МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

КУРСОВА РОБОТА

з дисципліни «Об’єктно-орієнтоване програмування»

за темою «Створення ієрархії класів для лікарні»

|  |  |
| --- | --- |
|  | Керівник роботи: |
|  | Любченко Н.Ю. |
|  |  |
|  | Виконавець: |
|  |  |

Харків, 2021

ЗАВДАННЯ

на курсову роботу

з курсу «Об’єктно-орієнтоване програмування»

Тема: «Створення бази даних лікарні»

Постановка задачі: спроектувати та реалізувати програму графічного інтерфейсу користувача для лікарні

Необхідно створити ієрархію із 10-ти і більше класів. Виконання необхідного завдання здійснюється на основі об’єктів класів.

Інтерфейс користувача передбачає створення вікна з головним меню, необхідними кнопками, таблицями для введення та редагування даних.

Реалізація програми повинна здійснюватися з використанням об’єктно-орієнтованих технологій. Слід представити необхідну інформацію про програму та способи її застосування.

Короткий зміст роботи.

А) Реферативна частина.

Дослідження поставленої задачі. Вибір та обґрунтування необхідних програмних методів розв’язання задачі.

Б) Теоретична частина.

Визначення вимог щодо програмного забезпечення. Проектування інтерфейсу користувача. Розробка необхідних UML-діаграм.

В) Програмна частина.

Реалізація та тестування програми графічного інтерфейсу користувача.

РЕФЕРАТ

Ключові слова: CPP, наслідування класів, UML-діаграми, графічний інтерфейс, лікарня, об’єкти класу

Пояснювальна записка містить: 25 стор., 9 рис., 10 джерел.

Предмет дослідження – сучасні методи розробки програм таких, як об'єктно-орієнтоване програмування та візуальне проектування, а також структурний і модульне програмування.

Курсова робота виконується з метою закріплення, поглиблення і активного застосування знань з програмування, закріплення знань, отриманих в лекційному курсі, а також на практичних і лабораторних заняттях.

Метод дослідження – вивчення літератури, складання і налагодження програм на комп'ютері.

Об'єктивним дослідженням була розробка прикладної програми графічного інтерфейсу користувача для створення бази даних лікарні

При виконанні робіт було використано:

1) наслідування класів;

2) поліморфізм;

3) інкапсуляція;

4) роздільна компіляція;

5) створення UML-діаграми в MS Visual Studio.

ЗМІСТ

[ВСТУП 5](#_Toc89948345)

[1. Інтегровані середовища розробки 6](#_Toc89948346)

[1.1 QT Creator 6](#_Toc89948347)

[1.2 DE Eclipse 6](#_Toc89948348)

[1.3 MS Visual Studio 7](#_Toc89948349)

[1.4 Обгрунтування вибору середовища програмування 7](#_Toc89948350)

[2. Проектування програмного забезпечення 9](#_Toc89948351)

[2.1 Опис програмного забезпечення 9](#_Toc89948352)

[2.2 Створення діаграми класів в середовищі VS 10](#_Toc89948353)

[3. ОПИС ПРОГРАМНОГО ПРОДУКТУ 12](#_Toc89948354)

[3.1 Опис інтерфейсу розробленого продукту 12](#_Toc89948355)

[3.2 Тестування розробленого продукту 13](#_Toc89948356)

[Висновок 18](#_Toc89948357)

[Список використаних джерел 19](#_Toc89948358)

[ДОДАТОК А 20](#_Toc89948359)

# ВСТУП

Під час виконання курсової роботи були отримані навички створення програмного забезпечення на мові С++ з використанням Windows Forms, навички у наслідуванні класів та UML-діаграми.

Курсова робота виконується для ознайомлення з об’єктно-орієнтованими технологіями, об’єктів класів, систематизації, поглиблення і активного застосування знань з програмування, закріплення знань, отриманих в лекційному курсі, а також на практичних і лабораторних заняттях.

Мета роботи – створити додаток для роботи з базою даних у лікарні з комфортним для користувача інтерфейсом. Інтерфейс користувача передбачає створення вікна з головним меню, необхідними кнопками, таблицями для введення та редагування даних.

# 1. Інтегровані середовища розробки

Інтегроване середовище розробки, ІСP (англ. Integrated development environment – IDE ), також єдине середовище розробки, ЕСР – комплекс програмних засобів, який використовується програмістами для розробки програмного забезпечення (ПО).

Середовище розробки включає в себе:

– текстовий редактор;

– компілятор і / або інтерпретатор;

– засоби автоматизації збирання;

– відладчик.

Існує багато ІСР, в яких можна створити програму графічного інтерфейсу далі розбір деяких із них.

## 1.1 QT Creator

Qt Creator – інтегроване середовище розробки, призначене для створення крос-платформових застосунків з використанням бібліотеки Qt. Підтримується розробка як класичних програм мовою C++, так і використання мови QML, для визначення сценаріїв, в якій використовується JavaScript, а структура і параметри елементів інтерфейсу задаються CSS-подібними блоками. Qt Creator може використовувати GCC або Microsoft VC++ як компілятор і GDB як зневаджувач. Для Windows версій бібліотека комплектується компілятором, заголовними і об'єктними файлами MinGW.

## 1.2 DE Eclipse

Eclipse – вільне інтегроване середовище розробки модульних кроссплатформенних додатків. Розвивається та підтримується Eclipse Foundation.

Найбільш відомі програми на основі Eclipse Platform — різні «Eclipse IDE» для розробки програмного забезпечення на багатьох мовах (наприклад, найбільш популярний «Java IDE», який спочатку підтримувався, не покладається на будь-які закриті розширення, використовує стандартний відкритий API для доступу до Eclipse Platform ). Eclipse служить насамперед платформою для розробки розширень, чим він і завоював популярність: будь-який розробник може розширити Eclipse своїми модулями. Вже існують Java Development Tools (JDT), C/C++ Development Tools (CDT), що розробляються інженерами QNX спільно з IBM, та засоби для мов Ada (GNATbench, Hibachi), COBOL, FORTRAN, PHP, X10 (X10DT) та ін. різних розробників. Багато розширень доповнює середовище Eclipse диспетчерами для роботи з базами даних, серверами додатків та ін.

