Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра “Електронних обчислювальних машин”



**Звіт з лабораторної роботи №7**

на тему:

“Множинне спадкування. Поліморфізм”

**Виконав:**

Ст. гр. КІ-15

Зектер В.О.

**Перевірив:**

Викладач

Козак Н.Б.

Львів – 2020

**Мета роботи:** познайомитися із множинним спадкуванням класів та поліморфізмом.

***Завдання:***

Спроектувати і реалізувати ієрархію класів, що описують предметну область згідно варіанту, яка реалізується класом 1. Клас 1 в свою чергу утворюється шляхом множинного спадкування класів 2 і 3 кожен з яких в свою чергу успадковує клас 4. Додаткові вимоги:

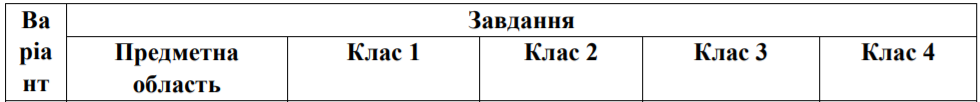
1. Базовий клас містить мінімум один віртуальний метод, один невіртуальний метод і одну динамічно створювану властивість.

2. Забезпечити механізми коректної роботи конструкторів і деструкторів.

3. Перевантажити оператор присвоєня з метою його коректної роботи.

4. Кожен з класів має містити мінімум одину властивість і 4 методи.

5. Написати main() функцію де створити об‘єкт класу 1 і продемонструвати різницю між статичним і динамічним поліморфізмом.

******

******

*Код програми:*

#include <iostream>

#include <conio.h>

using namespace std;

class CTransport {

private:

double weight;

double price;

char company\_name[30];

public:

CTransport()

{

this->weight = 5000;

this->price = 15000;

}

CTransport(double Weight, double Price)

{

this->weight = Weight;

this->price = Price;

}

CTransport& operator= (const CTransport& drob)

{

weight = drob.weight;

price = drob.price;

return \*this;

}

void set\_name(const char\* name)

{

strcpy\_s(company\_name, name);

}

void set\_price(double Price)

{

this->price = Price;

}

void set\_weight(double Weight)

{

this->weight = Weight;

}

virtual void show\_info()

{

cout << "Weight is: " << this->weight << " kg" << endl;

cout << "rice is: " << this->price << " dolars" << endl;

cout << "Company is: " << this->company\_name << endl;

}

virtual ~CTransport() {};

};

class CCar : CTransport {

private:

int speed;

bool povniyprivod;

public:

CCar()

{

this->speed = 70;

std::cout << std::boolalpha;

this->povniyprivod = true;

}

CCar(int Speed, bool Povniyprivod)

{

this->speed = Speed;

std::cout << std::boolalpha;

this->povniyprivod = Povniyprivod;

}

void set\_speed(int sp)

{

this->speed = sp;

}

void set\_povniyprivod(bool test)

{

this->povniyprivod = test;

}

CCar& operator= (const CCar& drob)

{

speed = drob.speed;

povniyprivod = drob.povniyprivod;

return \*this;

}

void show\_info()

{

cout << "Max speed is: " << this->speed << " km/h" << endl;

cout << "Povniyprivod availability: " << this->povniyprivod;

}

virtual ~CCar() {};

};

class CTruck : CTransport {

private:

double gasoline;

bool signal;

public:

CTruck()

{

this->signal = false;

this->gasoline = 1;

}

CTruck(double Gasoline, bool Signal)

{

this->gasoline = Gasoline;

this->signal = Signal;

}

CTruck& operator= (const CTruck& drob)

{

signal = drob.signal;

gasoline = drob.gasoline;

return \*this;

}

void set\_gasoline(double Gasoline)

{

this->gasoline = Gasoline;

}

void set\_signal(bool Signal)

{

this->signal = Signal;

}

void show\_info()

{

cout << "Signal availability : " << this->signal << endl;

cout << "The amount of gasoline: " << this->gasoline << " liters" << endl;

}

virtual ~CTruck() {};

};

class CMC : CTruck, CCar {

private:

int func\_number;

public:

CMC()

{

this->func\_number = 0;

};

CMC(int Func\_num)

{

this->func\_number = Func\_num;

};

CMC& operator= (const CMC& drob)

{

func\_number = drob.func\_number;

return \*this;

}

void set\_func\_num(int Func\_num)

{

this->func\_number = Func\_num;

}

void show\_info()

{

cout << "CMC has " << func\_number << " functions" << endl;

}

virtual ~CMC() {};

};

int main()

{

CTransport ct;

CTruck ctr;

CCar cc;

CMC cmc;

ct.set\_name("Lada");

ct.set\_price(10000);

ct.show\_info();

cout << endl;

cc.set\_speed(210);

cc.set\_povniyprivod(false);

cc.show\_info();

cout << endl;

ctr.set\_signal(true);

ctr.set\_gasoline(70);

ctr.show\_info();

cout << endl;

cmc.set\_func\_num(125);

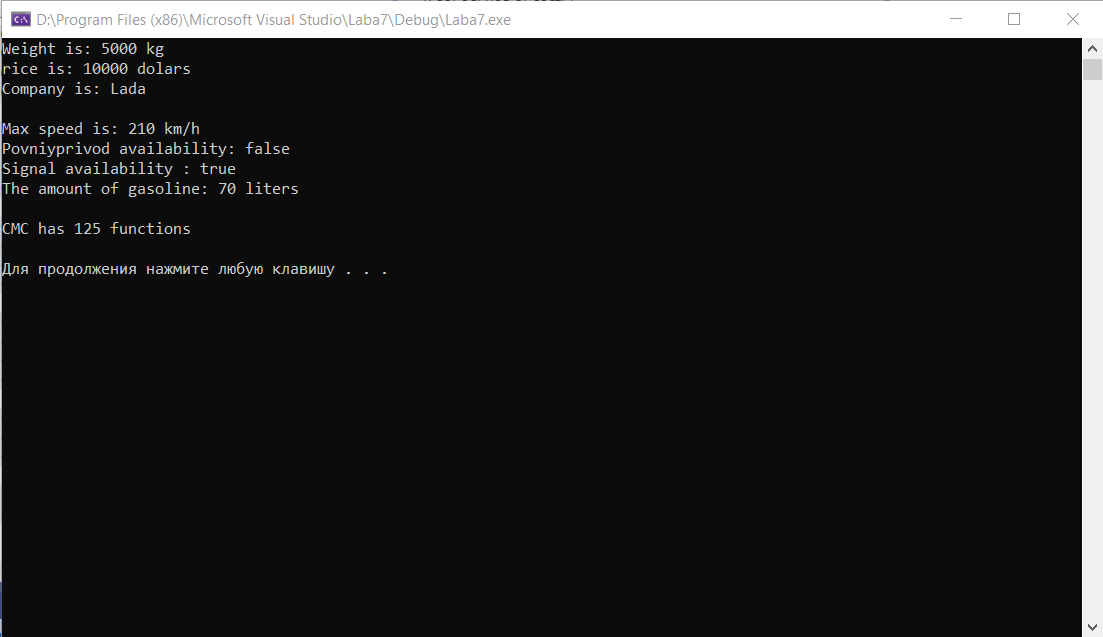
cmc.show\_info();

cout << endl;

return 0;

}

*Вікно результату:*

**

*Висновок:* я познайомився з множинним спадкуванням класів та з поліморфізмом.