Міністерство освіти і науки України

Національний лісотехнічний університет України

Кафедра інженерії програмного забезпечення

**ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №5**

з навчальної дисципліни

«Об’єктно – орієнтоване програмування»

на тему: «Поліморфізм та віртуальні функції»

Виконав:

студент групи ІПЗС-11

Возьний С.П.

Перевірила:

Головата С. Б.

Львів – 2023

**Лабораторна робота №5**

**Тема:** Поліморфізм та віртуальні функції

**Мета:** освоїти механізм пізнього зв’язування через використання віртуальних функції при простому й множинному успадкуванні.

**Хід виконання**

**Завдання 1.** Створити ієрархію класів ОСОБА і СТУДЕНТ. У класі ОСОБА визначити віртуальну операторну функцію для зміни віку особи. Перевизначити цю функцію у класі СТУДЕНТ для переходу на наступний курс. Використати операторну віртуальну функцію для ілюстрації дії механізму пізнього зв’язування.

//Файл Person.h

#pragma once

#include <string>

using namespace std;

class Person{

string surename;

string name;

string midleName;

int year;

public:

Person(string surename, string name, string midleName, int year);

~Person();

virtual void operator+();

virtual void Show();

};

//Файл Person.cpp

#include "Person.h"

#include <iostream>

using namespace std;

Person::Person(string surename, string name, string midleName, int year) {

this->surename = surename;

this->name=name;

this->midleName = midleName;

this->year = year;

}

Person::~Person() {

cout << "Деструктор класу Person" << endl;

}

void Person::operator+(){

this->year++;

}

void Person::Show() {

cout << "Прізвище: " << surename << endl;

cout << "Ім'я: " << name << endl;

cout << "По батькові: " << midleName << endl;

cout << "Вік: " << year << endl;

}

//Файл Student.h

#pragma once

#include <string>

#include "Person.h"

using namespace std;

class Student:public Person{

int numberKours;

string nameSpeciation;

public:

Student(string surename, string name, string midleName, int year, int numberKours, string nameSpeciation);

~Student();

void operator+();

void Show();

};

//Файл Studen.cpp

#include "Student.h"

#include <iostream>

using namespace std;

Student::Student(string surename, string name, string midleName, int year, int numberKours, string nameSpeciation):Person(surename,name,midleName,year){

this->nameSpeciation = nameSpeciation;

this->numberKours = numberKours;

}

Student::~Student() {

cout << "Деструктор Student" << endl;

}

void Student::operator+() {

this->numberKours++;

}

void Student::Show() {

Person::Show();

cout << "Назва спеціальності: " << nameSpeciation << endl;

cout << "Номер курсу: " << numberKours << endl;

}

//Файл lab5.cpp

#include <iostream>

#include "Windows.h"

#include <string>

#include "Person.h"

#include "Student.h"

using namespace std;

int main(){

SetConsoleOutputCP(1251);

SetConsoleCP(1251);

Person \*p;

Person ob("Петренко", "Сергій","Степанович",45);

p = &ob;

cout << "Інформація особи до зміни віку: " << endl;

p->Show();

p->operator+();

cout << "Інформація особи після зміни віку: " << endl;

p->Show();

Student ob1("Стерео", "Оксана", "Петрівна", 22, 2, "ІПЗ");

p = &ob1;

cout << "Інформація студента до зміни віку: " << endl;

p->Show();

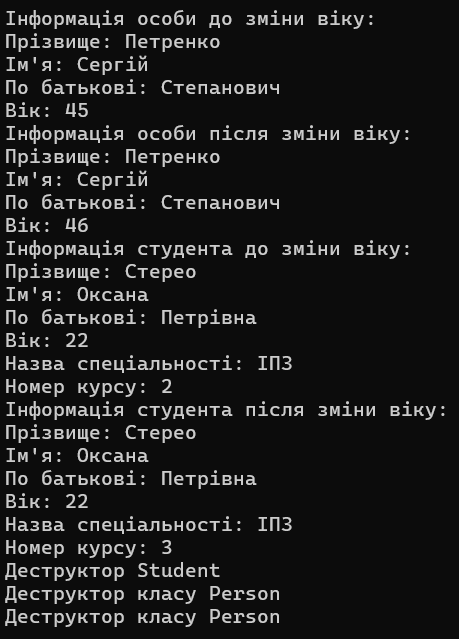
p->operator+();

cout << "Інформація студента після зміни віку: " << endl;

p->Show();

}

**Результати виконання:**



**Контрольні запитання**

1. **Чи проявляється поліморфізм при перезавантаженні операторів і функцій?**

Так, оскільки поліморфізм статичний реалізується перевантаженням функцій

1. **Чи наслідується конструктор у похідному класі?**

Ні конструктори самі не наслідуються, їх потрібно викликати в конструкторах похідних класів.

1. **Для чого використовуються віртуальні функції?**

Вони використовуються для реалізації динамічного поліморфізму.

1. **Чи можуть функції похідного класу використовувати закриті члени базового класу?**

Ні вони не можуть їх використовувати лише за допомогою відкритих методів базового класу.

1. **Чи можливе повторне визначення віртуальних функцій у похідному класі?**

Так можна це їх основою поліморфізму.

1. **Що таке повністю віртуальна функція?**

Це віртуальна функція не визначена в базовому класі і присвоюється 0 але в похідному класі вона повинна бути визначена.

1. **Які властивості мають дружні функції та дружні класи?**

Ці функції мають доступ до закритих полів класу. Вони не є членами класу. Дружність не успадковується.

1. **Що таке поліморфізм в ООП?**

Поліморфізм в об’єктно – орієнтованому проектуванні – це властивість

вказівника (або посилання) на базовий клас, проініціалізованого адресою (або іменем – для посилання) об’єкта похідного класу, викликати віртуальні методи похідного класу у межах дії поліморфічного кластера.

1. **Чи може виклик однієї і тієї ж функції з однаковими параметрами при використанні принципів ООП дати різні результати?**

Так це можливе за допомогою віртуальних функцій.

1. **В якому випадку множинного наслідування базовий клас та його функції описуються віртуальними?**

В тому випадку коли є один базовий клас від нього є два похідні і від цих двох похідних є ще однин похідний. Тут вже потрібно для уникнення дублювання самого базового класу в самому похідному і віртуальна функція залишається однозначною.

1. **Поясніть суть правила домінування імен при множинному наслідуванні.**

Якщо похідний клас успадковується від базового і має власний варіант віртуальної функції тоді її пріорітет вищий ніж в базового а якщо не визначена тоді викликатиметься сама ближча функція з базового класу