## 1.3 MS Visual Studio

Microsoft Visual Studio – серія продуктів фірми Майкрософт, які містять інтегроване середовище розробки програмного забезпечення та низку інших інструментальних засобів. Ці продукти дають змогу розробляти як консольні програми, так і програми з графічним інтерфейсом, включно з підтримкою технології Windows Forms, а також вебсайти, вебзастосунки, вебслужби як у рідному, так і в керованому кодах для всіх платформ, що підтримуються Microsoft Windows, Windows Mobile, Windows Phone, Windows CE, .NET Framework, .NET Compact Framework та Microsoft Silverlight.

## 1.4 Обгрунтування вибору середовища програмування

Багато функціональних можливостей Visual Studio спрямовані на те, щоб допомагати розробнику робити свою роботу якнайшвидше. Зручні функції, на зразок функції IntelliSense (яка вміє перехоплювати помилки та пропонувати правильні варіанти), функції пошуку та заміни (яка дозволяє відшукувати ключові слова як в одному файлі, так і у всьому проекті) та функції автоматичного додавання та видалення коментарів (яка може тимчасово приховувати блоки коду), дозволяють розробнику працювати швидко та ефективно. Запропоновані Visual Studio інструменти налагодження є найкращим засобом для відстеження помилок і діагностування дивної поведінки. Розробник може виконувати свій код за рядком за раз, встановлювати інтелектуальні точки переривання, при бажанні зберігаючи їх для використання в майбутньому, і будь-коли переглядати поточну інформацію з пам'яті. Visual Studio також має безліч інших функцій: можливість управління проектом; вбудована функція керування вихідним кодом; можливість рефакторизації коду; потужна модель розширюваності. Більше того, у разі використання Team System розробник отримує розширені можливості для модульного тестування, спільної роботи та управління версіями коду (що значно більше того, що пропонується у більш простих інструментах типу Visual SourceSafe).

# 2. Проектування програмного забезпечення

## 2.1 Опис програмного забезпечення

Програмне забезпечення створене для комфортної роботи з базою даних лікарні. Воно складається з 10 класів(hospital, cardiology, cardiologist, traumatology, traumatologist, retinology, retinologist, card\_pat, trau\_pat, ret\_pat) і чотирьох ступінчастої ієрархії (лікарня – відділення – лікарі – пацієнти).

Функції:

У класах cardiology, traumatology, retinology :get\_spec – повертає значення змінної spec.

У класах cardiologist, traumatologist, retinologist:

size\_vector – повертає розмір вектора;

add\_cardiologist – додає до вектора нового лікаря;

delete\_ cardiologist – видаляє лікаря з вектора;

add\_ traumatologist – додає до вектора нового лікаря;

delete\_ traumatologist– видаляє лікаря з вектора;

add\_ retinologist – додає до вектора нового лікаря;

delete\_ retinologist – видаляє лікаря з вектора;

get\_fio\_doc – повертає ПІБ лікаря;

get\_post – повертає посаду лікаря;

get\_ndoc – повертає номер лікаря.

У класах card\_pat, trau\_pat, ret\_pat:

get\_fio – повертає ПІБ пацієнта;

get\_phone – повертає номер телефону пацієнта;

get\_birthday – повертає дату народження пацієнта;

get\_days – повертає кількість днів на які надано лікарняний пацієнту;

get\_disp – повертає надобність стаціонару;

get\_n – повертає № лікаря для пацієнта;

get\_addition – повертає примітку пацієнта;

get\_illness – повертає діагноз пацієнта.

## 2.2 Створення діаграми класів в середовищі VS

UML (англ. Unified Modeling Language – уніфікована мова моделювання) – мова графічного опису для об'єктного моделювання в області розробки програмного забезпечення, для моделювання бізнес-процесів, системного проектування та відображення організаційних структур.

Діаграма класів (Class diagram) – статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм. Діаграма класів служить для представлення статичної структури моделі системи в термінології класів об'єктноорієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

Процес створення UML-діаграми в середовищі Microsoft Visual Studio на мові програмування С++.

1) Натиснути правою кнопкою на рішення проекту.

2) Вибрати представлення, перейти до діаграми классів.

Або можна виконати наступні дії.

1. Додати елемент діаграма класів.
2. Відкрити представлення класів.
3. Перетягнути клас на робочу область.

На рис 2.1 зображена діаграма класів для програми графічного інтерфейсу користувача для лікарні.

