МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет

«Харківський політехнічний інститут»

Факультет Комп’ютерних наук та програмної інженерії

Кафедра Інформатики та інтелектуальної власності

ЗВІТ

До лабораторной роботи №3 з дисципліни

«Об’єкто-орієнтоване програмування»

Студент Карауланов А.О

Викладач Івашко А.В.

Харків 2022

**Лабораторна робота №3**

**Тема: «Спадкування класів, абстрактні класи, інтерфейси, поліморфізм»**

**Мета роботи:** навчитись реалізовувати спадкування класів, використовувати абстрактні класи та інтерфейси під час спадкування, застосовувати принцип поліморфізму після спадкування.

**Завдання:** Окрім вашого класу додати до абстрактного класу чи інтерфейсу, створеному у попередньому пункті, ще два класи спадкоємця першого рівня. Наприклад до класу Book (Книга) можна додати класи Newspaper (Газета) Magazine (Журнал) з відповідними полями та методами.

## 1 Виконання роботи

Згідно завдання було створено класи hospital та доктора з чотирма полями даних типу int experience, roomnum, hospitalnum та string docsurname, docname, docsecname, hospital і room. Для цих класів було створено два конструктори за замовченням. А також для зручного виведення такого масиву було створено метод show. Метод data ініціалізує елементи. Приклад коду класу наведено нижче.

**class** doctor:**public** patient

{

**public**:

string docsurname;

string docname;

string docsecname;

**int** experience;

doctor();

**void** data(patient patient[], doctor doctor[], **short** **int** count);

**void** show(patient patient[], doctor doctor[], **short** **int** count);

};

**class** hospital:**public** patient{

**public**:

string hospital1;

**int** hosnum;

string room1;

**int** roomnum;

**void** data(patient patient[], hospital hospital[], **short** **int** count);

**void** show(patient patient[], hospital hospital[], **short** **int** count);

hospital();

};

Повний код наведено у додатку А.

