Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра “Електронних обчислювальних машин”



**Звіт з лабораторної роботи №7**

на тему:

**“ Поліморфізм”**

**Виконала:**

ст. гр. КІ-15

Аркавенко Л.С.

**Перевірив:**

асист. каф. ЕОМ

Козак Н. Б.

Львів – 2020

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

**Теоретична частина:**

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. У загальному випадку синтаксис множинного спадкування має вигляд:

class Ім'яПохідногоКласу : [модифікатор] Ім'яБазовогоКласу1, ..., [модифікатор] Ім'яБазовогоКласуN {тіло класу};

Розглянемо узагальнений приклад множинного спадкування:

class A {оголошення класу};

class B { оголошення класу };

class C : public A { оголошення класу };

class D: public B, public C { оголошення класу }; // аналогічно оголошенню class D: public B, C,{оголошення класу };

**Індивідуальне 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.



**Хід роботи:**

**Код програми:**

Файл CTVVideo.h

#pragma once

#ifndef b

#include "Cdevice.h"

#include <iostream>

#include <string>

using namespace std;

class CTV : public Cdevice {

int \*number\_of\_channels = new int;

string channel\_name[50];

int channels\_now = 0;

public:

CTV(int namber) {

\*number\_of\_channels = namber;

}

CTV() {

channels\_now = 0;

\*number\_of\_channels = 2;

channel\_name[0] = "Pershuy";

channel\_name[1] = "1+1";

}

~CTV()

{

delete number\_of\_channels;

delete[] channel\_name;

}

void Set\_Сhannel\_list()

// встановленн всіх каналів вибірково

{

if (\*number\_of\_channels == 2) {

string \*vubir = new string;

cout << "If you want to change the standard number of channels, enter 1: ";

cin >> \*vubir;

if (\*vubir == "1") {

cout << "Enter number of channels: ";

cin >> \*number\_of\_channels;

}

delete vubir;

}

A: cout << "Enter the channel number you want to record (maximum channal " << \*number\_of\_channels << "): ";

int i;

cin >> i;

if (i > \*number\_of\_channels)

goto A;

cout << "If you do not want to install a channel on this number then enter 'N'" << endl;

cout << "Enter name channel number " << i << ": ";

cin >> channel\_name[i - 1];

if (channel\_name[i - 1] == "N") {

channel\_name[i - 1] = "//Сhannel not installed\\";

}

cout << "If you want to continue installing channels then enter 1: ";

cin >> i;

if (i == 1)

goto A;

else cout << "Installation of canals is complete!" << endl;

}

void Get\_channel\_now()

// демонстрація обраного каналу

{

cout << "Now the plugged in channel: " << channel\_name[channels\_now] << endl;

cout << "By order: " << channels\_now + 1;

}

void Get\_Сhannel\_list()

// демонстрація всіх каналів

{

for (int i = 0; i < \*number\_of\_channels;i++)

if (channels\_now == i)

cout << i + 1 << ": " << channel\_name[i] << " " << "-Now turned on-" << endl;

else cout << i + 1 << ": " << channel\_name[i] << endl;

}

void Select\_Channel()

// становлення поточного каналу

{

cout << "Choose a channel: ";

cin >> channels\_now;

channels\_now--;

}

};

#endif

Файл CTV.h

#include <string>

#include <iostream>

#ifndef a

using namespace std;

class Cdevice {

int \*sound\_level = new int;

bool on = false;

public: Cdevice() {

\*sound\_level = 50;

}

Cdevice(int level) {

\*sound\_level = level;

}

~Cdevice() {

delete sound\_level; on = false;

}

void Set\_sound\_level() // встановлення рівеня звуку

{

int \*level = new int;

cout << "Enter sound level: ";

cin >> \*level;

if (\*level >= 0 && \*level <= 100)

\*sound\_level = \*level;

if (\*level < 0)

\*sound\_level = 0;

if (\*level > 100)

\*sound\_level = 100;