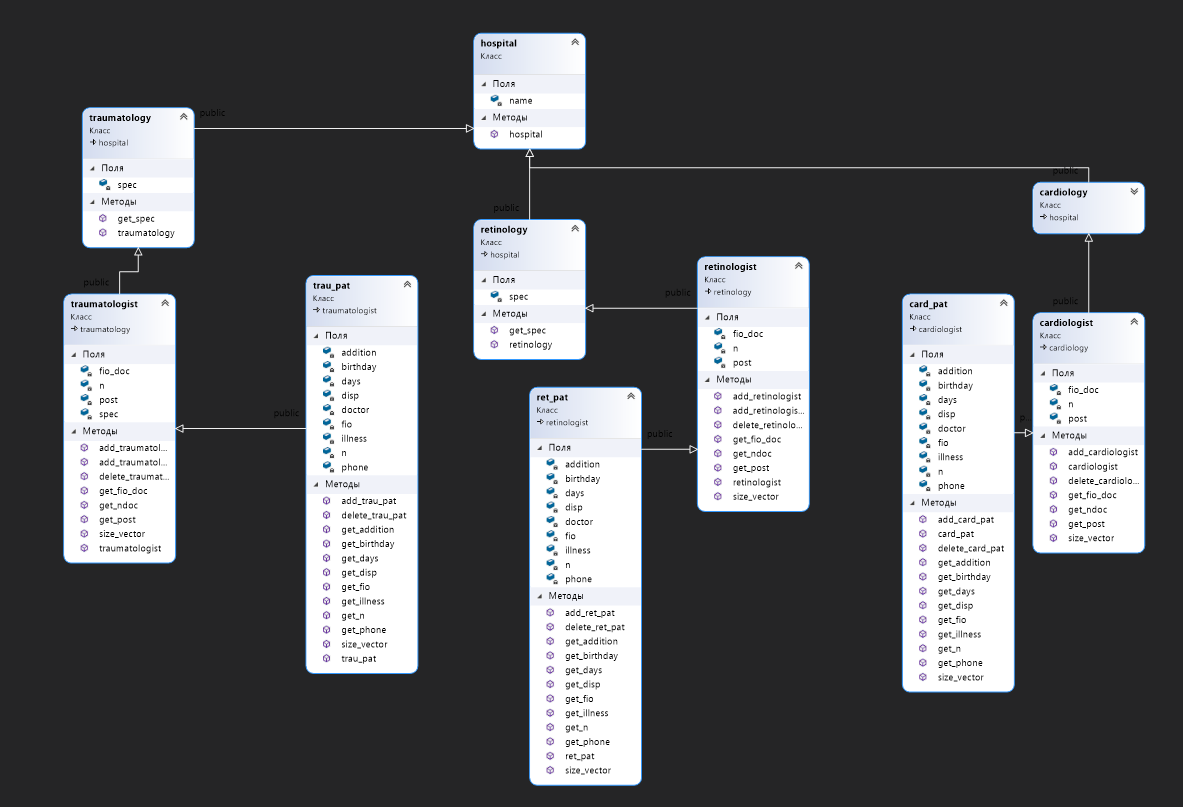


Рисунок 2.1 – Діаграма класів

# 3. ОПИС ПРОГРАМНОГО ПРОДУКТУ

## 3.1 Опис інтерфейсу розробленого продукту

На рисунку 3.1 представлений інтерфейс розробленого продукту.

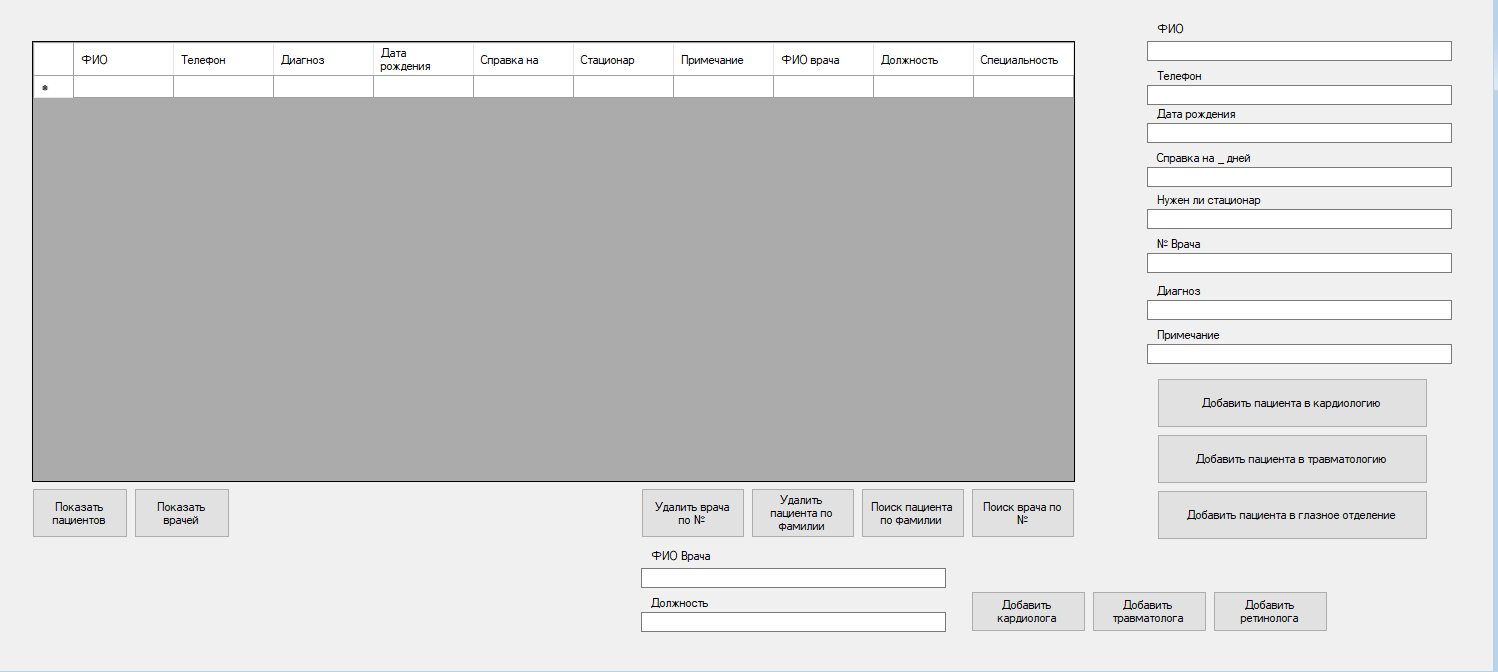


Рисунок 3.1 – Інтерфейс додатку

Інтерфейс додатку складається з наступних компонентів.