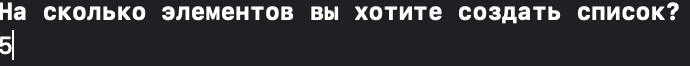


Рисунок 1.1 – Приклад запиту кількості елементів списку

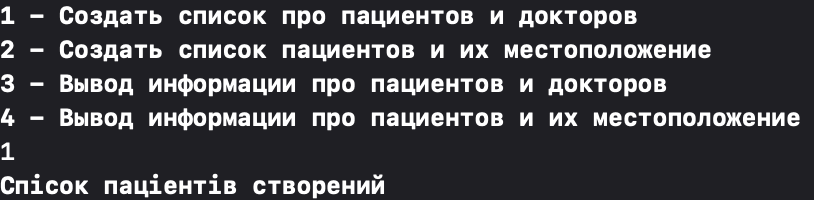


Рисунок 1.2 – Приклад створення списку з доктором

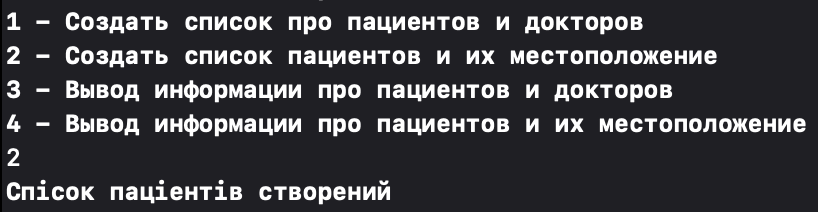


Рисунок 1.3 – Приклад створення спису з лікарнею



Рисунок 1.4 – Приклад виводу списку з доктором

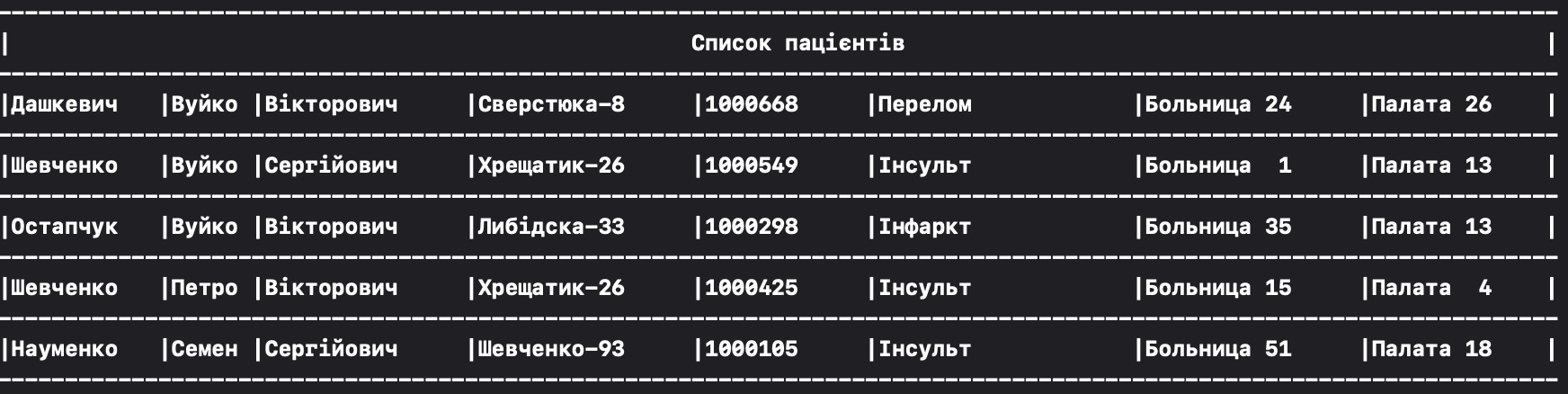


Рисунок 1.5 – Приклад виводу списку з госпіталем

**Висновок**

Під час даної лабораторної роботи було закріплено навички з наслідування класу.

**Додаток А**

Header.h

#pragma once

#include <iostream>

#include <string>

#include <iomanip>

#include <ctime>

#define N 10

**using** **namespace** std;

**class** patient{

**public**:

string surname;

string name;

string secname;

string adress;

**int** num;

string diagnosis;

patient();

**void** setsurname(string valuesurname);

string getsurname();

**void** setname(string valuename);

string getname();

**void** setsecname(string valuesecname);

string getsecname();

**void** setadress(string valueadress);

string getadress();

**void** setnum(**int** valuenum);

**int** getnum();

**void** setdiagnosis(string valuediagnosis);

string getdiagnosis();

**void** data(patient patient[], **short** **int** count);

};

**class** doctor:**public** patient

{

**public**:

string docsurname;

string docname;

string docsecname;

**int** experience;

doctor();

**void** data(patient patient[], doctor doctor[], **short** **int** count);

**void** show(patient patient[], doctor doctor[], **short** **int** count);

};

**class** hospital:**public** patient{

**public**:

string hospital1;

**int** hosnum;

string room1;

**int** roomnum;

**void** data(patient patient[], hospital hospital[], **short** **int** count);

**void** show(patient patient[], hospital hospital[], **short** **int** count);

hospital();

};

**class** Main

{

**public**:

**void** main() **const**;

**Додаток Б**

Source.cpp

#include "lab.hpp"

patient::patient(){

surname = "Шевченко";

name = "Тарас";

secname = "Григорович";

adress = "Шевченко-93";

num = 0;

diagnosis = "Перелом";

}

doctor::doctor(){

docsurname = "Шевченко";

docname = "Тарас";

docsecname = "Григорович";

}

hospital::hospital(){

hospital1 = "Больница";

hosnum = 1;

room1 = "Палата";

roomnum = 1;

}

**void** Main::main() **const**{

patient patient[N];

doctor doctor[N];

hospital hospital[N];

**short** **int** count;

cout << "На сколько элементов вы хотите создать список?" << endl;

cin >> count;

**int** button;

**while** (**true**) {

cout << "1 - Создать список про пациентов и докторов\n2 - Создать список пациентов и их местоположение\n3 - Вывод информации про пациентов и докторов\n4 - Вывод информации про пациентов и их местоположение" << endl;

cin >> button;

**if** (button == 1) {

doctor -> data(patient, doctor, count);

}

**else** **if** (button == 2){

hospital -> data(patient, hospital, count);

}

**else** **if** (button == 3){

doctor -> show(patient, doctor, count);

}

**else** **if** (button == 4){

hospital -> show(patient, hospital, count);

}

}

}

**void** doctor::data(patient patient[], doctor doctor[], **short** **int** count){

**static** string surname\_rand[] = {"Шевченко", "Дашкевич", "Остапчук", "Сидорчук", "Науменко"};

**static** string name\_rand[] = {"Сидор", "Семен", "Тарас", "Вуйко", "Петро", "Денис"};

**static** string secname\_rand[] = {"Андрійович", "Михайлович", "Сергійович", "Вікторович", "Васильвоич"};

**static** string adress\_rand[] = {"Шевченко-93", "Хрещатик-26", "Нікітіна-27", "Либідска-33", "Сверстюка-8"};

**static** string diagnosis\_rand[] = {"Перелом", "Інфаркт", "Опухоль", "Склероз", "Інсульт"};

**static** string docsurname\_rand[] = {"Шевченко", "Дашкевич", "Остапчук", "Сидорчук", "Науменко"};

**static** string docname\_rand[] = {"Сидор", "Семен", "Тарас", "Вуйко", "Петро", "Денис"};

**static** string docsecname\_rand[] = {"Андрійович", "Михайлович", "Сергійович", "Вікторович", "Васильвоич"};

**for** (**int** i = 0; i < count; i++) {

patient[i].surname = surname\_rand[rand() % 5];

patient[i].name = name\_rand[rand() % 5];

patient[i].secname = secname\_rand[rand() % 5];

patient[i].adress = adress\_rand[rand() % 5];

patient[i].num = rand() % 1000 + 1000000;;

patient[i].diagnosis = diagnosis\_rand[rand() % 5];

doctor[i].docsurname = docsurname\_rand[rand() % 5];

doctor[i].docname = docname\_rand[rand() % 5];

doctor[i].docsecname = docsecname\_rand[rand() % 5];

doctor[i].experience = rand() % 50;

