Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ



**Звіт**

з лабораторної роботи №7

з дисципліни: “Програмування, ч.2 (ООП)”

на тему: “Множинне cпадкування”

Варіант 19

Виконав: ст.гр. КІ-15

Мармура В.І.

Прийняла: Викладач

Козак Н.Б.

Львів

2020

Мета: познайомитися із множинним спадкуванням і поліморфізмом.

Теоретичні відомості :

**Множинне спадкуування**

Якщо спадкування здійснюється від декількох батьківських класів одночасно, тоді воно називається множинним спадкуванням. Визначальним для похідного класу породженого множинним спадкуванням є те, що він явно чи неявно повинен успадковувати характеристики декількох базових класів. Основні принципи одинарного спадкування, зокрема спадкування членів, модифікаторів доступу до членів базових класів, розширення та обмеження характеристик, без жодних доповнень можуть бути перенесені на множинне спадкування. Неявним множинним спадкуванням можуть бути випадки змішаного спадкування. Результатом цих спадкуваннь є ієрархія, в якій похідний клас неявно (через один проміжний) успадкував характеристики двох базових класів. Якщо похідний клас породжується від декількох базових, то в декларації класу треба вказати усі базові класи, розділяючи їх комою, разом зі специфікаторами спадкування.

**Виклики членів базових класів**

Для звертання до будь-яких членів базових класів при множинному спадкуванні як і при одинарному спадкуванні слід використати оператор (::) з іменем класу. Цей підхід є найпростішим і, очевидно, найефективнішим. Більшість логічних суперечностей, які виникають при маніпуляціях з успадкованими членами в похідних класах, розв'язуються за допомогою оператора (::). Правило домінування, яке визначало порядок виклику успадкованих членів при простому спадкуванні, в цілому підтримується при множинному спадкуванні, але його застосування вимагає додаткових розумових зусиль. Правило домінування застосовується лише стосовно тих класів, які породжують одне одного, тобто є сусідніми (Подібно до одинарного спадкування при множинному спадкуванні здійснюється автоматичне створення зв‘язків від об'єктів нижчих класів до об'єктів вищих. Єдиною проблемою можуть стати неоднозначності батьківських зв'язків). При множинному спадкуванні цю умову треба особливо відслідковувати. В іншому випадку правило домінування не можна застосовувати і треба повертатись до використання оператора (::). Якщо два класи породжені один від одного, то член похідного класу є домінуючим. Правило пошуку відповідного члена для виклику відбувається на стадії компіляції. У випадку, коли правило домінування не спрацьовує треба використовувати механізм (::) і явно вказувати ідентифікатор класу, якому належить даний член.

Завдання (Варіант 19) -

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.

19-Ліхтарик з електрошокером   
Класс 1 - CShockingLiter

Класс 2 - CElectroshocker

Класс 3 - CFlashlight

Класс 4 - CDevice

Код для вирішення задачі даного варіанту -

#include <iostream>

using namespace std;

class CDevice

{

public:

virtual void state(bool whatState) {

if (whatState) {

cout << "Device is on." << endl;

}

else {

cout << "Device is off" << endl;

}

}

void setID(int myID) {

id = myID;

}

int id;

};

class CFlashlight : public CDevice {

public:

CFlashlight() {

power = 1;

}

CFlashlight(int pow) {

power = pow;

}

void operator = (CFlashlight torch) {

this->power = torch.power;

}

void lightPower(int pow) {

power = pow;

cout << "Power is set." <<endl;

}

protected:

int power;

};

class CElectroShocker : public CDevice {

public:

CElectroShocker() {

timeUsed = 0;

}

~CElectroShocker() {

//null

}

void operator = (CElectroShocker shocker1) {

this->timeUsed = shocker1.timeUsed;

}

void shock(int time) {

cout << "Shocking for " << time << " seconds" << endl;

timeUsed += time;

cout << "Total usage time - " << timeUsed << endl;

}

protected:

int timeUsed;

};

class CShockingLighter : public CFlashlight, public CElectroShocker {

public:

CShockingLighter() {

timeUsed = 0;

power = 0;

}

virtual void state(bool whatState) override {

if (whatState) {

cout << "ShockingLighter is on." << endl;

}

else {

cout << "ShockingLighter is off" << endl;

}

}

};

int main()

{

CShockingLighter my1;

my1.state(true);

my1.lightPower(10);

my1.shock(5);

}

Скріншот виконання програми –



Висновок : в даній лабораторній роботі я опрацював роботу з спадкуванням, віртуальними функціями, базовими классами, множинним спадкуванням, застосуванням їх і використанням віртуальних функцій.