* DataGridView використовується для відображення бази даних.
* Кнопки «Показати лікарів» та «Показати пацієнтів» використовується для відображення відповідних баз даних по натисканню.
* textbox ПІБ, телефон, дата народження, довідка на \_ днів, стаціонар, № лікаря, хвороба, примітка використовується для отримання інформації про пацієнта.
* textbox ПІБ лікаря, посада використовується для отримання інформації про лікаря.
* Кнопки «Додати пацієнта в кардіологію», «Додати пацієнта в травматологію», «Додати пацієнта в очне відділення» додають пацієнтів до бази даних.
* Кнопки «Додати кардіолога», «Додати травматолога», «Додати ретинолога» додають відповідних лікарів до бази даних.
* Кнопки «Видалити лікаря по №» та «Видалити пацієнта по ПІБ» видаляють лікаря та пацієнта по відповідних полях.
* Кнопки «Пошук пацієнта по ПІБ» та «Пошук лікаря по №» відображають пацієнта або лікаря з бази даних по відповідних полях.

## 3.2 Тестування розробленого продукту

Додаток має такі функції: відображення пацієнтів, відображення лікарів, можливість додати пацієнта, можливість видалити пацієнта, можливість додати лікаря, можливість видалити лікаря, можливість знайти конкретного пацієнта, можливість знайти конкретного лікаря.

Додаток від початку не має бази даних пацієнтів. База даних лікарів заповнена, щоб побачити її потрібно натиснути на кнопку «Показати лікарів», її можна побачити на рисунку 3.2.

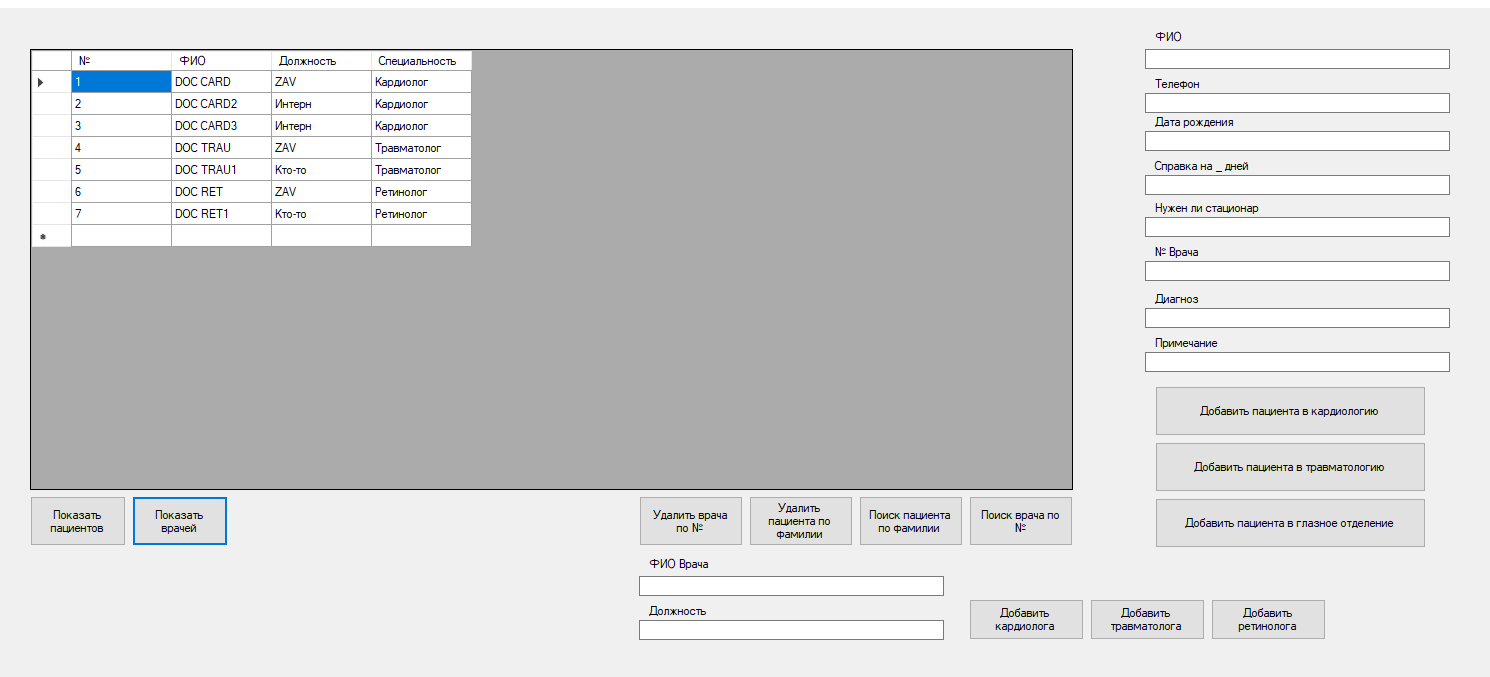


Рисунок 3.2 – Початкова база даних лікарів

Щоб додати лікаря до бази необхідно заповнити textbox «ПІБ лікаря» та «посада» й обрати його спеціальність за допомогою кнопок «Додати кардіолога», «Додати травматолога» або «Додати ретинолога». Результат цих дій можна побачити на рисунку 3.3, у цьому прикладі до бази даних був доданий ретинолог Іванов, кардіолог Петров, травматолог Сідоров у цьому ж порядку. № лікаря задається автоматично.

