C++ Project:

Hospital Information Management System

（Development version)

Team members:
Danying Xu（50%）
Ruibo Ma（50%）

Contents

Chapter 1 The introduction 3

1.1 The purpose of curriculum design 3

1．2 Background and significance of curriculum design 3

1．3 Curriculum design environment 3

Chapter 2 Demand analysis 4

2.1 Problem description 4

2.2 Functional requirements 4

2.3Problems develop 4

Chapter 3 The system design 5

3.1 Class design 5

3.2 The function design 5

Chapter 4 System implementation 6

Doctor.h 6

Doctor.cpp 7

Patient.h 9

Main.cpp 18

Result 36

Chapter 5 Discussion 39

Chapter 6 Conclusion 39

Reference 39

Chapter 1 The introduction

1.1 The purpose of curriculum design

（1）Students are required to master the basic knowledge and skills of C++ language.

（2）Master the basic ideas and methods of object-oriented programming.

(3) Be able to use the basic knowledge and skills to solve simple object-oriented programming problems.

1．2 Background and significance of curriculum design

1．2．1 Background of course design

(1) Theoretical Research basis

Good command of C++ curriculum design language。

(2) Technical support

Proficient application of class declaration and object definition, inheritance, derivation and file flow in C++ object-oriented programming

1．2．2 The meaning of curriculum design

After a semester of learning c + + program design, curriculum design can be a good test knowledge, for itself, is a good time to check leakage fill a vacancy, can will be to review and consolidate knowledge, will all previous knowledge together, good together, organic union, form a knowledge network system, to achieve mastery through a comprehensive study of knowledge, can let oneself further master program design language, and able to skillfully use.

1．3 Curriculum design environment

Microsoft Visual Studio 2019

Chapter 2 Demand analysis

2.1 Problem description

Construct a hospital information system to store the information of hospital, department and doctors. The system can add, edit, search and delete doctors. It also can clear all the data that has been stored.

2.2 Functional requirements

Design Doctor class. There are some information about doctors, including full name, birthday, hire date, job, sex and ID. ID can be input and used as the pointer to locate the information of the doctor in the specific txt file. There are a few more functions that are used to make sure the main functions can work. For example, before the main functions are generated every time, a function can help to store all ID in an advanced-set vector to make the process of main functions easier.

2.3 Problems develop

Add a patient on the basis of the doctor, and by using the case as a connecting tool, you can see some basic information of the patient, especially that of the treating doctor. This enables one-to-one correspondence between the patient and the doctor.

Chapter 3   The system design

3.1 Class design

(1) Defines the Doctor class, which contains the data member ID, firstname, lastname, sex, byear, bmonth, bday, syear, smonth, sday.

(2) Defines the MedicalRecord class, which contains the data member ID, firstname, lastname, sex, date, diagnose, advice, doctors' signatures.

(3) Defines the Date class, which contains the data member year, month, day.

(4) Defines the PatientDataBase class, which contains basic information about the patient.

3.2 The function design

In the Doctor class, DoctorAdd() increase doctor information; DoctorDelete() delete doctor information; DoctorChange() change doctor information; DoctorPrint() print doctor’s information; bool checkdocid() checks to see if the ID exists; cleandocdata() clean all data; setVectordoc() stores the existing ID of the file in doc after each open.

In the MedicalRecord class, PatientAdd() increase patient information; PatientDelete() delete patient information; PatientChange() change patient information; clear() clear patient information; queryID() and void queryDS() look up patients from their IDS and doctors' signatures, respectively; disp() outputs all the medical records

