Міністерство освіти і науки України

Національний університет „Львівська політехніка”

Кафедра ЕОМ

**ЛАБОРАТОРНА РОБОТА № 7.**

**Множинне спадкування. Поліморфізм**

Варіант 16

Виконав: ст.гр. КІ-15

Леземезюк Р.Т.

Прийняв: асистент

Козак Н.Б.

Львів – 2020

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

***ТЕОРЕТИЧНІ ВІДОМОСТІ***

***Множинне спадкування***

***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 -;***

***Відповідна схема утвореної ієрархії класів матиме наступний вигляд (див. рис. 7.1).***

***Як видно з даної схеми клас В є самостійним класом, клас А успадковується одинарно класом С, а клас D множинно успадковує класи В і С, прицьому клас D, опосередковано через клас С успадковує клас А. Тож при створенні об‘єкту класу D він міститиме в собі всі характеристики об‘єктів класів А, В і С. При цьому оскільки кожен з об‘єктів входить в об‘єкт D один раз, то ніяких конфліктів при створенні об‘єкту D не буде.***

**Код програми**

#include <iostream>

#include <iomanip>

#include <Windows.h>

#define w 10

using namespace std;

class CHuman

{

protected:

int height, weight;

char\* Name;

public:

CHuman(const char\* Name, int height, int weight)

{

setInfo(Name, height, weight);

}

~CHuman()

{

delete[] Name;

}

void setInfo(const char\* Name, int height, int weight)

{

this->Name = new char[strlen(Name) + 1];

strcpy\_s(this->Name, strlen(Name) + 1, Name);

this->height = height;

this->weight = weight;

}

virtual void PrintInfo() = 0;

};

class CDiver : public CHuman

{

protected:

int oxygen;;

public:

CDiver(const char\* Name, int height, int weight, int oxygen) : CHuman(Name, height, weight)

{

setInfo(oxygen);

}

void setInfo(int oxygen)

{

this->oxygen = oxygen;

}

virtual void PrintInfo()

{

cout << "Diver info:" << endl;

cout << setw(w) << left << "Name: " << CHuman::Name << endl;

cout << setw(w) << left << "Height: " << CHuman::height << " sm" << endl;

cout << setw(w) << left << "Weight: " << CHuman::weight << " kg" << endl;

cout << setw(w) << left << "Oxygen: " << oxygen << endl;

}

};

class CSodier : public CHuman

{

protected:

int age;

public:

CSodier(const char\* Name, int height, int weight, int age) : CHuman(Name, height, weight)

{

setInfo(age);

}

void setInfo(int age)

{

this->age = age;

}

virtual void PrintInfo()

{

cout << "Sodier info:" << endl;

cout << setw(w) << left << "Name: " << CHuman::Name << endl;

cout << setw(w) << left << "Height: " << CHuman::height << " sm" << endl;

cout << setw(w) << left << "Weight: " << CHuman::weight << " kg" << endl;

cout << setw(w) << left << "Age: " << age << endl;

}

};

class SMilitaryDiver : public CDiver, public CSodier

{

public:

SMilitaryDiver(const char\* Name, int height, int weight, int oxygen, int age) : CDiver(Name, height, weight, oxygen), CSodier(Name, height, weight, age)

{

}

virtual void PrintInfo()

{

cout << "MilitaryDiver info:" << endl;

cout << setw(w) << left << "Name: " << CHuman::Name << endl;

cout << setw(w) << left << "Height: " << CHuman::height << " sm" << endl;

cout << setw(w) << left << "Weight: " << CHuman::weight << " kg" << endl;

cout << setw(w) << left << "Oxygen: " << oxygen << endl;

cout << setw(w) << left << "Age: " << age << endl;

}

};

int main()

{

CDiver Boxer1("Karl", 185, 70, 1200);

Boxer1.PrintInfo();

cout << endl << endl;

CSodier Kicker1("Brain", 175, 68, 35);

Kicker1.PrintInfo();

cout << endl << endl;

SMilitaryDiver KickBoxer1("Robert", 185, 75, 300, 24);

KickBoxer1.PrintInfo();

cout << endl;

system("pause");

return 0;

}

**Результат**

