|  |
| --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  **VIỆN ĐIỆN TỬ - VIỄN THÔNG**  logo_128  **BÁO CÁO BÀI TẬP LỚN**  **KỸ THUẬT LẬP TRÌNH C/C++**  **Đề tài:**  ***Hệ thống quản lý bệnh viện***  Giảng viên hướng dẫn: PGS. TS. Nguyễn Đức Minh  Sinh viên thực hiện: Trần Thị Thảo Anh  MSSV: 20193287  Lớp: CTTT Hệ thống nhúng thông minh & IoT  Khóa: K64  Hà Nội, 3 - 2022 |

# LỜI NÓI ĐẦU

Hiện nay, công nghệ thông tin đang phát triển nhanh chóng và ngày càng được ứng dụng rộng rãi vào cuộc sống của con người. Cũng như trên thế giới, Việt Nam là một nước đang phát triển và công nghệ thông tin cũng đang phát triển nhanh chóng thì việc đầu tư cho công cuộc phát triển, nghiên cứu và ứng dụng công nghệ thông tin là rất quan trọng.

Một trong số ứng dụng của công nghệ thông tin là vào lĩnh vực quản lý. Hiện nay, công nghệ thông tin (CNTT) đang hiện diện và đóng vai trò quan trọng không thể thiếu trong quá trình quản trị, điều hành các hoạt động của các bệnh viện. Sự phát triển và ứng dụng của Internet đã làm thay đổi mô hình và cách thức hoạt động của bệnh viện.Vì vậy mà em chọn đề tài “Xây dựng chương trình quản lý bệnh viện bằng ngôn ngữ C/C++” làm đề tài nghiên cứu.

Em xin chân thành cảm ơn thầy PGS.TS. Nguyễn Đức Minh và anh Nguyễn Huy Đông đã giúp đỡ em trong suốt quá trình học môn Kỹ thuật lập trình C/C++. Và từ đó qua môn học này, em đã hiểu hơn về các kiến thức cơ bản về Hướng đối tượng(OOP) và nhiều kiến thức bổ ích khác.

# MỤC LỤC

[LỜI NÓI ĐẦU iii](#_Toc97217740)

[MỤC LỤC i](#_Toc97217741)

[DANH MỤC HÌNH ẢNH iii](#_Toc97217742)

[CHƯƠNG 1. GIỚI THIỆU CHUNG 4](#_Toc97217743)

[1.1 Lập trình hướng đối tượng C/C++ 4](#_Toc97217744)

[1.2 QT Creator 4](#_Toc97217745)

[CHƯƠNG 2. PHÂN TÍCH XÂY DỰNG CHƯƠNG TRÌNH 6](#_Toc97217746)

[2.1 Giới thiệu thuật toán 6](#_Toc97217747)

[2.2 Các module chính trong chương trình 6](#_Toc97217748)

[2.2.1 Sơ đồ tổng quát 6](#_Toc97217749)

[2.2.2 Xây dựng các lớp cơ sở 7](#_Toc97217750)

[2.2.3 Xây dựng các cửa sổ với chức năng chính 7](#_Toc97217751)

[2.2.4 Sơ đồ khối các chức năng 8](#_Toc97217752)

[CHƯƠNG 3. TRIỂN KHAI THỰC HIỆN 12](#_Toc97217753)

[3.1 Cấu trúc chung 12](#_Toc97217754)

[3.1.1 Class People và các lớp kế thừa 12](#_Toc97217755)

[3.1.2 Class Hospital 13](#_Toc97217756)

[3.1.3 Class DoctorList và Class PatientList 14](#_Toc97217757)

[3.2 Hàm main 15](#_Toc97217758)

[CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM 16](#_Toc97217759)

[4.1 Giao diện chung 16](#_Toc97217760)

[4.2 Giao diện quản lý danh sách bác sĩ 16](#_Toc97217761)

[4.3 Giao diện quản lý danh sách bệnh nhân 17](#_Toc97217762)

[4.4 Thêm 1 bác sĩ hoặc bệnh nhân mới 17](#_Toc97217763)

[4.5 In ra danh sách đầy đủ 18](#_Toc97217764)

[4.6 Sắp xếp danh sách theo tên 18](#_Toc97217765)

[4.7 Đọc dữ liệu từ file có sẵn 19](#_Toc97217766)

[4.8 Xuất dữ liệu ra file 19](#_Toc97217767)

[KẾT LUẬN 20](#_Toc97217768)

[TÀI LIỆU THAM KHẢO 20](#_Toc97217769)

# DANH MỤC HÌNH ẢNH

[Hình 2.1 Sơ đồ tổng quát của chương trình 8](#_Toc97217529)

[Hình 2.2 Sơ đồ các lớp cơ sở 8](#_Toc97217530)

[Hình 2.3 Ý tưởng các cửa sổ với chức năng chính 9](#_Toc97217531)

[Hình 2.4 Đọc file dữ liệu 9](#_Toc97217532)

[Hình 2.5 Ghi ra file dữ liệu 9](#_Toc97217533)

[Hình 2.6 Thêm đối tượng vào danh sách 10](#_Toc97217534)