Chapter 4　System implementation

Doctor.h

```
#pragma once
#include<iostream>
#include<array>
#include<iomanip>
#define _CRT_SECURE_NO_WARNINGS
#pragma warning(disable:4996)


#ifndef DOCTOR_H
#define DOCTOR_H
class Doctor
{
private:
    int id, job, sex;
    char last[15];
    char first[10];
    int byear, bmonth, bday;
    int syear, smonth, sday;
public:
    Doctor(int i = 0, int s = 0, int j = 0, const char* b = " ",
const char* c = " ", int by = 1900, int bm = 1,
        int bd = 1, int sy = 1920, int sm = 1, int sd = 1);
    bool leapyear(int);
    void setbirthdate(int, int);
    void setbday(int);
    void setstartdate(int, int);
    void setsday(int);
    void setfirst(const char*);
    void setlast(const char*);
    void setid(int);
    void setsex(int);
    void setjob(int);
    char* getfirst();
    char* getlast();
    int getid();
    int getsex();
    int getjob();
    int getbyear();
    int getbmonth();
    int getbday();
    int getsyear();
    int getsmonth();
    int getsday();
};
#endif
```

Doctor.cpp

```
#include<iostream>
#include<array>
#include<string>
#include<ctime>
#include"Doctor.h"
using namespace std;
#define _CRT_SECURE_NO_WARNINGS
#pragma warning(disable:4996)



const int D[13] = { 0,31,28,31,30,31,30,31,31,30,31,30,31 };//days
Doctor::Doctor(int i, int s, int j, const char* b, const char* c, int
by, int bm,
    int bd, int sy, int sm, int sd)//constructor
{
    setbirthdate(bm, by);
    setbday(bd);
    setstartdate(sm, sy);
    setsday(sd);
    setfirst(b);
    setlast(c);
    setid(i);
    setsex(s);
    setjob(j);
}
bool Doctor::leapyear(int t)//判断是闰年
{
    if (t % 400 == 0 || (t % 100 != 0 && t % 4 == 0))return true;
    else return false;
}
```

```
void Doctor::setbirthdate(int a, int c)
{
    if (a >= 1 && a <= 12)bmonth = a;
    else cout << "months must be 1-12\n";
    if (c >= 1900 && c <= 2100)byear = c;
    else cout << "years must be >=1900 and <=2100\n";

}
void  Doctor::setbday(int b)
{
    if ((bmonth == 2 && leapyear(byear) && b >= 1 && b <= 29) ||
(b >= 1 && b <= D[bmonth]))bday = b;
    else cout << "days is out of range for current month and year\n";

}
void  Doctor::setstartdate(int a, int c)
{
    if (a >= 1 && a <= 12)smonth = a;
    else cout << "months must be 1-12\n";
    if (c >= 1900 && c <= 2100)syear = c;
    else cout << "years must be >=1900 and <=2100\n";

}
void  Doctor::setsday(int b)
{
    if ((smonth == 2 && leapyear(syear) && b >= 1 && b <= 29) ||
(b >= 1 && b <= D[smonth]))sday = b;
    else cout << "days is out of range for current month and year\n";

}
void  Doctor::setfirst(const char* a) { strcpy(first, a); }//把a的地
址赋值给first的首地址
void Doctor::setlast(const char* a) { strcpy(last, a); }
void Doctor::setid(int a) { id = a; }
void Doctor::setsex(int s)
{
    sex = s;
}
void Doctor::setjob(int a) { job = a; }

char* Doctor::getfirst() { return first; }
char* Doctor::getlast() { return last; }
int Doctor::getid() { return id; }
int Doctor::getsex() { return sex; }
int Doctor::getbyear() { return byear; }
int Doctor::getbmonth() { return bmonth; }
int Doctor::getbday() { return bday; }
int Doctor::getsyear() { return syear; }
int Doctor::getsmonth() { return smonth; }
int Doctor::getsday() { return sday; }
int Doctor::getjob() { return job; }
```

