Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ

**ЛАБОРАТОРНА РОБОТА № 6.**

**Спадкування**

**Варіант 16**

Виконав: ст.гр. КІ-15

Леземезюк Р.Т.

Прийняв: асистент

Козак Н.Б.

Львів – 2020

***Мета: познайомитися із спадкуванням класів.***

***ТЕОРЕТИЧНІ ВІДОМОСТІ***

***Спадкування (ієрархія "іs a")***

***Спадкування – це механізм, за допомогою якого один клас може одержувати атрибути та функціональність іншого. Спадкування дозволяє створювати ієрархію класів. При створенні нового класу, що повністю дублює існуючий клас і дещо розширяє його новими властивостями і функціональністю програміст може не дублювати існуючий клас і дописувати в нього розширену функціональність, а вказати, що новий клас є спадкоємцем елементів попередньо визначеного класу і визначити у ньому лише необхідну нову функціональність. В цьому випадку існуючий клас, функціональність якого розширюється у новому класі, називається базовим класом (base class). Новостворений клас називається похідним класом (derived class), або спадкоємцем. Кожен похідний клас може бути використаним у ролі базового класу для майбутніх похідних класів створюючи при цьому дерево спадкування, яке ще називають ієрархією спадкування класів (class hierarchy). Спадкування прийнято відображати у вигляді графу (дерева) у напрямку зверху-вниз. При цьому клас, що є у самому верху є самим першим базовим класом і називається кореневим класом або коренем дерева спадкування класів. Похідний клас, через проміжний, може наслідувати характеристики базового класу. У цьому випадку говорять, що базовий клас є непрямим базовим класом (indirect base class) для похідного. Зокрема, корінь дерева наслідувань є непрямим базовим класом для усіх класів, які знаходяться нижче першого рівня ієрархії. Клас, який При одиночному спадкуванні (single inheritance) клас породжується одним базовим класом. При множинному спадкуванні (multiple inheritance) похідний клас успадковує властивості декількох базових класів, причому можлива ситуація коли один базовий клас буде успадкований кілька разів по кількох гілках. При створенні об'єкта похідного класу в пам'яті зберігаються копії усіх класів, які становлять вітку, що породила даний клас.***

**Код програми**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <string.h>

using namespace std;

class Device

{

public:

Device(const char\* set\_Model)

{

Model = new char[strlen(set\_Model) + 1];

strcpy(Model, set\_Model);

Name = 0;

}

virtual ~Device()

{

if (Name != 0)

{

delete[] Name;

}

if (Model != 0)

{

delete[] Model;

}

}

virtual char\* Mode() = 0;

virtual void setMode(bool k) = 0;

virtual bool Status()

{

return IsWork;

}

public:

char\* Name;

char\* Model;

bool IsWork = 0;

};

class Recorder : public Device

{

public:

Recorder(const char\* set\_Model, const char\* set\_Name, int memory, string audioFormat) : Device(set\_Model)

{

IsWork = 0;

Name = new char[strlen(set\_Name) + 1];

strcpy(Name, set\_Name);

this->Memory = memory;

this->audioFormat = audioFormat;

}

virtual ~Recorder()

{

}

virtual char\* Mode()

{

char\* curMode = new char[20];

if (mode == 0)

{

strcpy(curMode, "Pause");

}

else

{

strcpy(curMode, "Play");

}

return curMode;

}

virtual void setMode(bool k)

{

mode = k;

}

virtual int Get\_Memory()

{

return Memory;

}

virtual string Get\_Format()

{

return audioFormat;

}

private:

int Memory = 0;

string audioFormat;

bool mode = 0;

};

int main()

{

Recorder Rec("Samsung", "Rec1", 1024, "mp3");

Rec.setMode(1);

cout << "Model: " << Rec.Model << endl;

cout << "Name: " << Rec.Name << endl;

cout << "Memory amount: " << Rec.Get\_Memory() << endl;

cout << "Current mode: " << Rec.Mode() << endl;

cout << "Audio format: " << Rec.Get\_Format() << endl;

system("pause");

return 0;

}

**Результат**