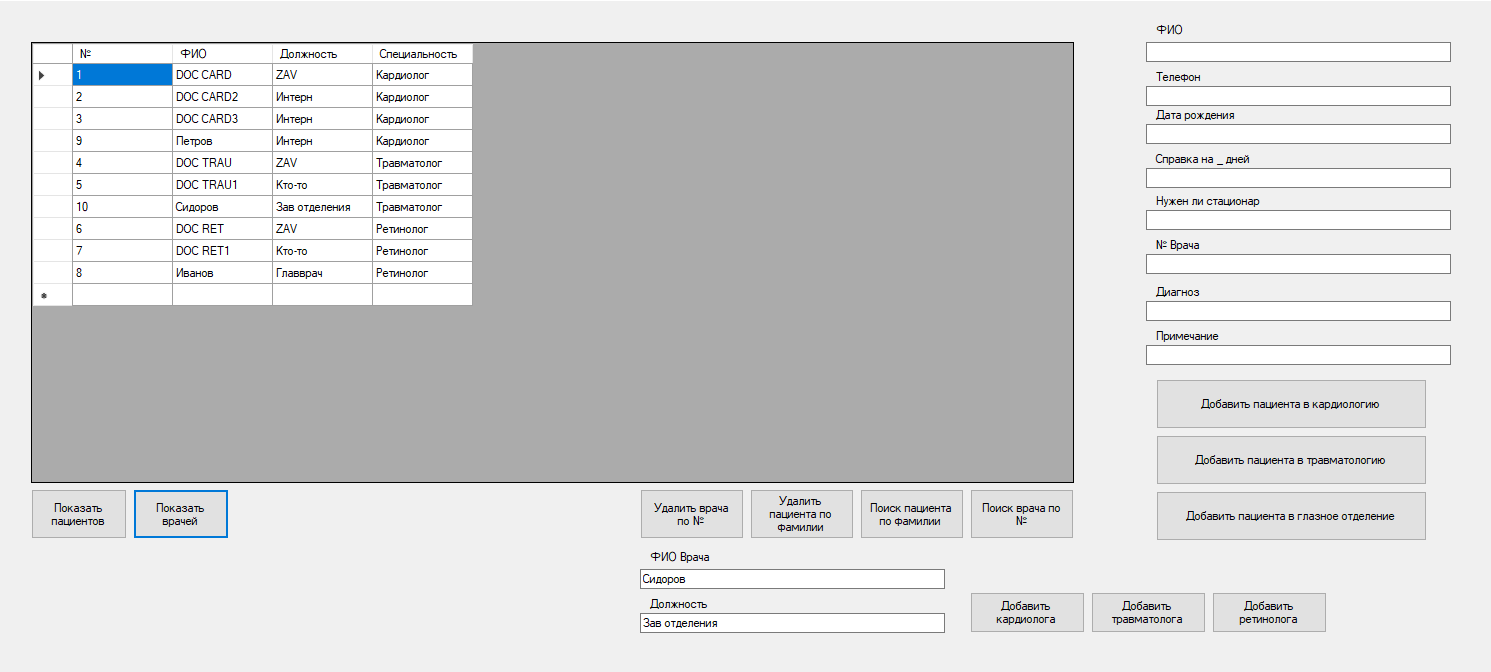


Рисунок 3.3 – Приклад додавання лікарів до бази даних

Щоб додати пацієнта необхідно заповнити textbox «ПІБ», «Телефон», «Дата народження», «Довідка на \_ днів», «Чи потрібен стаціонар», «№ Лікаря», «Діагноз», «Примітка» й обрати до якого відділення його додати за допомогою кнопок «Додати до кардіології», «Додати до травматології» або «Додати до очного відділення». Результат цих дій можна побачити на рисунку 3.4, у цьому прикладі до бази даних був доданий пацієнт кардіології Стрельцов з № лікаря 9, пацієнт травматології Жук з № лікаря 10, пацієнт очного відділення Сонін з № лікаря 8.

Щоб знайти пацієнта потрібно заповнити textbox «ПІБ» та натиснути кнопку «Пошук пацієнта по ПІБ». Результат цих дій можна побачити на рисунку 3.5, у цьому прикладі було знайдено пацієнта з ПІБ Жук.

Щоб знайти лікаря необхідно заповнити textbox «№ Лікаря» та натиснути кнопку «Пошук лікаря по №». Результат цих дій можна побачити на рисунку 3.6, у цьому прикладі було знайдено лікаря з номером 4.