Patient.h

```
#include <fstream>
#include <iostream>
#include <iomanip>
#include <string>
#include <cstdlib>
#ifndef PATIENT_H
#define PATIENT_H
#define _CRT_SECURE_NO_WARNINGS
#pragma warning(disable:4996)
using namespace std;
class Date                    //日期类
{
    int year;
    int month;
    int day;
public:
    Date(int a = 2015, int b = 5, int c = 20)
    {
        year = a;
        month = b;
        day = c;
    }
    int getyear() { return year; }
    int getmonth() { return month; }
    int getday() { return day; }
    int setyear(int y) { year = y; }
    int setmonth(int m) { month = m; }
    int setday(int d) { day = d; }
    friend class PatientDataBase;
};
class MedicalRecord
{
private:
    int ID;
    char last[15];
    char first[10];
    char sex[5];
    int age;
    Date date;
    char diagnose[20];//诊断
    char advice[20]; //意见
    char DS[20];     //医生签名


public:
    MedicalRecord(int IDp = 0, const char* lastp = "", const char*
firstp = "", const char* sexp = "", int agep = 0, int a = 2015, int b
= 5, int c = 21, const char* MHp = "",
        const char* diagnosep = "", const char* advicep = "", const
char* DSp = "")
    {
        date = Date(a, b, c);
        ID = IDp;
        strcpy(first, firstp);
        strcpy(last, lastp);
        strcpy(sex, sexp);
        age = agep;
        strcpy(diagnose, diagnosep);
        strcpy(advice, advicep);
        strcpy(DS, DSp);



    }
    int getID() { return ID; }
    char* getlast() { return last; }
    char* getfirst() { return first; }
    char* getsex() { return sex; }
    int getage() { return age; }
    Date* getdate() { return &date; }
    char* getdiagnose() { return diagnose; }
    char* getadvice() { return advice; }
    char* getDS() { return DS; }
    void setID(int IDp) { ID = IDp; }
    void setfirst(const char* a)
    {
        strcpy(first, a);
    }
    void setlast(const char* a)
    {
        strcpy(last, a);
    }
    void setsex(const char* sexp)
    {
        strcpy(sex, sexp);
    }
    void setage(int agep)
    {
        age = agep;
    }
    void setdate(Date datep)
    {
        date = datep;
    }
    void setdiagnose(char* diagnosep)
    {
        strcpy(diagnose, diagnosep);
    }
    void setadvice(char* advicep)
    {
        strcpy(advice, advicep);
    }
    void setDS(char* DSp)
    {
        strcpy(DS, DSp);
    }

    void disp()
    {
        cout <<
"**********************************************************************************" << endl;
        cout << "ID:" << ID << endl << "firstname:" << first <<
setw(10) << "lastname" << last << endl << "sex:" << sex << endl;
        cout << "age:" << age << endl << "date:" << date.getyear() <<
"/" << date.getmonth() << "/" << date.getday() << endl;
        cout << "DS:" << DS << endl;
        cout <<
"**********************************************************************************" << endl;
    } //病历信息显示
    friend class PatientDataBase;
};
```

```
class PatientDataBase
{
    int nElem;//病历信息表中的元素个数
    int maxNum;//最多纪录
    MedicalRecord* MR;//指向病历信息的指针
public:
    PatientDataBase(int n = 15)
    {
        maxNum = n;
        if (n)
            MR = new MedicalRecord[n];
        else
            MR = 0;
        fstream in("patient.txt", ios::in | ios::binary);
        if (!in)
        {
            cout << "病例加载失败." << endl;
            exit(1);
        }
        int i = 0;
        char meanless[200];
        in.getline(meanless, 200);
        while (!in.eof())
        {
            in >> MR[i].ID >> MR[i].first >> MR[i].last >>
MR[i].sex >> MR[i].age >> MR[i].date.year >> MR[i].date.month;
            in >> MR[i].date.day >> MR[i].diagnose >> MR[i].advice >>