}

void Get\_sound\_level() //демонстрація рівня звуку

{

cout << "Now the sound level is: " << \*sound\_level << endl;

}

void On\_Off\_Device() // вкл / викл девайсу

{

if (on == false)

{

on = true;

cout << "Devic is on!" << endl;

}

else

{

on = false;

cout << "Devic is off!" << endl;

}

}

void Restart\_divace() {

delete sound\_level;

on = false;

}

};

#endif

Файл CvideoPlayer.h

#pragma once

#ifndef d

#include <string>

#include "Cdevice.h"

#include <iostream>

using namespace std;

class CvideoPlayer : public Cdevice

{

double \*flowing\_browsing\_time = new double;

public:

CvideoPlayer() {

\*flowing\_browsing\_time = 10;

}

CvideoPlayer(double time) {

\*flowing\_browsing\_time = time;

}

~CvideoPlayer() {

delete flowing\_browsing\_time;

}

void Set\_current\_time(double videotime) {

C: double time;

cout << "Rewind time: ";

cin >> time;

if (videotime > time) {

\*flowing\_browsing\_time = \*flowing\_browsing\_time + time;

}

else {

cout << "The film continues: " << videotime << "You entered: " << time;

goto C;

}

}

void Get\_flowing\_browsing\_time() {

\*flowing\_browsing\_time = 10; cout << "Current movie viewing time: " << \*flowing\_browsing\_time << endl;

}

void rewind\_forward() {

flowing\_browsing\_time = flowing\_browsing\_time + 1;

}

void rewind\_forward(double time) {

\*flowing\_browsing\_time = \*flowing\_browsing\_time + time;

}

void To\_throw\_back() { flowing\_browsing\_time = flowing\_browsing\_time - 1; }

void To\_throw\_back(double time) { \*flowing\_browsing\_time = \*flowing\_browsing\_time + time; }

};

#endif

Файл Main.cpp

#include "CTVVideo.h"

#include <iostream>

int main() {

CTVVideo a;

a.Set\_videotime(45.45);

a.Set\_namevideo("Narodnuy");

a.Set\_Videoformat();

a.Get();

a.Set\_Сhannel\_list();

a.Select\_Channel();

a.Get\_channel\_now();

a.Get\_Сhannel\_list();

a.Set\_current\_time(24.5);

a.rewind\_forward();

a.To\_throw\_back();

a.Get\_flowing\_browsing\_time();

CvideoPlayer b;

b.On\_Off\_Device();

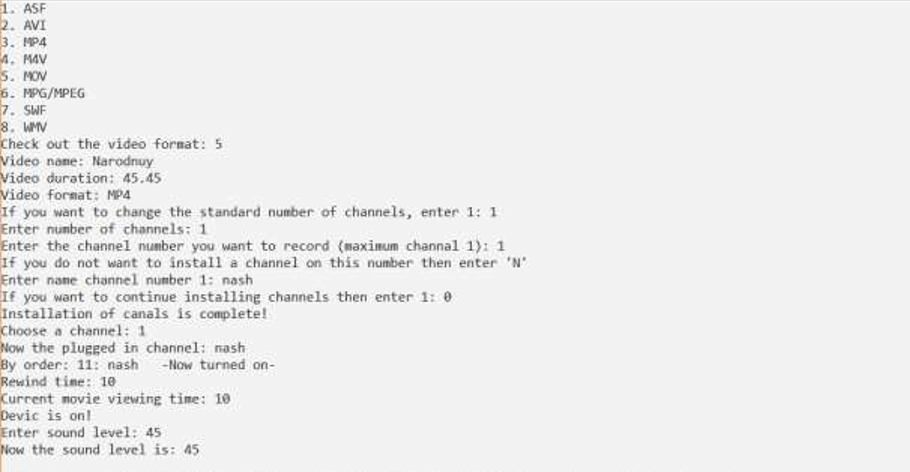
b.Set\_sound\_level();

b.Get\_sound\_level();

b.Restart\_divace();

return 0;

}

****Рис 1. Програма та консольний вивід до завдання