[Hình 2.7 Xóa đối tượng khỏi danh sách 10](#_Toc97217535)

[Hình 2.8 Sắp xếp theo tên 10](#_Toc97217536)

[Hình 2.9 Tìm đối tượng theo tên hoặc mã số 11](#_Toc97217537)

[Hình 2.10 Sắp xếp theo viện phí 12](#_Toc97217538)

[Hình 3.1 Các thư mục chính trong chương trình 15](#_Toc97217539)

[Hình 3.2 File “DoctorList.cpp” 15](#_Toc97217540)

[Hình 3.3 File “DoctorList.cpp” 16](#_Toc97217541)

[Hình 4.1 Giao diện chung 17](#_Toc97217542)

[Hình 4.2 Giao diện quản lý bác sĩ 17](#_Toc97217543)

[Hình 4.3 Giao diện quản lý bệnh nhân 18](#_Toc97217544)

[Hình 4.4 Thêm bác sĩ mới 18](#_Toc97217545)

[Hình 4.5 Thêm bệnh nhân mới 19](#_Toc97217546)

[Hình 4.6 In ra danh sách bác sĩ đầy đủ thông tin 19](#_Toc97217547)

[Hình 4.7 In ra danh sách bác sĩ đầy đủ thông tin đã được sắp xếp theo tên 19](#_Toc97217548)

[Hình 4.8 In ra danh sách bác sĩ đọc từ file dữ liệu Doctor.txt 20](#_Toc97217549)

[Hình 4.9 Dữ liệu đã xuất ra file DoctorWrite.txt 20](#_Toc97217550)

# GIỚI THIỆU CHUNG

## Lập trình hướng đối tượng C/C++

Lập trình hướng đối tượng (OOP) là một mô hình lập trình máy tính tổ chức thiết kế phần mềm xung quanh dữ liệu hoặc đối tượng. Một đối tượng có thể được định nghĩa là một trường dữ liệu có các thuộc tính và hành vi duy nhất.

Bước đầu tiên trong OOP là thu thập tất cả các đối tượng mà một lập trình viên muốn thao tác và xác định cách chúng liên quan tới nhau – hay còn gọi là mô hình hóa dữ liệu.

Khi một đối tượng được xác định, nó được gắn nhãn với một lớp đối tượng xác định loại dữ liệu mà nó chứa và bất kỳ chuỗi logic nào có thể thao tác với nó. Mỗi chuỗi logic riêng biệt được gọi là một phương thức. Các đối tượng có thể giao tiếp với các giao diện được gọi là message.

Cấu trúc hoặc khối xây dựng của OOP bao gồm:

* ***Lớp*** là kiểu dữ liệu do người dùng xác định, đóng vai trò là bản thiết kế cho các đối tượng, thuộc tính và phương thức riêng lẻ.
* ***Đối tượng*** là các thể hiện của một lớp được tạo bằng các dữ liệu được xác định cụ thể. Các đối tượng có thể tương ứng với các đối tượng trong thế giới thực hoặc một thực thể trừu tượng.
* ***Phương thức*** là các chức năng được xác định bên trong một lớp mô tả các hành vi của một đối tượng. Mỗi phương thức có trong định nghĩa lớp bắt đầu bằng một tham chiếu đến một đối tượng thể hiện. Ngoài ra, các chương trình con chứa trong một đối tượng được gọi là các phương thức thể hiện. Sử dụng các phương thức để tái sử dụng hoặc giữ chức năng được gói gọn trong một đối tượng tại một thời điểm.
* ***Thuộc tính*** được định nghĩa trong mẫu lớp và đại diện cho trạng thái của đối tượng. Các đối tượng sẽ có dữ liệu được lưu trữ trong trường thuộc tính. Các thuộc tính của lớp thuộc về chính lớp đó.

## QT Creator

QT Creator là một IDE cung cấp các công cụ để thiết kế và phát triển ứng dụng với khung ứng dụng QT. QT Creator cung cấp các công cụ để hoàn thành nhiệm vụ trong toàn bộ vòng đời phát triển ứng dụng, từ tạo dự án đến triển khai ứng dụng cho các nền tảng mục tiêu.