MR[i].DS;
            i++;
        }
        in.close();
        nElem = i;
    }   //构造函数，初始化病历信息表，将patien.txt文件中数据读到MR[]
中

    ~PatientDataBase()
    {
        delete[maxNum] MR;
    }//析构函数
    void writetofile()
    {
        fstream out("Patient.txt", ios::out);
        if (!out)
        {
            cout << "病例写入失败." << endl;
            exit(2);
        }
        else {
            out << "ID       lastname    firstname        sex   age
date              diagnose             advice              DS
";
            for (int i = 0; i < nElem; i++)
            {
                out << endl;
                out << setiosflags(ios::left) << setw(10) << MR[i].ID;
                out << setiosflags(ios::left) << setw(20) <<
MR[i].first;
                out << setiosflags(ios::left) << setw(20) <<
MR[i].last;
                out << setiosflags(ios::left) << setw(5) << MR[i].sex;
                out << setiosflags(ios::left) << setw(5) << MR[i].age;
                out << setiosflags(ios::left) << setw(5) <<
MR[i].date.year << setiosflags(ios::left) << setw(3) <<
MR[i].date.month << setiosflags(ios::left) << setw(7) <<
MR[i].date.day;
                out << setiosflags(ios::left) << setw(10) <<
MR[i].diagnose;
                out << setiosflags(ios::left) << setw(20) <<
MR[i].advice;
                out << setiosflags(ios::left) << setw(20) << MR[i].DS;

            }
            if (nElem == 0) cout << "无数据可写!" << endl;
            else cout << "数据写入成功!" << endl;
            out.close();
        }
    }
    void clear() //删除所有病历信息
    {
        char a;
        cout << "是否删除所有数据y/n?" << endl;
        cin >> a;
        if (a == 'y')
        {
            for (int i = 0; i <= nElem - 1; i++)
                strcpy((char*)(&MR[i]), "");
            nElem = 0;
            cout << "已清空! " << endl;
        }
    }
    void PatientAdd() //添加病历记录
    {
        int ID;
        char last[15];
        char first[10];
        char sex[5];
        int age;
        int a, b, c;
        char diagnose[20];
        char advice[20];
        char DS[20];
        ofstream outPatient("Patient.txt", ios::ate | ios::in |
ios::binary);//没有生成，有在文件尾追加ios::app|
        if (!outPatient)
        {
            cerr << "File could not be opened." << endl;
            exit(EXIT_FAILURE);
        }
        if (maxNum == nElem) cout << "已达到最大记录数无法继续添加!"
<< endl;
        else
        {
            cout << "请输入ID" << endl;
            cin >> ID;
            cout << "请输入姓名" << endl;
            cin >> first;
            cin >> last;
            cout << "请输入性别" << endl;
            cin >> sex;
            cout << "请输入年龄" << endl;
            cin >> age;
            cout << "请输入就诊日期" << endl;
            cout << "年:";
            cin >> a;
            cout << "月:";
            cin >> b;
            cout << "日:";
            cin >> c;

            cout << "请输入诊断" << endl;
            cin >> diagnose;
            cout << "请输入建议" << endl;
            cin >> advice;
            cout << "请输入医师姓名" << endl;
            cin >> DS;
            MR[nElem] = MedicalRecord(ID, first, last, sex, age, a, b, 
c, diagnose, advice, DS);
            nElem++;
            cout << "添加成功" << endl;
        }
    }
    void PatientDelete()//删除病历信息
    {
        char a;
        int j = getorder();
        if (j == -1)  cout << "查找失败，该病人不存在" << endl;
        else
        {
            cout << "是否确定删除y/n?" << endl;
            cin >> a;
            if (a == 'y')
            {
                for (; j < nElem; j++)
                {
                    MR[j].ID = MR[j + 1].ID;
                    strcpy(MR[j].first, MR[j + 1].first);