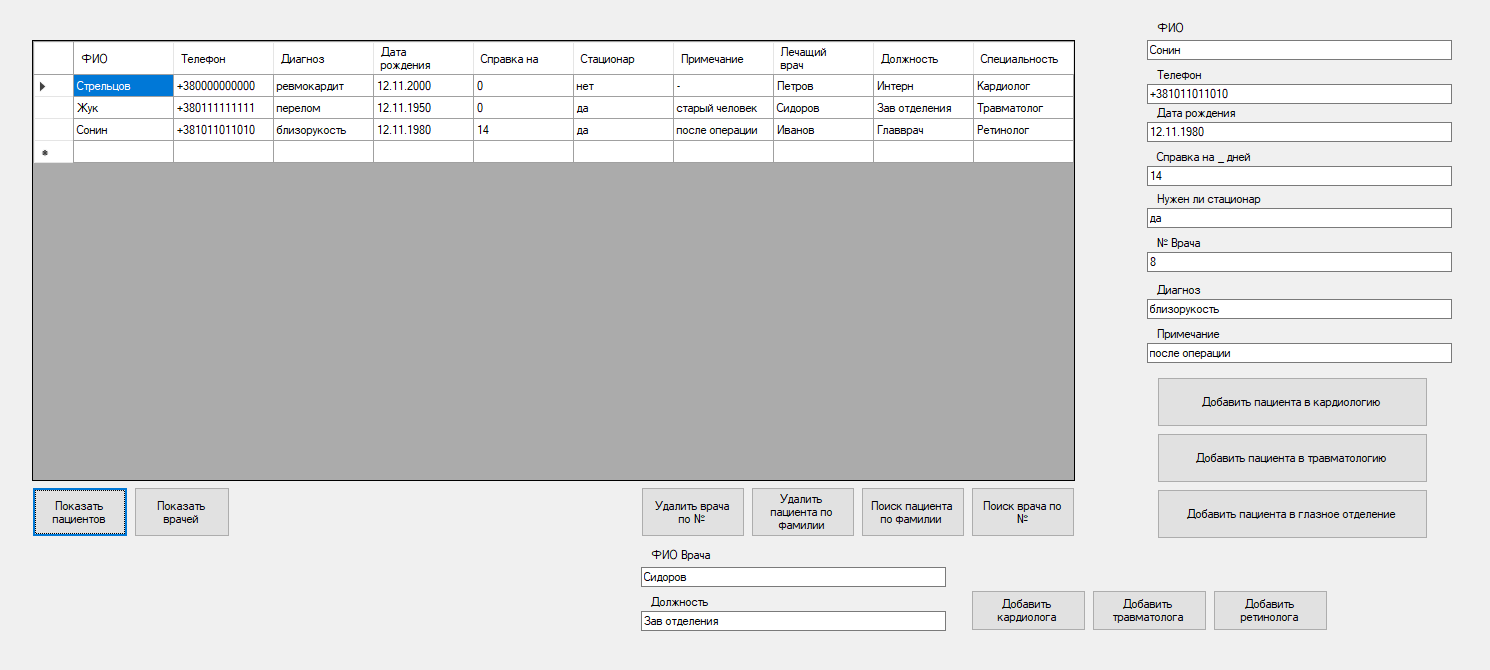


Рисунок 3.4 – Приклад додавання пацієнтів до бази даних

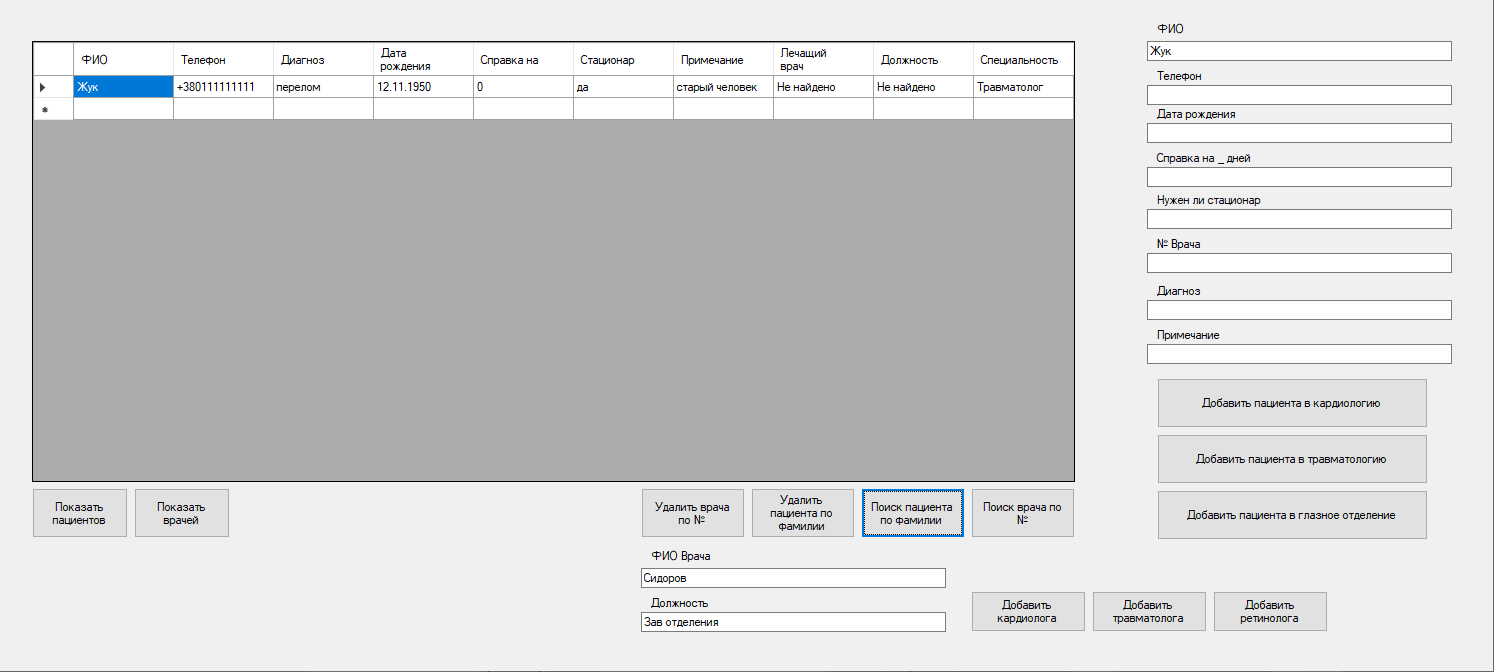


Рисунок 3.5 – Приклад пошуку пацієнта

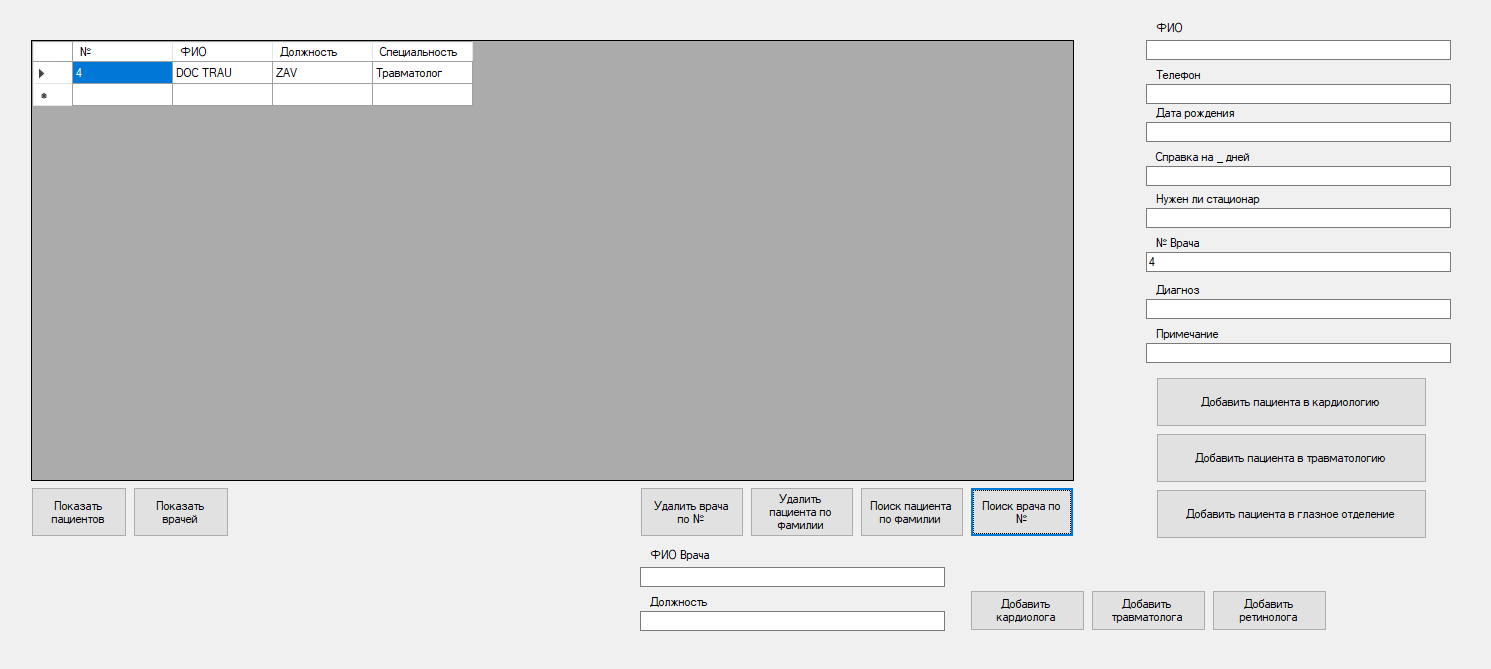


Рисунок 3.6 – Приклад пошуку лікаря

Щоб видалити лікаря необхідно заповнити textbox «№ Лікаря» та натиснути кнопку «Видалити лікаря по №». Результат