* ***Quản lý dự án***: Để có thể xây dựng  và chạy các ứng dụng, Qt Creator cần thông tin giống như một trình biên dịch cần. Thông tin này được chỉ định trong cài đặt dự án.
* ***Thiết kế giao diện người dùng***: Để tạo giao diện người dùng linh hoạt, hiện đại, trực quan, có thể sử dụng Qt Quick và Qt Design Studio. Nếu cần một giao diện người dùng truyền thống được cấu trúc rõ ràng và thực thi giao diện nền tảng, có thể sử dụng Qt Designer tích hợp.
* ***Lập trình***: Qt Creator biết cách xây dựng và chạy các ứng dụng, hiểu các ngôn ngữ C ++ và QML dưới dạng mã. Điều này cho phép nó cung cấp các tính năng hữu ích, chẳng hạn như kiểm tra cú pháp mã, hoàn thành mã và các hành động tái cấu trúc.
* ***Xây dựng và chạy chương trình***: Qt Creator được tích hợp với các hệ thống đa nền tảng để tự động hóa xây dựng: qmake, Qbs, CMake và Autotools. Ngoài ra, có thể nhập dự án dưới dạng dự án chung và kiểm soát hoàn toàn các bước và lệnh được sử dụng để xây dựng dự án. Qt Creator cung cấp công cụ để chạy và triển khai các ứng dụng Qt được xây dựng cho môi trường máy tính để bàn hoặc thiết bị.
* ***Kiểm thử***: Qt Creator được tích hợp với một số trình gỡ lỗi gốc bên ngoài: Trình gỡ lỗi biểu tượng GNU (GDB), Trình gỡ lỗi bảng điều khiển Microsoft (CDB) và trình gỡ lỗi JavaScript nội bộ. Trong chế độ Gỡ lỗi có thể kiểm tra trạng thái ứng dụng của mình. Qt Creator được tích hợp vào khung kiểm tra Qt Test, Boost.Test, Catch 2 và Google C ++ Testing cho các ứng dụng và thư viện kiểm tra đơn vị. Sử dụng Qt Creator để tạo, xây dựng và thử nghiệm tự động.
* ***Xuất bản***: Qt Creator cho phép tạo các gói cài đặt cho thiết bị di động phù hợp để xuất bản lên kho ứng dụng và các kênh khác. Phải đảm bảo rằng nội dung gói đáp ứng các yêu cầu để xuất bản trên kênh.

# PHÂN TÍCH XÂY DỰNG CHƯƠNG TRÌNH

## Giới thiệu thuật toán

Chương trình quản lý bệnh viện quản lý 2 đối tượng chính là bệnh nhân và bác sĩ. Hai đối tượng này là 1 class kế thừa và mang các đặc tính chung của class People (Người):

* Mã số
* Tên
* Ngày sinh
* Giới tính
* Khoa

Bên cạnh đó, mỗi đối tượng sẽ mang đăc tính riêng.

Người dùng quản lý sản phẩm thông qua vector với dữ liệu nhập từ file data sẵn có và sau khi kết thúc công việc sẽ ghi lại thông tin các sản phẩm vào file đó. Các phương thức được sử dụng để quản lý sản phẩm gồm:

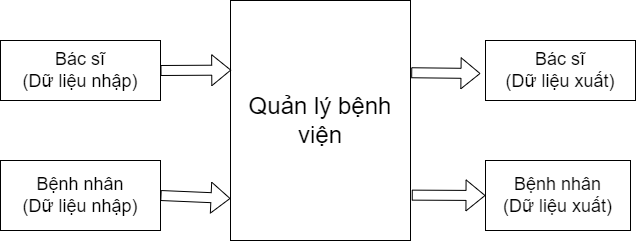
* Nhập người mới
* Xóa 1 người
* Xem toàn bộ danh sách
* Tìm kiếm 1 người
* Sắp xếp ( theo tên / theo viện phí – với bệnh nhân)

## Các module chính trong chương trình

### Sơ đồ tổng quát

* Đầu vào: File dữ liệu dạng .txt
* Đầu ra: File dữ liệu đã được xử lý dạng .txt

Hình 1 thể hiện sơ đồ tổng quát của chương trình

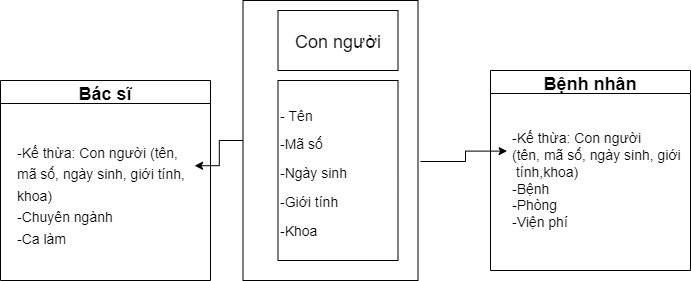


Hình 2.1 Sơ đồ tổng quát của chương trình

### Xây dựng các lớp cơ sở

Lớp cơ sở là Con người (class People) mang các thuộc tính đặc trưng là mã số, tên, ngày sinh, giới tính, khoa.

Các lớp thừa kế thuộc tính của con người là Bác sĩ (class Doctor) và Bệnh nhân (class Patient). Bên cạnh đó, các thuộc tính thêm đặc trưng riêng cũng được triển khai như hình 2.

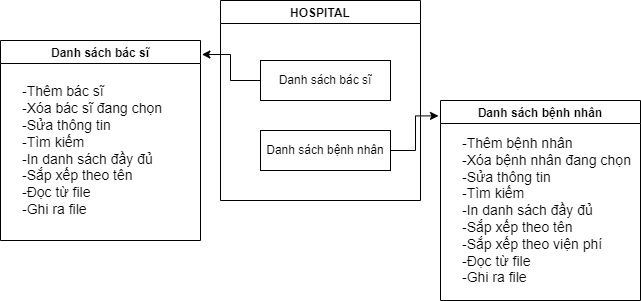


Hình 2.2 Sơ đồ các lớp cơ sở

### Xây dựng các cửa sổ với chức năng chính

Cửa sổ chính Hospital được mở ra với chức năng chọn danh sách truy cập bệnh nhân hay bác sĩ.