                    strcpy(MR[j].last, MR[j + 1].last);
                    strcpy(MR[j].sex, MR[j + 1].sex);
                    MR[j].age = MR[j + 1].age;
                    MR[j].date = MR[j + 1].date;
                    strcpy(MR[j].diagnose, MR[j + 1].diagnose);
                    strcpy(MR[j].advice, MR[j + 1].advice);
                    strcpy(MR[j].DS, MR[j + 1].DS);
                }
                cout << "删除成功" << endl;
                nElem -= 1;
            }
        }
    }
    void PatientChange()
    {
        char last[15];
        char first[10];
        char sexp[5];
        int agep;
        int a, b, c;
        char diagnosep[20];
        char advicep[20];
        char DSp[20];
        int i = getorder();
        if (i == -1) { cout << "查找失败，该病人不存在" << endl; }
        else
        {
            cout << "请输入姓名" << endl;
            cin >> first >> last;
            cout << "请输入性别" << endl;
            cin >> sexp;
            cout << "请输入年龄" << endl;
            cin >> agep;
            cout << "请输入日期" << endl;
            cout << "年:";
            cin >> a;
            cout << "月:";
            cin >> b;
            cout << "日:";
            cin >> c;
            cout << "请输入诊断" << endl;
            cin >> diagnosep;
            cout << "请输入建议" << endl;
            cin >> advicep;
            cout << "请输入医师姓名" << endl;
            cin >> DSp;
            MR[i].setfirst(first);
            MR[i].setlast(last);
            MR[i].setsex(sexp);
            MR[i].setage(agep);
            Date datep(a, b, c);
            MR[i].setdate(datep);
            MR[i].setdiagnose(diagnosep);
            MR[i].setadvice(advicep);
            MR[i].setDS(DSp);

            cout << "修改成功。" << endl;
        }
    }
    void queryID() //按照病历号查找，顺序查找
    {
        int ID;
        cout << "请输入ID" << ends;
        cin >> ID;
        for (int i = 0; i <= nElem - 1; i++)
        {
            if (MR[i].ID == ID)
            {
                MR[i].disp(); break;
            }

            if (i == nElem)
            {
                cout << "查找失败，该病人不存在" << endl;
            }
        }
    }
    void queryDS() //按照医生签名查找，折半查找
    {
        char DS[20];
        cout << "请输入DS" << ends;
        cin >> DS;
        int low = 0, mid, up = nElem - 1;
        while (low <= up)
        {
            mid = (low + up) / 2;
            if (strcmp(MR[mid].DS, DS) == 0)
            {
                MR[mid].disp(); low += 1;
            }
            else if (strcmp(MR[mid].DS, DS) == 1)
                up = mid - 1;
            else
                low = mid + 1;
        }
        if (low > up)
        {
            cout << "查找结束" << endl;
        }
    }
    void disp() //输出所有病历信息
    {
        if (nElem != 0)
        {
            for (int i = 0; i <= (nElem - 1); i++)
                MR[i].disp();
        }
        else
```

```
            cout << "没有病人记录! " << endl;

    }



    int getorder()
    {
        int IDp;
        cout << "请输入ID" << ends;
        cin >> IDp;
        int i = 0;
        while (i <= nElem)
        {
            if (MR[i].ID == IDp)
                break;
            i++;
        }
        if (i > nElem)
            return (-1);
        else
            return (i);
    }//用病人ID获得病人顺序

    char quit()
    {
        char a, b;
        cout << "是否退出y/n?" << endl;
        cin >> a;
        if (a == 'y')
        {
            cout << "是否输出成文本y/n?" << endl;
            cin >> b;
            if (b == 'y')
            {
                writetofile();
            }
            cout << "再见，欢迎再次光临." << endl;
        }
        return (a);
    }//功能0，退出

};
    #endif
```

