Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»



Звіт

З лабораторної роботи № 7, варіант 23

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

На тему: «Множинне спадкування. Поліморфізм»

Виконав: ст. гр. КІ-15

Пилипів Р. С.

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 присвоєння з метою його коректної роботи.
4. Кожен з класів має містити мінімум одну властивість і 4 методи.
5. Написати main() функцію де створити об‘єкт класу 1 і продемонструвати різницю між статичним і динамічним поліморфізмом.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Варіант | Завдання | | | | |
| Предметна  область | 1 клас | 2 клас | 3 клас | 4 клас |
| 23 | Колонка з флешплеєром | CMixerSpeaker | CFlashPlayer | CLoudSpeaker | CDevice |

**Код програми:**

//CDevice.h

#include <stdlib.h>

#include <cstring>

#include <iostream>

using namespace std;

#ifndef Device

#define Device

class CDevice

{

protected:

int num\_Dev;

string\* str\_Dev;

public:

CDevice();

CDevice(string name);

CDevice(const CDevice& a);

virtual ~CDevice();

int Get();

void Set(int);

void ShowProperties();

virtual void har() = 0;

CDevice& operator=(const CDevice&);

};

#endif

//CLoudSpeaker.h

#include "CDevice.h"

#ifndef LSpeaker

#define LSpeaker

class CLoudSpeaker : virtual public CDevice

{

protected:

string\* str\_LS;

public:

CLoudSpeaker();

CLoudSpeaker(string name, string modelLS);

CLoudSpeaker(const CLoudSpeaker& a);

virtual ~CLoudSpeaker();

void ShowProperties();

virtual void har();

CLoudSpeaker& operator= (CLoudSpeaker&);

};

#endif

//CFlashPlayer.h

#include "CDevice.h"

#ifndef FPlayer

#define FPlayer

class CFlashPlayer : virtual public CDevice

{

protected:

string\* str\_FP;

public:

CFlashPlayer();

CFlashPlayer(string name, string modelFP);

CFlashPlayer(const CFlashPlayer& a);

virtual ~CFlashPlayer();

void ShowProperties();

virtual void har();

CFlashPlayer& operator= (CFlashPlayer&);

};

#endif

//CMixerSpeaker.h

#include "CFlashPlayer.h"

#include "CLoudSpeaker.h"

#ifndef MX

#define MX

class CMixerSpeaker : public CLoudSpeaker, CFlashPlayer

{

public:

CMixerSpeaker();

CMixerSpeaker(string name, string modelLS, string modelFP);

CMixerSpeaker(const CMixerSpeaker& a);

~CMixerSpeaker();

void ShowProperties();

virtual void har();

CMixerSpeaker& operator= (CMixerSpeaker&);

};

#endif

//CDevice.cpp

#include "CDevice.h"

CDevice::CDevice() : str\_Dev(new string), num\_Dev(0)

{

cout << "Device default constructor" << endl;

str\_Dev = new string;

\*str\_Dev = "No name";

cout << "====================" << endl << endl;

}

CDevice::CDevice(string name)

{

cout << "Device constructor" << endl;

str\_Dev = new string;

\*str\_Dev = name;

cout << "====================" << endl << endl;

}

CDevice::CDevice(const CDevice& a)

{

cout << "Device copy constructor" << endl;

str\_Dev = a.str\_Dev;

cout << "====================" << endl << endl;

}

CDevice::~CDevice()

{

cout << "Device destructor" << endl;

delete str\_Dev;

cout << "====================" << endl << endl;

}

int CDevice::Get()

{

return num\_Dev;

}

void CDevice::Set(int a)

{

num\_Dev = a;

cout << "====================" << endl << endl;

}

void CDevice::ShowProperties()

{

cout << "I have no properties" << endl;

cout << "====================" << endl;

}

CDevice& CDevice::operator=(const CDevice& a)

{

cout << "Device operator =" << endl;

delete str\_Dev;

str\_Dev = a.str\_Dev;

return \*this;

}

//CLoudSpeaker.cpp

#include "CLoudSpeaker.h"

CLoudSpeaker::CLoudSpeaker() : CDevice()

{

cout << "Loud Speaker default constructor" << endl;

str\_LS = new string;

\*str\_LS = "No name";

cout << "====================" << endl << endl;

}

CLoudSpeaker::CLoudSpeaker(string name, string modelLS): CDevice(name)

{

cout << "Loud Speaker constructor " << endl;

str\_LS = new string;

\*str\_LS = modelLS;

cout << "====================" << endl << endl;

}

CLoudSpeaker::CLoudSpeaker(const CLoudSpeaker& a) : CDevice(a)