Trong cửa sổ danh sách bác sĩ hay danh sách bệnh nhân đều có các ô click tương ứng với từng chức năng như hình 3.





Hình 2.3 Ý tưởng các cửa sổ với chức năng chính

### Sơ đồ khối các chức năng

Mỗi chức năng trong cửa sổ danh sách bệnh nhân/ bác sĩ được thể hiện qua các sơ đồ khối dưới đây

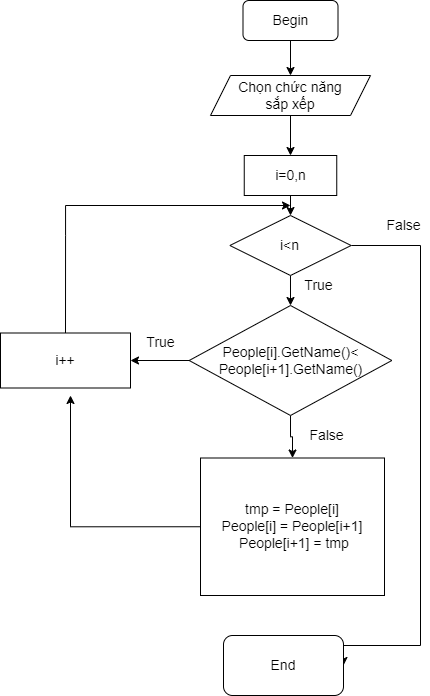
*Chức năng đọc và ghi ra file dữ liệu*

|  |  |
| --- | --- |
| Hình 2.4 Đọc file dữ liệu | Hình 2.5 Ghi ra file dữ liệu |

*Chức năng thêm và xóa đối tượng vào danh sách*

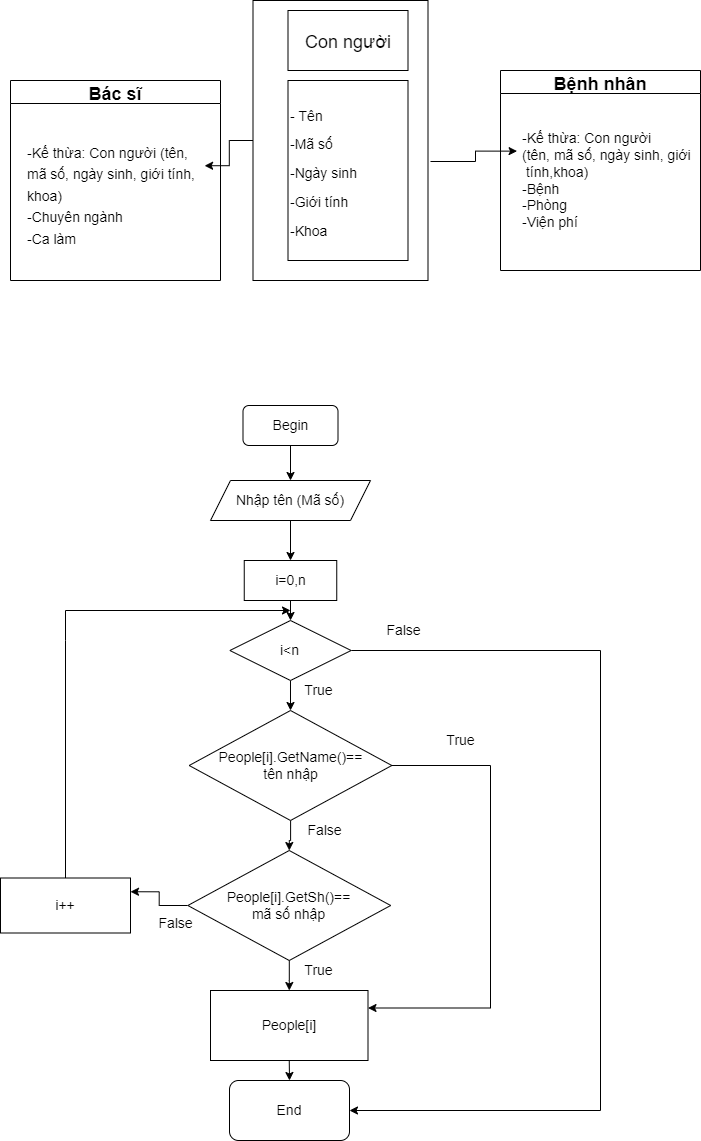
|  |  |
| --- | --- |
| Hình 2.6 Thêm đối tượng vào danh sách | Hình 2.7 Xóa đối tượng khỏi danh sách |