Main.cpp

```
#include <iostream>
#include<fstream>
#include<array>
#include<cstdlib>
#include<iomanip>
#include<vector>
#include<iterator>
#include<windows.h>
#include "Doctor.h"
#include"Patient.h"
using namespace std;
#define _CRT_SECURE_NO_WARNINGS
#pragma warning(disable:4996)

string HSaddress = " CODE county, C++ city";
string HSname = "DEBUG HOSPITAL";
int HSyear = 1983;
int HSmonth = 01;
int HSday = 01;//医院资料

int id;
char first[11];
char last[16];
int sex;
int job;
int byear, bmonth, bday;
int syear, smonth, sday;
const int D[13] = { 0,31,28,31,30,31,30,31,31,30,31,30,31 };//days
vector<int>doc = { 0 };

bool checkdocid(int a)//检查 id 是否存在
{
    for (int i = 0; i < doc.size(); i++)
    {
        if (doc[i] == a)
            return true;//id 存在返回真
    }
    return false;
}
void inputid(int a)
{
    for (int i = 0; i < doc.size(); i++)
    {
        if (doc[i] == 0)//把 0 元素替换，节省空间
        {
            doc[i] = a;
            return;
        }
    }
    doc.push_back(a);//当前面一直没有空位时再进行补位
    return;
}
bool leapyear(int t)//判断是闰年
{
    if (t % 400 == 0 || (t % 100 != 0 && t % 4 == 0))return true;
    else return false;
}

void DoctorAdd()
{
    cout << "Please enter a new ID number: " << endl;
    cin >> id;
    while (id == 0 || checkdocid(id))
    {
        cout << "Wrong input. Please enter the RIGHT number: ";
        cin >> id;
    }
    if (id)
    {
        inputid(id);
    }//确保输入 id 有效，放到 doc 中

    Doctor d;
    ofstream outHospital("Hospital.txt", ios::ate | ios::in | ios::binary);
    if (!outHospital)
    {
        cerr << "File could not be opened." << endl;
        exit(EXIT_FAILURE);
    }
    cout << "Enter the firstname,lastname: " << endl;
    cin >> first >> last;

    cout << "Enter the year of birthday (in NUMBERS): ";//输入生日
    cin >> byear;
    while (!(byear >= 1900 && byear <= 2010))
    {
       cout << "Year is out of range for current month and year. Input the right one:\n";
        cin >> byear;
    }
    cout << "Enter the month of birthday (in NUMBERS): ";
    cin >> bmonth;
    while (!(bmonth >= 1 && bmonth <= 12))
    {
        cout << "Month is out of range for current month and year. Input the right
one:\n";
        cin >> bmonth;
    }
    cout << "Enter the day of birthday (in NUMBERS): ";
    cin >> bday;
    while (!(bmonth == 2 && leapyear(byear) && bday >= 1 && bday <= 29)
&& !(bday >= 1 && bday <= D[bmonth]))//闰 2 月或正常
    {
        cout << "Day is out of range for current month and year. Input the right one: 
\n";
        cin >> bday;
    }
    cout << "Enter the year of job starting date(in NUMBERS): ";//输入入职时间
    cin >> syear;
    while (!(syear >= 1920 && syear <= 2019))
    {
       cout << "Year is out of range for current month and year. Input the right one:\n";
        cin >> syear;
    }
    cout << "Enter the month of job starting date (in NUMBERS): ";
    cin >> smonth;
    while (!(smonth >= 1 && smonth <= 12))
    {
        cout << "Month is out of range for current month and year. Input the right one: 
\n";
        cin >> smonth;
    }
    cout << "Enter the day of job starting date(in NUMBERS): ";
    cin >> sday;
    while (!(smonth == 2 && leapyear(syear) && sday >= 1 && sday <= 29) 
&& !(sday >= 1 && sday <= D[bmonth]))
    {
        cout << "Day is out of range for current month and year. Input the right one: 
\n";
        cin >> sday;
    }

    cout << "Sex? (1 for male, 2 for female): ";
    cin >> sex;
    while (sex != 1 && sex != 2)
    {
        cout << "Wrong input. Please enter the RIGHT number: ";
        cin >> sex;
    }
    cout << "Job?  呼吸内科请输入 11;     消化内科请输入 12;     神经内科请输入
13;   心血管内科请输入 14\n"
        << "       普通外科请输入 21;     神经外科请输入 22;     心胸外科请输入
23;   骨外科请输入 24\n"
        << "       耳鼻喉科请输入  31;     眼科请输入  32;           口腔科请输入
