Міністерство освіти і науки України

Вінницький національний технічний університет

Факультет інформаційних технологій та комп’ютерної інженерії

Кафедра ПЗ

Лабораторна робота №6

з дисципліни «Операційні системи»

з теми «Спадчина і віртуальні функції»

Виконали: ст. 1ПІ-21Б Миронюк О.В.

Гиренко В.В.

Коцюбняк В.А.

Максименко О.В.

Перевірив: Рейда О. М.

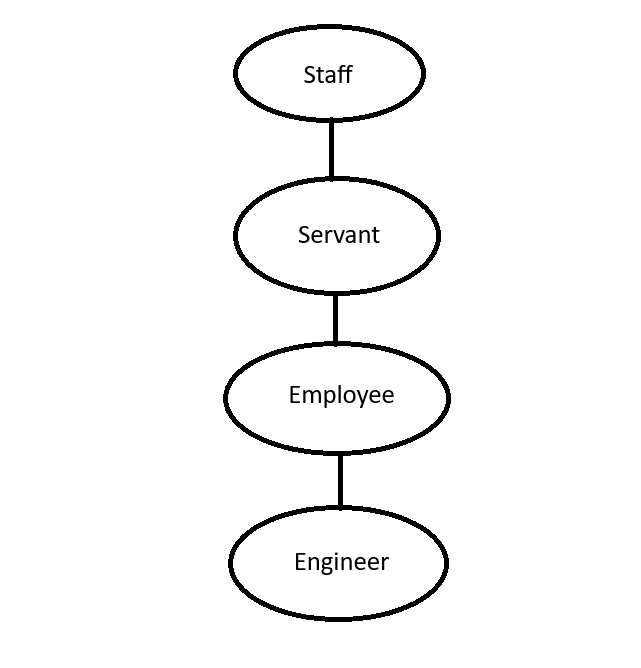
Вінниця – 2023

**Мета роботи:** Отримати практичні навички створення ієрархії класів та використання статичних компонентів класу.

**Теоретичні відомості**

**Завдання:** написати програму, у якій створюється ієрархія класів. Включити поліморфні об'єкти у пов'язаний список, використовуючи статичні компоненти класу. Показати використання віртуальних функцій.

Граф ієрархії класів, де Staff – базовий клас:



**Код програми**

**Модуль Staff.h**

#pragma once

#ifndef STAFF\_H

#define STAFF\_H

#include <string>

using namespace std;

class staff

{

protected:

staff\* next\_;

static staff\* begin\_;

public:

//Constructors

staff();

staff(const staff&);

//Destructor

virtual ~staff();

//Functions

static void show(); // Перегляд списку

void add();

virtual void print();

};

#endif

**Модуль Staff.cpp**

#include "staff.h"

#include <iostream>

//Constructors

staff::staff() = default;

staff::staff(const staff&) {}

//Destructor

staff::~staff() = default;

//Functions

void staff::add() {

staff\* s = begin\_;

while (s->next\_) {

s = s->next\_;

}

s->next\_ = this;

}

void staff::show() {

staff\* s = begin\_;

do {

s->print();

s = s->next\_;

} while (s);

}

void staff::print()

{

}

**Модуль SERVANT.h**

#pragma once

#include "staff.h"

class servant :

public staff

{

protected:

string name\_;

int age\_;

int experience\_;

public:

using staff::staff; // inherit base class constructor

servant(string, int, int);

servant(servant&);

//Properties

\_\_declspec(property (put = set\_name, get = get\_name)) string Name;

\_\_declspec(property (put = set\_age, get = get\_age)) int Age;

\_\_declspec(property (put = set\_experience, get = get\_experience)) int Experience;

//Accessors

string get\_name();

void set\_name(string name);

int get\_age();

void set\_age(int age);

int get\_experience();

void set\_experience(int experience);

//Functions

void print();

};

**Модуль SERVANT.cpp**

#include "servant.h"

#include <iostream>

//Accessors

string servant::get\_name()

{

return name\_;

}

void servant::set\_name(const string name)

{

name\_ = name;

}

int servant::get\_age()

{

return age\_;

}

void servant::set\_age(const int age)

{

age\_ = age;

}

int servant::get\_experience()

{

return experience\_;

}

void servant::set\_experience(const int experience)

{

experience\_ = experience;

}

//Constructors

servant::servant(string name = "Default\_name", int age = 18, int experience = 0)

{

name\_ = name;

age\_ = age;

experience\_ = experience;

};

servant::servant(servant& s) : servant(s.Name, s.Age, s.Experience) {}

void servant::print()

{

std::cout << "Servant: Name: " << this->Name << "; Age: " << this->Age << "; Experience: " << this->Experience << ";\n";

}

**Модуль engineer.h**

#pragma once

#include "employee.h"

class engineer :

public employee

{

public:

using employee::employee;

virtual void print();

};

**Модуль engineer.cpp**

#include "engineer.h"

#include <iostream>

void engineer::print()

{

std::cout << "Engineer: Name: " << this->Name << "; Age: " << this->Age << "; Experience: " << this->Experience << ";\n";

}

**Модуль employee.h**

#pragma once

#include "servant.h"

class employee :

public servant

{

public:

using servant::servant;

virtual void print();

};

**Модуль employee.cpp**

#include "employee.h"

#include <iostream>

void employee::print()

{

std::cout << "Employee: Name: " << this->Name << "; Age: " << this->Age << "; Experience: " << this->Experience << ";\n";

}

**Модуль main.cpp**

#include <iostream>

#include "employee.h"

#include "engineer.h"

#include "servant.h"

#include "staff.h"

using namespace std;

staff\* s\_ = new staff();

staff\* staff::begin\_ = s\_;

int main()

{

servant serv("jack", 23, 3);

employee empl ("rob", 25, 6);

engineer engin("chris", 30, 10);

serv.add();

empl.add();

engin.add();

staff::show();

}

**Результат тестування роботи програми**

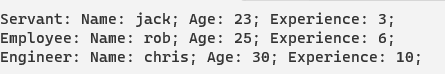


Рисунок 1.1 – Результат тесутвання програми

Висновок: ми навчились створювати програму на мові програмування С++, працювати з класами, їх ієрархією, вказівниками та віртуальними функціями.