*Chức năng sắp xếp theo tên*



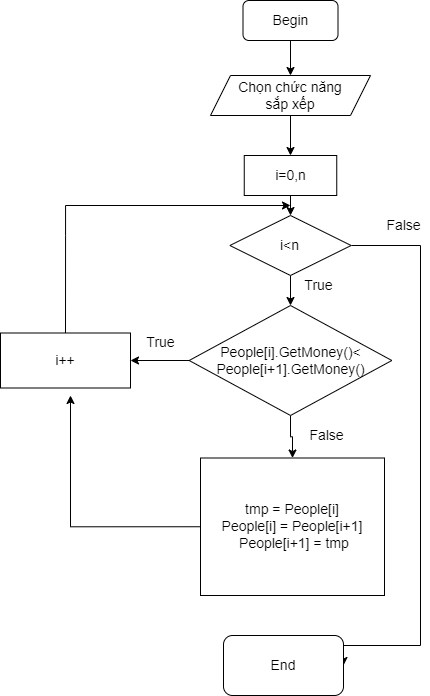
Hình 2.8 Sắp xếp theo tên

*Chức năng tìm đối tượng theo tên hoặc mã số*



Hình 2.9 Tìm đối tượng theo tên hoặc mã số

*Chức năng sắp xếp theo viện phí (chỉ trong giao diện danh sách bệnh nhân)*



Hình 2.10 Sắp xếp theo viện phí

# TRIỂN KHAI THỰC HIỆN

## Cấu trúc chung

Phần này trình bày cấu trúc chung của các tệp chương trình dùng để triển khai thuật toán.

### Class People và các lớp kế thừa

class **People**: public QListWidgetItem{

public:

**People**(){ }

**People**(QString name);

**People**(QString sh,QString name ,QString date,QString gender,QString department);

~***People***(){}

void **SetName**(QString name);

void **SetDate**(QString date);

void **SetDepartment**(QString department);

void **SetSh**(QString SH);

void **SetGender**(QString gender);

QString **GetName**() ;

QString **GetDate**() ;

QString **GetDepartment**() ;

QString **GetSh**();

QString **GetGender**();

protected:

QString Name;

QString Sh;

QString Date;

QString Gender;

QString Department;

};

Code: “people.h”

Class People gồm các thuộc tính được ghi trong file “people.h” và phương thức khởi tạo, cài đặt, lấy ra các thuộc tính để quản lý và hiển thị được ghi trong file “people.cpp”.

Các class con kế thừa của People là class Doctor (bác sĩ) , class Patient (bệnh nhân) có thêm thuộc tính riêng:

* Class Doctor: Department (Khoa), Shift (Ca làm)
* Class Patient: Disease (Bệnh), Room (Phòng), Money (Viện phí)

### Class Hospital

Class Hospital tạo giao diện ban đầu cho chương trình.

Hospital::**Hospital**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::Hospital)

{

ui->setupUi(this);

}

Hospital::~***Hospital***()

{

delete ui;

delete m\_DoctorList;

delete m\_PatientList;

}

void Hospital::**on\_pushButton\_released**()

{

m\_DoctorList = new DoctorList;

m\_DoctorList->Show();

}

void Hospital::**on\_pushButton\_2\_clicked**()

{

m\_PatientList = new PatientList;

m\_PatientList->Show();

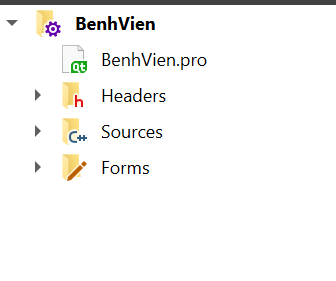
}

Code: “hospital.cpp”

File “hospital.cpp” chứa các thuộc tính m\_DoctorList và m\_PatientList là các file tạo ra các giao diện riêng của bác sĩ, bệnh nhân với các chức năng chứa trong đó.

### Class DoctorList và Class PatientList

Mỗi class đều chứa các thuộc tính và phương thức liên quan đến các chức năng được trình bày ở mục 2.2. Trong đó, các file header (.h) dùng để khai báo, phương thức khởi tạo, cài đặt, lấy ra các thuộc tính để quản lý và hiển thị được ghi trong file source (.cpp) và giao diện được trình bày trong các file form (.ui)



Hình 3.1 Các thư mục chính trong chương trình

*Class DoctorList*

Class DoctorList chứa các hàm nhập, xóa thông tin của bác sĩ và các hàm quản lý nút bấm dẫn đến các chức năng khác đã được nêu ở mục 2.2 đối với đối tượng là bác sĩ.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 3.2 File “DoctorList.cpp”

*Class PatientList*

Class DoctorList chứa các hàm nhập, xóa thông tin của bệnh nhân và các hàm quản lý nút bấm dẫn đến các chức năng khác đã được nêu ở mục 2.2 đối với đối tượng là bệnh nhân

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 3.3 File “DoctorList.cpp”

## Hàm main

Hàm “main.cpp” chứa đoạn lệnh chạy chương trình dưới dạng 1 application có giao diện hoàn thiện

#include "hospital.h"

#include "doctorlist.h"

#include <QApplication>

int main(int argc, char \*argv[]){

QApplication a(*argc*, *argv*);

Hospital w;

w.show();