33\n"
        << "       中医保健科请输入 41;  中医按摩科请输入 42;  中医全科请输入
43\n";
    cin >> job;
    while (job != 11 && job != 12 && job != 13 && job != 14 && job != 21 &&
job != 22 && job != 23 && job != 24
        && job != 31 && job != 32 && job != 33 && job != 41 && job != 42 &&
job != 43)
    {
        cout << "Wrong input. Please enter the RIGHT number: ";
        cin >> job;
    }//所有信息添加完成
    d.setbirthdate(bmonth, byear);
    d.setbday(bday);
    d.setstartdate(smonth, syear);
    d.setsday(sday);
    d.setfirst(first);
    d.setlast(last);
    d.setid(id);
    d.setsex(sex);
    d.setjob(job);
    outHospital.seekp((d.getid() - 1) * sizeof(Doctor));
    outHospital.write(reinterpret_cast<const char*>(&d), sizeof(Doctor));
    outHospital.close();
    cout << "Add finished.\n" << endl;
```

```
}
void DoctorChange()
{
        int changeid;
        Doctor d;
        int choice;
        cout << "Enter the ID to change(0 to end changing process):    \n";
        cin >> changeid;
        if (changeid == 0)return;
        if (changeid != 0)
        {
                while (!checkdocid(changeid))//确保 ID 输入正确,id 不存在时启动函数
                {
                        cout << "Wrong input. Please enter the RIGHT number: ";
                        cin >> changeid;
                }
                fstream inoutdoctor("Hospital.txt", ios::in | ios::ate | ios::out | ios::binary);//指
针定位文件尾
                if (!inoutdoctor)
                {
                        cerr << "File could not be opened." << endl;
                        exit(EXIT_FAILURE);
                }
                cout << "Which information do u wanna change?\n"
                        << " Enter 1 for first name; Enter 2 for last name; Enter 3 for job(0 TO
END):    " << endl;

                cin >> choice;
                while (choice != 1 && choice != 2 && choice != 3 && choice != 0)
                {
                        cout << "wrong input. enter the right number: ";
                        cin >> choice;
                }
                switch (choice)
```

```
{
case 0:
{
      changeid = 0;
      cout << "CANCELLED\n";
      break; }
case 1://first name
{
      Doctor p;
      inoutdoctor.seekg((changeid - 1) * sizeof(Doctor));
      inoutdoctor.read(reinterpret_cast<char*>(&d), sizeof(Doctor));
      cout << "Input new first name: ";
      cin >> first;
      p.setfirst(first);
      p.setlast(d.getlast());
      p.setbirthdate(d.getbmonth(), d.getbyear());
      p.setbday(d.getbday());
      p.setstartdate(d.getsmonth(), d.getsyear());
      p.setsday(d.getsday());
      p.setid(changeid);
      p.setsex(d.getsex());
      p.setjob(d.getjob());
      inoutdoctor.seekp((changeid - 1) * sizeof(Doctor));
      inoutdoctor.write(reinterpret_cast<const char*>(&p), sizeof(Doctor));
inoutdoctor.close();
      //cout << "which information do u wanna change?(0 to end changing
process)\n"
      //    << "1:firstname; 2:lastname; 3:job;(0 TO END)" << endl;
      //cin >> choice;
      break;
}
case 2://last name
{
      Doctor q;
```

```
            inoutdoctor.seekg((changeid - 1) * sizeof(Doctor));
            inoutdoctor.read(reinterpret_cast<char*>(&d), sizeof(Doctor));
            cout << "Input new last name: ";
            cin >> last;
            q.setlast(last);
            q.setfirst(d.getfirst());
            q.setbirthdate(d.getbmonth(), d.getbyear());
            q.setbday(d.getbday());
            q.setstartdate(d.getsmonth(), d.getsyear());
            q.setsday(d.getsday());
            q.setid(changeid);
            q.setsex(d.getsex());
            q.setjob(d.getjob());
            inoutdoctor.seekp((changeid - 1) * sizeof(Doctor));
            inoutdoctor.write(reinterpret_cast<const    char*>(&q),    sizeof(Doctor));