цих дій можна побачити на рисунку 3.7, у цьому прикладі було видалено лікаря з номером 4.

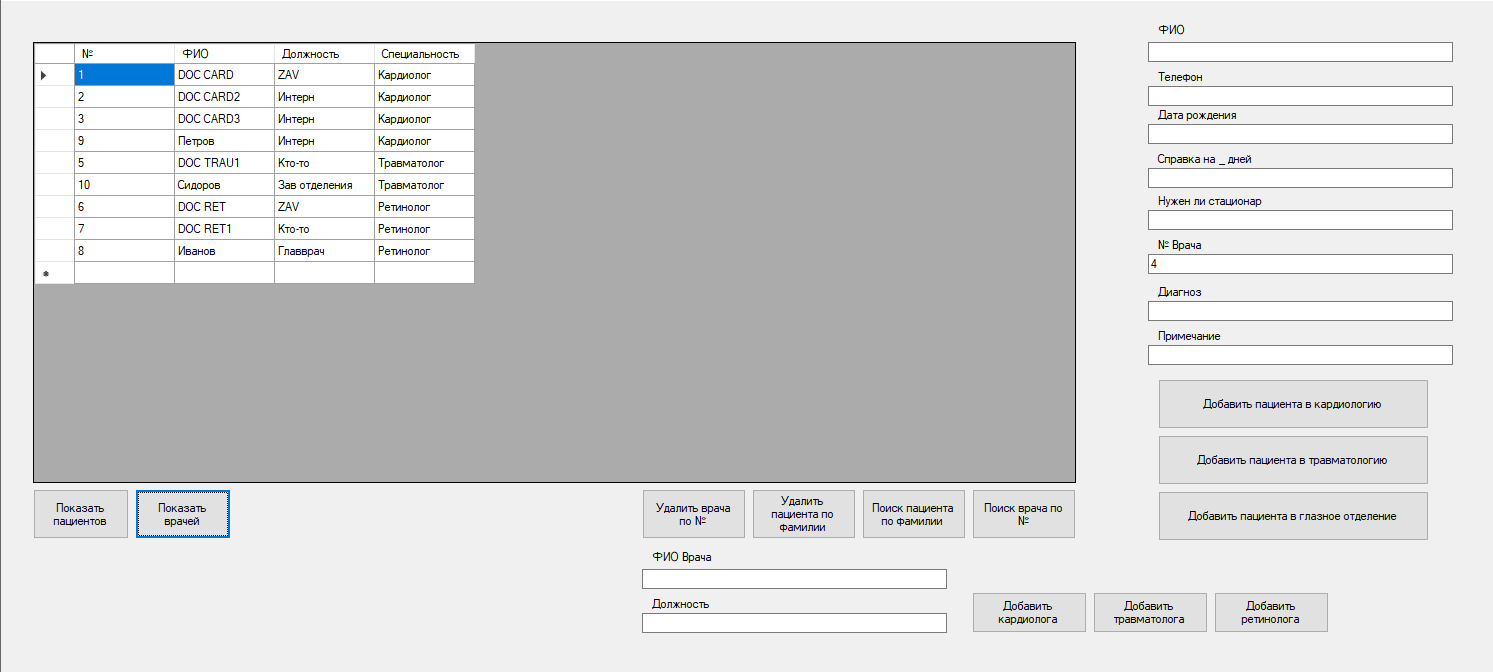


Рисунок 3.7 – Приклад видалення лікаря

Щоб видалити пацієнта потрібно заповнити textbox «ПІБ» та натиснути кнопку «Видалити пацієнта по ПІБ». Результат цих дій можна побачити на рисунку 3.8, у цьому прикладі було видалено пацієнта з ПІБ Жук.