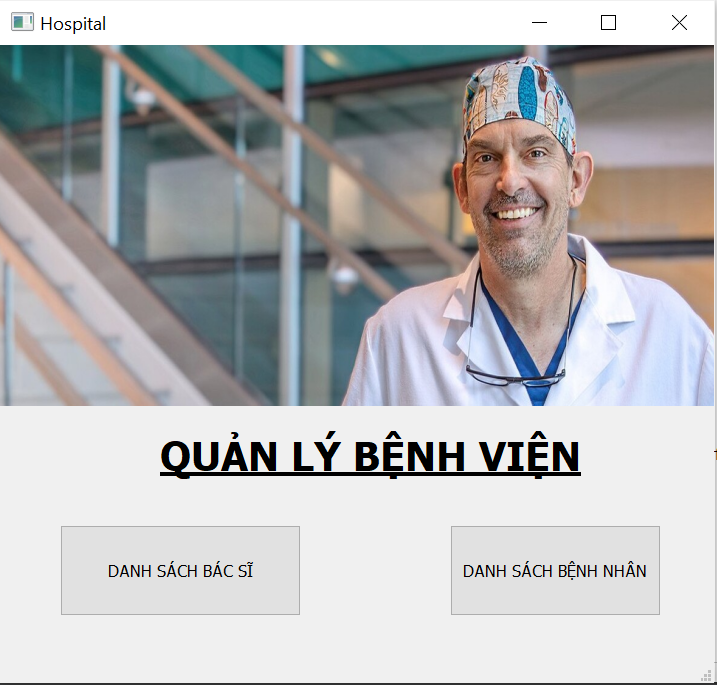
return a.exec();

}

Code: “main.cpp”

# KẾT QUẢ THỰC NGHIỆM

## Giao diện chung



Hình 4.1 Giao diện chung

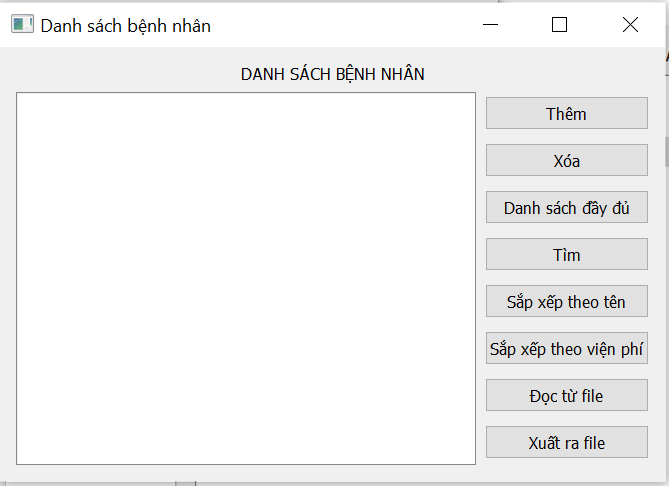
## Giao diện quản lý danh sách bác sĩ

Ảnh có chứa văn bản

Mô tả được tạo tự động

Hình 4.2 Giao diện quản lý bác sĩ

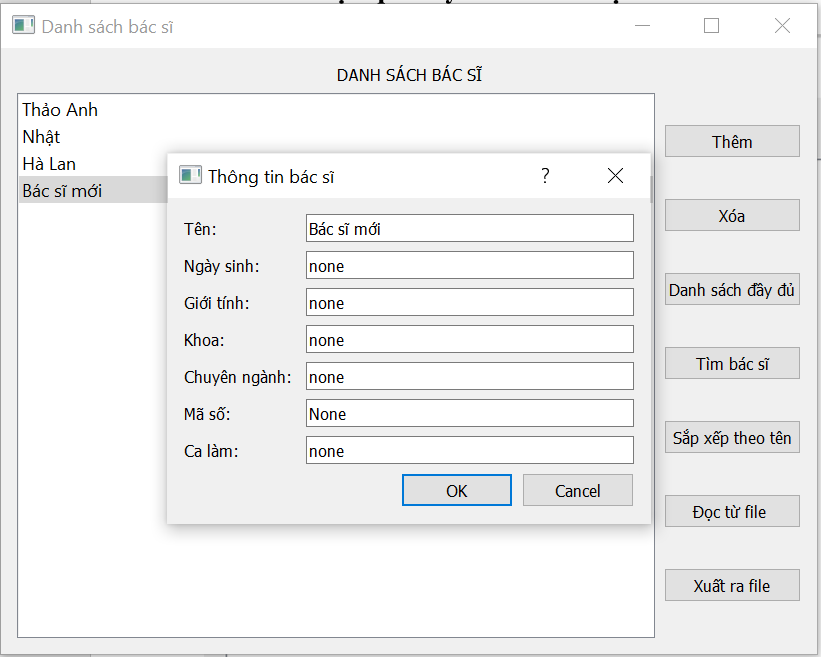
## Giao diện quản lý danh sách bệnh nhân