}

cout << "Спісок паціентів створений" << endl;

}

**void** hospital::data(patient patient[], hospital hospital[], **short** **int** count){

**static** string surname\_rand[] = {"Шевченко", "Дашкевич", "Остапчук", "Сидорчук", "Науменко"};

**static** string name\_rand[] = {"Сидор", "Семен", "Тарас", "Вуйко", "Петро", "Денис"};

**static** string secname\_rand[] = {"Андрійович", "Михайлович", "Сергійович", "Вікторович", "Васильвоич"};

**static** string adress\_rand[] = {"Шевченко-93", "Хрещатик-26", "Нікітіна-27", "Либідска-33", "Сверстюка-8"};

**static** string diagnosis\_rand[] = {"Перелом", "Інфаркт", "Опухоль", "Склероз", "Інсульт"};

;

**for** (**int** i = 0; i < count; i++) {

patient[i].surname = surname\_rand[rand() % 5];

patient[i].name = name\_rand[rand() % 5];

patient[i].secname = secname\_rand[rand() % 5];

patient[i].adress = adress\_rand[rand() % 5];

patient[i].num = rand() % 1000 + 1000000;;

patient[i].diagnosis = diagnosis\_rand[rand() % 5];

hospital[i].hospital1 = "Больница";

hospital[i].hosnum = rand()%100;

hospital[i].room1 = "Палата";

hospital[i].roomnum = rand()%30;

}

cout << "Спісок паціентів створений" << endl;

}

**void** doctor::show(patient patient[], doctor doctor[], **short** **int** count){

cout << setw(4) <<"-----------------------------------------------------------------------------------------------------------------\n";

cout << setw(4) << "| Список пацієнтів |\n";

cout << setw(4) <<"-----------------------------------------------------------------------------------------------------------------\n";

cout << "|" << "Прізвище" << "|" << "Ім'я" << setw(3) << "|" << "По-батькові" << "|" << "Адреса"<< setw(11) << "|" << "Номер мед. картки" << "|" << "Діагноз" << "|" << "Прізвище лікаря" << "|" << "Ім'я" << setw(2) << "|" << "По-батькові" << "|" << "Досвід" << "|" << endl;

cout << setw(4) <<"-----------------------------------------------------------------------------------------------------------------\n";

**for** (**int** i = 0; i < count; i++) {

cout << "|"<<setw(10) << patient[i].surname;

cout << "|" << setw(4) << patient[i].name << setw(2);

cout << "|" << setw(6) << patient[i].secname << setw(2);

cout << "|" << setw(3) << patient[i].adress <<setw(6);

cout << "|" << setw(3) << patient[i].num <<setw(11);

cout << "|" << setw(3) << patient[i].diagnosis;

cout << "|"<<setw(10) << doctor[i].docsurname << setw(8);

cout << "|" << setw(4) << doctor[i].docname;

cout << "|" << setw(6) << doctor[i].docsecname << setw(2);

cout << "|" << setw(3) << doctor[i].experience << setw(5);

cout<<"|\n";

cout << setw(4) <<"-----------------------------------------------------------------------------------------------------------------\n";

}

}

**void** hospital::show(patient patient[], hospital hospital[], **short** **int** count){

cout << setw(4) <<"---------------------------------------------------------------------------------------------------------------------\n";

cout << setw(4) << "| Список пацієнтів |\n";

cout << setw(4) <<"---------------------------------------------------------------------------------------------------------------------\n";

**for** (**int** i = 0; i < count; i++) {

cout << "|"<<setw(10) << patient[i].surname << setw(4);

cout << "|" << setw(4) << patient[i].name << setw(2);

cout << "|" << setw(6) << patient[i].secname << setw(6);

cout << "|" << setw(3) << patient[i].adress <<setw(6);

cout << "|" << setw(3) << patient[i].num <<setw(6);

cout << "|" << setw(3) << patient[i].diagnosis <<setw(13);

cout << "|" << setw(3) << hospital[i].hospital1 <<setw(6);

cout << setw(3) << hospital[i].hosnum <<setw(6);

cout << "|" << setw(3) << hospital[i].room1 <<setw(6);

cout << setw(3) << hospital[i].roomnum <<setw(6);

cout<<"|\n";

cout << setw(4) <<"---------------------------------------------------------------------------------------------------------------------\n";

}

}