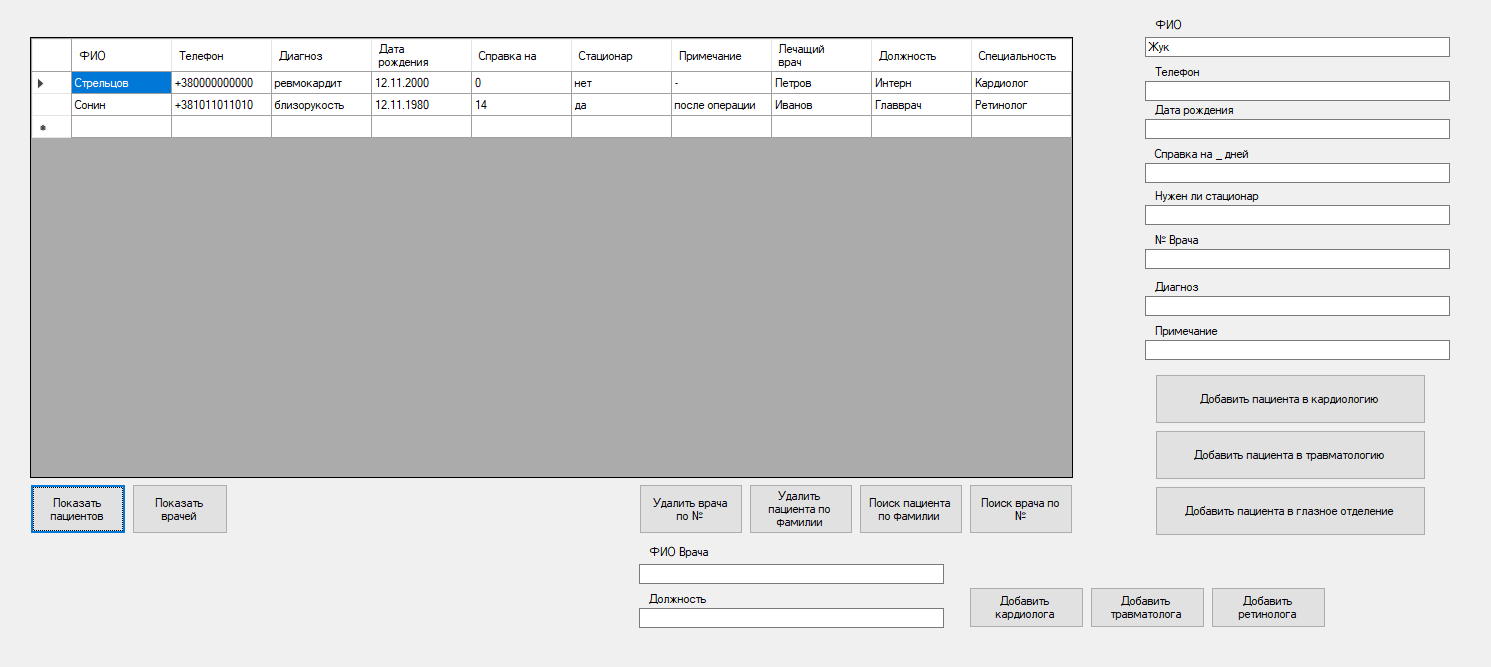


Рисунок 3.8 – Приклад видалення пацієнта

# Висновок

Під час виконання курсової роботи були отримані навички створення програмного забезпечення на мові С++ з використанням наслідування класів. У MS Visual Studio була створена діаграма класів із точки зору реалізації.

Розроблений програмний продукт дозволяє створити та редагувати базу даних.

Програмний продукт реалізує такі функції: введення даних в базу даних пацієнтів, друк бази даних пацієнтів, введення даних в базу даних лікарів, друк бази даних лікарів.

Розроблений застосунок може використовуватись у цілях зберігання інформації про лікарів та пацієнтів.

# Список використаних джерел

1) Інформаціїні технологіїї в суспільстві ХХІ століття [Електронний ресурс]. URL: [https://stud.com.ua/86670/informatika/informatsiyni\_tehnologiyi\_ suspilstvi\_stolittya](https://stud.com.ua/86670/informatika/informatsiyni_tehnologiyi_suspilstvi_stolittya)

2) Топ 7 IDE для програмування на С++ [Електронний ресурс]. URL: <https://itvdn.com/ru/blog/article/cplspls-top7(lfnf>

3) Інтегроване середовище розробки [Електронний ресурс]: URL: <https://ru.wikipedia.org/wiki/Інтегроване_середовище_розробки>

4) Діаграма класів [Електронний ресурс]: URL: <https://ru.wikipedia.org/wiki/UML#Діаграма_класів>

5) Ліберті Дж. Опануй самостійно С++ за 21 день. Видавничий дім «Вільямс», 2001. 832 с.

6) Страуструп Б. Мова програмування С++. Радіо та зв’язок ,1991. 352 с.

7) Лукас П. С ++ під рукою. - Київ: «ДиаСофт», 1993. 176 с.

8) Коллінз У. Структури даних і стандартна бібліотека шаблонів. «Біном-Пресс», 2004. 624 с

9) Додавання діаграми класів у проекти [Електронний ресурс]: <https://docs.microsoft.com/ru-ru/visualstudio/ide/class-designer/how-to-add-class-diagrams-to-projects?view=vs-2022>

10) Робота з конструктором класів [Електронний ресурс]: https://docs.microsoft.com/ru-ru/visualstudio/ide/class-designer/designing-and-viewing-classes-and-types?view=vs-2022

# ДОДАТОК А

MyForm.h:

class hospital {

std::string name;

public:

hospital() { name = "Госпиталь Синица"; };};

class cardiology : public hospital{

std::string spec;

public:

cardiology() { spec = "Кардиолог"; }

std::string get\_spec(){ return spec; };};

class cardiologist :public cardiology{

std::vector<std::string> fio\_doc, post, n;

public:

cardiologist() {}

int size\_vector() { return fio\_doc.size(); }

**и т.д. //ДОПИСЫВАЕТЕ свой код**