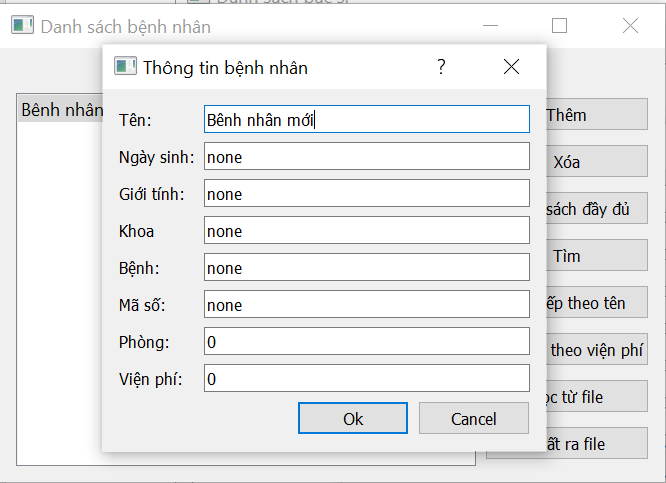
Hình 4.3 Giao diện quản lý bệnh nhân

## Thêm 1 bác sĩ hoặc bệnh nhân mới

Các chức năng của chương trình quản lý bác sĩ và quản lý bệnh nhân tương đồng nhau:

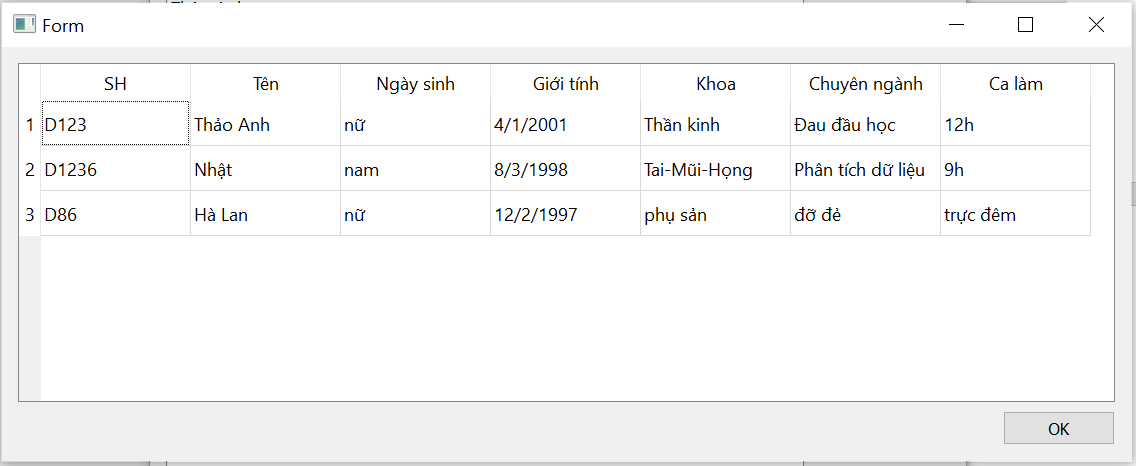


Hình 4.4 Thêm bác sĩ mới



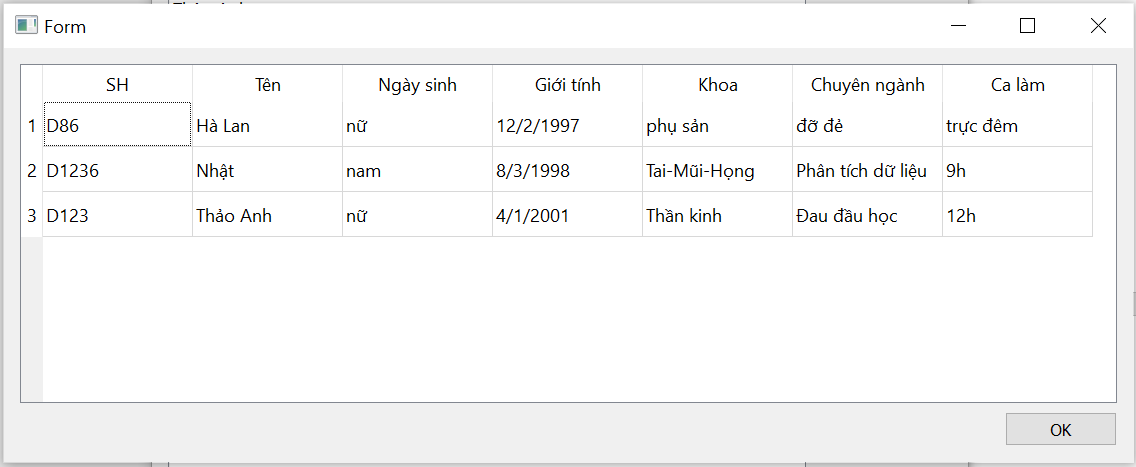
Hình 4.5 Thêm bệnh nhân mới

## In ra danh sách đầy đủ



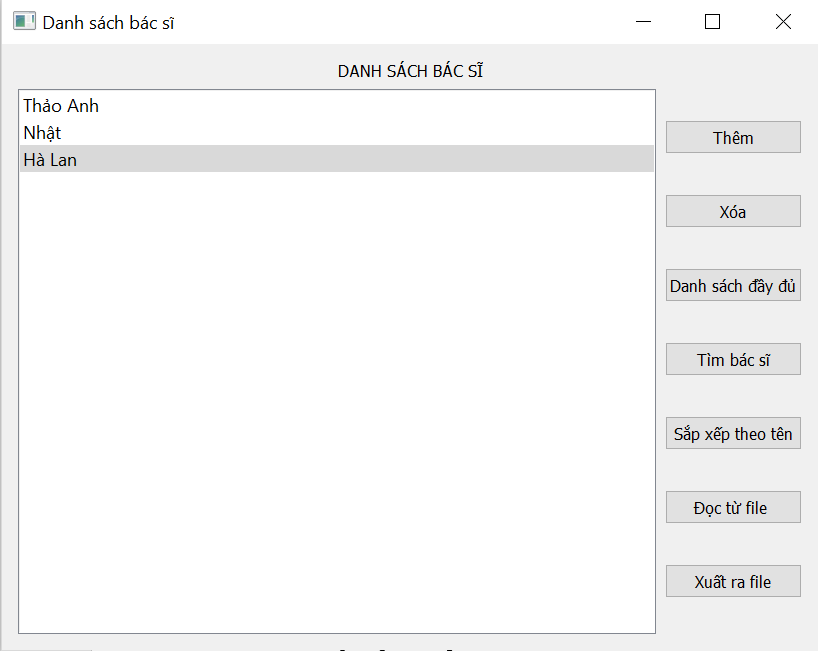
Hình 4.6 In ra danh sách bác sĩ đầy đủ thông tin

## Sắp xếp danh sách theo tên



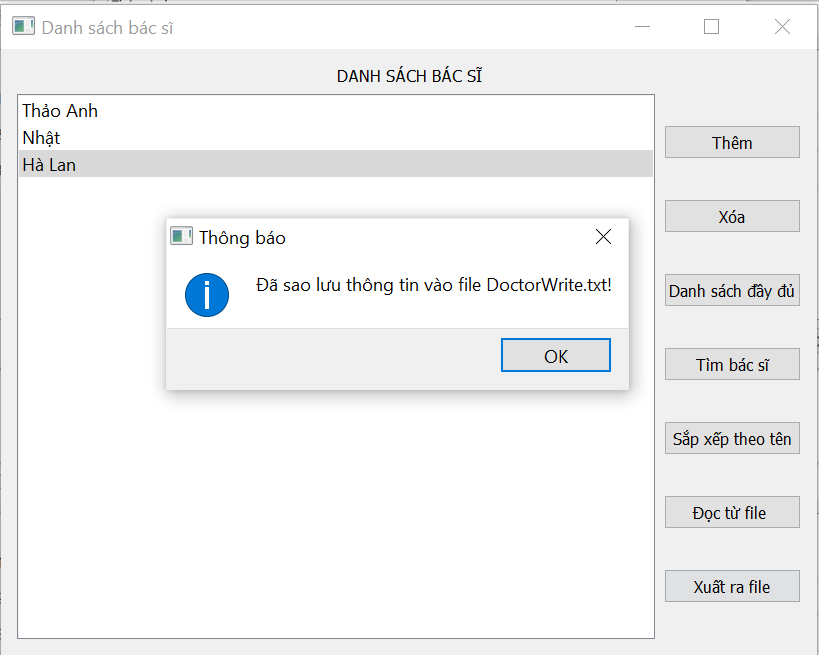
Hình 4.7 In ra danh sách bác sĩ đầy đủ thông tin đã được sắp xếp theo tên

## Đọc dữ liệu từ file có sẵn



Hình 4.8 In ra danh sách bác sĩ đọc từ file dữ liệu Doctor.txt

## Xuất dữ liệu ra file



Hình 4.9 Dữ liệu đã xuất ra file DoctorWrite.txt

KẾT LUẬN

Chương trình đã hoàn thiện các yêu cầu cơ bản của hệ thống quản lí bệnh viện với mô hình nhỏ, thu gọn. Giao diện của chương trình dễ thao sử dụng, thao tác. Tuy nhiên giao diện chưa được đẹp và các chức năng chưa được tối ưu hóa với khối lượng dữ liệu lớn.

Qua bài tập lớn này, em có thêm cho mình những tư duy về lập trình C/C++ cũng như lập trình hướng đối tượng. Em xin chân thành cảm ơn thầy Nguyễn Đức Minh và anh Nguyễn Huy Đông đã cùng đồng hành, hướng dẫn em hoàn thành bài tập này.

TÀI LIỆU THAM KHẢO

1. [https://nguyenvanhieu.vn/lap-trinh-huong-doi-tuong-cpp](https://nguyenvanhieu.vn/lap-trinh-huong-doi-tuong-cpp/)
2. [https://devnt.org/qt-app-example-quan-ly-sinh-vien](https://devnt.org/qt-app-example-quan-ly-sinh-vien/)
3. <https://www.youtube.com/channel/UCs6nmQViDpUw0nuIx9c_WvA>
4. [https://www.geeksforgeeks.org](https://www.geeksforgeeks.org/)
5. https://stackoverflow.com