Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт

З лабораторної роботи № 7, варіант 8

З дисципліни « ПРОГРАМУВАННЯ, ЧАСТИНА 2 (ОБ’ЄКТНООРІЄНТОВАНЕ ПРОГРАМУВАННЯ) »

Виконав: ст. гр. КІ-15

Гербей О.М.

Перевірив: викладач

Козак Н.Б

Львів – 2020

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:

// main.cpp

#include "caudiovideoplayer.h"

int main()

{

cout << "-----------Static polymorphism--------------------" << endl;

AudioVideoPlayer a;

a.SetVolume(25);

a.SetSong("Rammstein");

a.SetVideoName("MR. Robot");

a.SetVoltage(20);

a.SetModel("LG");

a.SetPrice(4500);

a.SetCountry("USA");

a.TurnOn();

AudioVideoPlayer b = a;// реалізація перегрузки= і самого динамічногшо поліморфізму через вказівник на базовий клас

b.SetVolume(150);

b.SetSong("Slipknot");

b.SetVideoName("Black Mirror");

cout << "--------------------Dynamic polymorphism-------------------" << endl;

AudioVideoPlayer \*action;

action = &b;// цьому вказівнику присвоюємо ссилку на об'єкт b

action->CheckVolume();// і викликаємо безполяредньо метод грати музику,тобто уникаємо зайвих дій і робим програму гнучкішою

}

//cdevice.h

#include <iostream>

#include<string>

#include"time.h"

using namespace std;

class Device {// базовий клас від якого наслідуєм класи 2,3

protected:

int \*voltage;// напруга

string model;

public:

Device() {

voltage = new int(0);

}

virtual~Device() {

delete voltage;

}

Device(const Device&object) {// конструктор копіювання,викликавши його ми копіюємо дані з попереднього об'єкту в новий

this->model = object.model;//і потім юзаєм оператор присвоєннф

this->voltage = new int(\*object.voltage);

}

virtual void TurnOn() = 0;//віртуальний метод

virtual void CheckVolume() {//для реалізації динамічного поліморфізму нам треба переориділити ті методи,що будуть юзатись в класах 2 і 3\

}

virtual void PlayVideo() {

}

void SetModel(string Model) {

this->model = Model;

}

string GetModel() {

return model;

}

void SetVoltage(int Voltage) {

voltage = new int(Voltage);

}

int GetVoltage() {

return \*voltage;

}

Device&operator=(const Device&obj) {//перегрузка =

this->model = obj.model;

this->voltage = new int(\*obj.voltage);

return \*this;

}

};

//secondary.h

#include "cdevice.h"

class AudioPlayer : virtual public Device// клас сканнер,унасідуємо від 1 класу+ пишем virtual шоб могти переоприділити віртуальний метод

{

protected:

int volume;

string song;

public:

void SetVolume(int volume) {

this->volume = volume;

}

int GetVolume() {

return volume;

}

void SetSong(string song) {

this->song = song;

}

string GetSong() {

return song;

}

void CheckVolume() {

cout << "Я Програю Пісню " << GetSong() << endl;

if (volume < 5) {

ChangeVolumeLevel();

}

if (volume > 20) {

ChangeVolumeLevel();

}

}

void ChangeVolumeLevel() {

if (volume < 5) {

cout << "Рівень гучності невеликий. Ви хочете збільшити його?(yes/no)" << endl;

string choose;

cin >> choose;

if (choose == "yes") {

int level;

cout << "Введіть значення збільшення?" << endl;

cin >> level;

volume += level;

cout << "Ваш поточний рівень:" << volume << endl;

}

}

if (volume > 20) {

string choose;

cout<<"Рівень гучності великий. Це дуже небезпечно для вашого здоров'я. Ви хочете зменшити його?(yes/no)"<<endl;

cin >> choose;

if (choose == "yes") {

int level;

cout << "Введіть значення зменшення?" << endl;

cin >> level;

volume -= level;

cout << "Ваш поточний рівень гучності:" << volume << endl;

}

}

}

AudioPlayer&operator=(const AudioPlayer&obj) {//перегрузка =

this->model = obj.model;

this->voltage = new int(\*obj.voltage);

this->volume = obj.volume;

this->song = obj.song;

return \*this;

}

};

class VideoPlayer : virtual public Device// та сама суть шо і в 2 класі

{

protected:

string NameOFVideo;

int quality;

public:

void SetVideoName(string name) {

this->NameOFVideo = name;

}

string GetVideoName() {

return NameOFVideo;

}

void SetQuality() {

this->quality =quality;

cout << "Введіть якість відео: " << endl;

cin >> quality;

PlayVideo();

}

int GetQuality() {

return quality;

}

void PlayVideo() {

cout << "Зараз відтворюється відео: " << GetVideoName() << endl;

if (quality < 480) {

srand(time(0));

cout << "Відео низької якості, бажаєте збільшити якість відео?(yes/no)" << endl;

string choose;

cin >> choose;

if (choose == "yes") {

quality += (rand() % 1000);

cout << "Поточна якість відео: " << quality << endl;

