

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 «Спадщина і віртуальні функції»

Виконали: ст. 1ПІ-21Б

Миронюк О.В.
Гиренко В.В.
Коцюбняк В.А.
Максименко О.В.

Перевірів:

Рейда О. М.

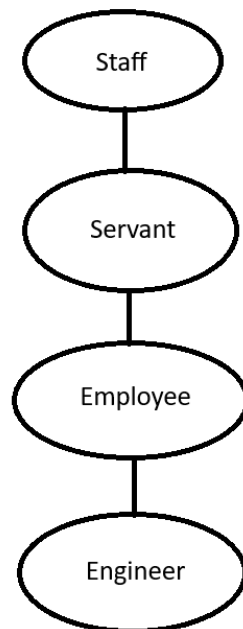
Вінниця – 2023

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

```
Servant: Name: jack; Age: 23; Experience: 3;
Employee: Name: rob; Age: 25; Experience: 6;
Engineer: Name: chris; Age: 30; Experience: 10;
```

Рисунок 1.1 – Результат тестування програми

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