{

cout << "Loud Speaker copy constructor" << endl;

str\_LS = a.str\_LS;

cout << "====================" << endl << endl;

}

CLoudSpeaker::~CLoudSpeaker()

{

cout << "Loud Speaker destructor" << endl;

delete str\_LS;

cout << "====================" << endl << endl;

}

void CLoudSpeaker::ShowProperties()

{

cout << "My model is " << str\_LS << endl;

cout << "====================" << endl << endl;

}

void CLoudSpeaker::har()

{

cout << "My characteristics " << num\_Dev << " max power" << endl;

cout << "====================" << endl << endl;

}

CLoudSpeaker& CLoudSpeaker::operator=(CLoudSpeaker& a)

{

cout << "Loud Speaker operatotor =" << endl;

delete str\_LS;

str\_LS = a.str\_LS;

CDevice::operator=(a);

return \*this;

}

//CFlashPlayer.cpp

#include "CFlashPlayer.h"

CFlashPlayer::CFlashPlayer() : CDevice()

{

cout << "Flash Player default constructor" << endl;

str\_FP = new string;

\*str\_FP = "No name";

cout << "====================" << endl << endl;

}

CFlashPlayer::CFlashPlayer(string name, string modelFP): CDevice(name)

{

cout << "Flash Player constructor " << endl;

str\_FP = new string;

\*str\_FP = modelFP;

cout << "====================" << endl << endl;

}

CFlashPlayer::CFlashPlayer(const CFlashPlayer& a) : CDevice(a)

{

cout << "Flash Player copy constructor" << endl;

str\_FP = a.str\_FP;

cout << "====================" << endl << endl;

}

CFlashPlayer::~CFlashPlayer()

{

cout << "Flash Player destructor" << endl;

delete str\_FP;

cout << "====================" << endl << endl;

}

void CFlashPlayer::ShowProperties()

{

cout << "My model is " << str\_FP << endl;

cout << "====================" << endl << endl;

}

void CFlashPlayer::har()

{

cout << "My characteristics " << num\_Dev << " max power" << endl;

cout << "====================" << endl << endl;

}

CFlashPlayer& CFlashPlayer::operator=(CFlashPlayer& a)

{

cout << "Flash Player operatotor =" << endl;

delete str\_FP;

str\_FP = a.str\_FP;

CDevice::operator=(a);

return \*this;

}

//CMixerSpeaker.cpp

#include "CMixerSpeaker.h"

CMixerSpeaker::CMixerSpeaker() : CDevice(), CLoudSpeaker(), CFlashPlayer()

{

cout << "Mixer Speaker default constructor" << endl;

cout << "====================" << endl << endl;

}

CMixerSpeaker::CMixerSpeaker(string name, string modelLS, string modelFP) : CDevice(name), CLoudSpeaker(name, modelLS), CFlashPlayer(name, modelFP)

{

cout << "Mixer Speaker constructor " << endl;

cout << "====================" << endl << endl;

}

CMixerSpeaker::CMixerSpeaker(const CMixerSpeaker& a) : CDevice(a), CLoudSpeaker(a), CFlashPlayer(a)

{

cout << "Mixer Speaker copy constructor" << endl;

cout << "====================" << endl << endl;

}

CMixerSpeaker::~CMixerSpeaker()

{

cout << "Mixer Speaker destructor" << endl;

cout << "====================" << endl << endl;

}

void CMixerSpeaker::ShowProperties()

{

cout << "My model is : Speakers " << str\_LS << " and Flash Player" << str\_FP << endl;

cout << "====================" << endl << endl;

}

void CMixerSpeaker::har()

{

cout << "My characteristics " << num\_Dev << " max power" << endl;

cout << "====================" << endl << endl;

}

CMixerSpeaker& CMixerSpeaker::operator=(CMixerSpeaker& a)

{

cout << "Mixer Speaker operatotor =" << endl;

CLoudSpeaker::operator=(a);

CFlashPlayer::operator=(a);

return \*this;

}

//main.cpp

#include "CMixerSpeaker.h"

#include <string>

int main(void)

{

cout << "----- Creating object -----" << endl;

cout << "First:" << endl;

CLoudSpeaker a1("Sven", "ST-134");

a1.Set(90);

a1.har();

a1.ShowProperties();

cout << "Second:" << endl;

CFlashPlayer a2("AUX", "3L");

a2.Set(45);

a2.har();

a2.ShowProperties();

cout << "Third:" << endl;

CMixerSpeaker a3("JBL", "345", "my");

a3.Set(50);

a3.har();

a3.ShowProperties();

cout << "Fourth" << endl;

CDevice\* a4;

a4 = &a3;

a4->Set(50);

a4->har();

a4->ShowProperties();

}

**Вікно результату:**