}

}

if (quality >720 ) {

cout << "Рівень якості хороший. Насолоджуйтесь! " << endl;

}

}

VideoPlayer&operator=(const VideoPlayer&obj) {//перегрузка =

this->model = obj.model;

this->voltage = new int(\*obj.voltage);

this->NameOFVideo = obj.NameOFVideo;

this->quality = obj.quality;

return \*this;

}

};

//caudiovideoplayer.h

#include "secondary.h"

class AudioVideoPlayer : public AudioPlayer, public VideoPlayer {//тут наслідуємо клас 2 і 3

private:

int price;

string country;

public:

void SetCountry(string name) {

this->country = name;

}

string GetCountry() {

return country;

}

void SetPrice(int cost) {

this->price = cost;

}

int GetPrice() {

return price;

}

void TurnOn() //тут переоприділяем метод,але можна було переоприділяти в класах 2,3 залежно від реалізації

// сама суть реалізації того,якщо напруга дорівнює 0,то по суті програма не працює далі

{

if (!GetVoltage()) {

cout << "Без електрики" << endl;

}

else MultiFunctoinal();

}

void Show(){

cout<<"Модель: "<<GetModel()<<endl;

cout<<"Країна виробник: "<<GetCountry()<<endl;

cout<<"Ціна: "<<GetPrice()<<endl;

}

void MultiFunctoinal() {

cout << "Я пристрій, моя модель " << GetModel() << endl;

Continue:

cout << "Що ви хочете зробити?(listen/watch/show)" << endl;

string choose;

cin >> choose;

if (choose == "listen") {

CheckVolume();

}

if (choose == "watch"){

SetQuality();

}

if (choose == "show"){

Show();

}

cout << "Хочете робити ще?(yes,no)" << endl;

string desition;

cin >> desition;

if (desition == "yes") {

goto Continue;

}

}

AudioVideoPlayer&operator=(const AudioVideoPlayer&obj) {//перегрузка =

this->model = obj.model;

this->voltage = new int(\*obj.voltage);

this->volume = obj.volume;

this->song = obj.song;

this->NameOFVideo = obj.NameOFVideo;

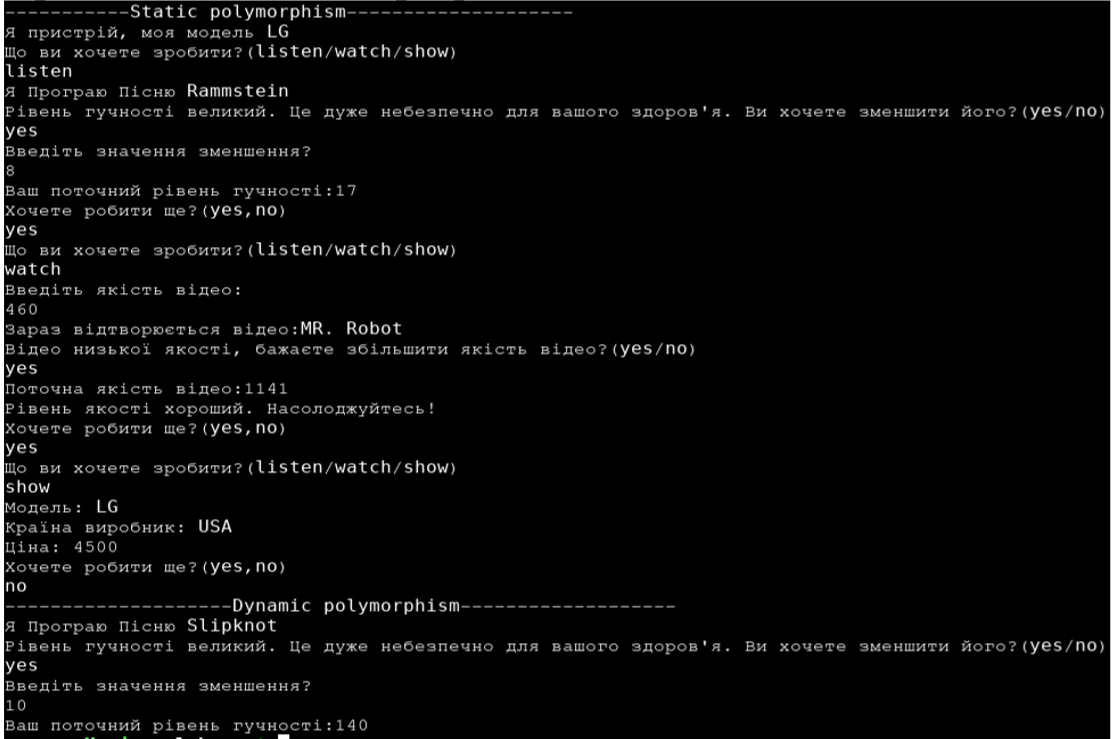
this->quality = obj.quality;

return \*this;

}

};

Приклад виконання програми:



Висновок: впродовж цієї лабораторної роботи я ознайомився із множинним спадкуванням класів та поліморфізмом.