.GetCar();

    car6.SetCar("Deviatka", 110, 1200);

    car6.GetCar();

    car6.CopyCar(car5);

    car6.GetCar();

    const int quantity = 2;

    Car Cars[quantity];

    Car car7;

    Cars[0] = car7;

    Cars[1].SetCar("Renault Master", 160, 2300);

    Cars[1].GetCar();

    Car \* Cars2;

    Cars2 = new Car[2];

    Cars2[0] = Car("Nissan Note", 120, 1500);

    Cars2[0].GetCar();

    Cars2[1] = Car("Nissan Leaf", 100, 3500);

    Cars2[1].GetCar();

    Cars2[1].CopyCar(Cars2[0]);

    Cars2[1].GetCar();

    void (Car::\*methodPtr)(const Car & refCar) = &Car::CopyCar;

    (Cars2[1].\*methodPtr)(car1);

    Cars2[1].GetCar();

    delete[] Cars2;

    return 0;

}

**Тестування програми**

PS D:\C-programs\Programming Labs 2 sem\prog\_lab\_4> .\output.exe

Constructor called for 0x2d8a3ffa90

Constructor called for 0x2d8a3ffa60

Setter called for Opel Zafira A 0x2d8a3ffa90

Getter called for Opel Zafira A 0x2d8a3ffa90

Mark: Opel Zafira A

Power: 190

Price: 5400

Setter called for Opel Zafira B 0x2d8a3ffa60

Getter called for Opel Zafira B 0x2d8a3ffa60

Mark: Opel Zafira B

Power: 130

Price: 4200

Copying called for Opel Zafira B 0x2d8a3ffa60

Getter called for Opel Zafira A 0x2d8a3ffa60

Mark: Opel Zafira A

Power: 190

Price: 5400

Constructor called for 0x1aa5e1a1770

Constructor called for 0x1aa5e1a19d0

Setter called for BMW E5 0x1aa5e1a1770

Getter called for BMW E5 0x1aa5e1a1770

Mark: BMW E5

Power: 210

Price: 5000

Setter called for Audi R8 0x1aa5e1a19d0

Getter called for Audi R8 0x1aa5e1a19d0

Mark: Audi R8

Power: 220

Price: 7090

Destructor called for BMW E5 0x1aa5e1a1770

Destructor called for Audi R8 0x1aa5e1a19d0

Constructor called for 0x1aa5e1a1770

Constructor called for 0x1aa5e1a19d0

Setter called for Tavria Nova 0x1aa5e1a1770

Getter called for Tavria Nova 0x1aa5e1a1770

Mark: Tavria Nova

Power: 90

Price: 900

Setter called for Deviatka 0x1aa5e1a19d0

Getter called for Deviatka 0x1aa5e1a19d0

Mark: Deviatka

Power: 110

Price: 1200

Copying called for Deviatka 0x1aa5e1a19d0

Getter called for Tavria Nova 0x1aa5e1a19d0

Mark: Tavria Nova

Power: 90

Price: 900

Constructor called for 0x2d8a3ffa10

Constructor called for 0x2d8a3ffa38

Constructor called for 0x2d8a3ff9e0

Setter called for Renault Master 0x2d8a3ffa38

Getter called for Renault Master 0x2d8a3ffa38

Mark: Renault Master

Power: 160

Price: 2300

Constructor called for 0x1aa5e1a5d58

Constructor called for 0x1aa5e1a5d80

Constructor called for 0x2d8a3ffc10

Setter called for Nissan Note 0x2d8a3ffc10

Destructor called for Nissan Note 0x2d8a3ffc10

Getter called for Nissan Note 0x1aa5e1a5d58

Mark: Nissan Note

Power: 120

Price: 1500

Constructor called for 0x2d8a3ffc70

Setter called for Nissan Leaf 0x2d8a3ffc70

Destructor called for Nissan Leaf 0x2d8a3ffc70

Getter called for Nissan Leaf 0x1aa5e1a5d80

Mark: Nissan Leaf

Power: 100

Price: 3500

Copying called for Nissan Leaf 0x1aa5e1a5d80

Getter called for Nissan Note 0x1aa5e1a5d80

Mark: Nissan Note

Power: 120

Price: 1500

Copying called for Nissan Note 0x1aa5e1a5d80

Getter called for Opel Zafira A 0x1aa5e1a5d80

Mark: Opel Zafira A

Power: 190

Price: 5400

Destructor called for Opel Zafira A 0x1aa5e1a5d80

Destructor called for Nissan Note 0x1aa5e1a5d58

Destructor called for 0x2d8a3ff9e0

Destructor called for Renault Master 0x2d8a3ffa38

Destructor called for 0x2d8a3ffa10

Destructor called for Opel Zafira A 0x2d8a3ffa60

Destructor called for Opel Zafira A 0x2d8a3ffa90