inoutdoctor.close();
            //cout << "which information do u wanna change?(0 to end changing
process)\n"
            //<< "1:firstname; 2:lastname; 3:job;(0 TO END)" << endl;
            //cin >> choice;
            break;
        }
        case 3://job
        {
            Doctor r;
            inoutdoctor.seekg((changeid - 1) * sizeof(Doctor));
            inoutdoctor.read(reinterpret_cast<char*>(&d), sizeof(Doctor));
            cout << "Input new job number:\n"
                << "呼吸内科请输入 11;     消化内科请输入 12;     神经内科请输
入 13;   心血管内科请输入 14\n"
                << "普通外科请输入 21;     神经外科请输入 22;     心胸外科请输
入 23;   骨外科请输入 24\n"
                << "耳鼻喉科请输入 31;     眼科请输入 32;           口腔科请输入
33\n"
```

```
                <<"中医保健科请输入41; 中医按摩科请输入42; 中医全科请输
入 43\n";
            cin >> job;
            while (job != 11 && job != 12 && job != 13 && job != 14 && job != 21
&& job != 22 && job != 23 && job != 24
                && job != 31 && job != 32 && job != 33 && job != 41 && job !=
42 && job != 43)
            {
                cout << "Wrong input. Please enter the RIGHT number: ";
                cin >> job;
            }
            r.setfirst(d.getfirst());
            r.setlast(d.getlast());
            r.setbirthdate(d.getbmonth(), d.getbyear());
            r.setbday(d.getbday());
            r.setstartdate(d.getsmonth(), d.getsyear());
            r.setsday(d.getsday());
            r.setid(changeid);
            r.setsex(d.getsex());
            r.setjob(job);
            inoutdoctor.seekp((changeid - 1) * sizeof(Doctor));
            inoutdoctor.write(reinterpret_cast<const  char*>(&r),  sizeof(Doctor));
inoutdoctor.close();
            //cout << "which information do u wanna change?(0 to end changing
process)\n"
            //<< "1:firstname; 2:lastname; 3:job(0 TO END)" << endl;
            //cin >> choice;
            break;
        }
        default: break;
        }
    }
    cout << "CHANGED\n\n";
    return;
}
void DoctorDelete()
{
    int deletenumber;
    Doctor d;
    Doctor blank;
    cout << "Enter the ID number to delete: ";
    cin >> deletenumber;
    while (!checkdocid(deletenumber) || deletenumber == 0)
    {
        cout << "Wrong input. Please enter the RIGHT number: ";
        cin >> deletenumber;
    }
    if (checkdocid(deletenumber))//把 check 容器中相应数字删去
    {
        for (int i = 0; i < doc.size(); i++)
        {
            if (doc[i] == deletenumber)doc[i] = 0;
        }
    }
    fstream inoutDoctor("Hospital.txt", ios::in | ios::ate | ios::out | ios::binary);
    if (!inoutDoctor)
    {
        cerr << "File could not be opened." << endl;
        exit(EXIT_FAILURE);
    }
    inoutDoctor.seekg((deletenumber - 1) * sizeof(Doctor));
    inoutDoctor.read(reinterpret_cast<char*>(&d), sizeof(Doctor));
    blank.setfirst("DELETED");
    blank.setlast("DELETED");
    inoutDoctor.seekp((deletenumber - 1) * sizeof(Doctor));
    inoutDoctor.write(reinterpret_cast<const char*>(&blank), sizeof(Doctor));
    inoutDoctor.close();
    cout << "DELETED\n\n";
}
void DoctorPrint()
{
    Doctor m;
    ifstream inhospital("Hospital.txt", ios::in | ios::binary);
    if (!inhospital)
    {
        cerr << "File could not be opened." << endl;
        exit(EXIT_FAILURE);
    }
    //cout << "****************" << HSname << "**************" << endl;
    //cout << "                                                  ----" << "Located in " <<
HSaddress << endl;
    //cout << "The hospital was founded in " << HSyear << "--" << HSmonth << "--"
<< HSday << endl;
    cout << "\n-----------Basic information of all Doctors------------\n" << left <<
setw(10) << "ID"
        << setw(11) << "First Name" << setw(16) << "Last Name"
        << setw(9) << "Sex" << setw(12) << "Job" << setw(6) << right << fixed <<
showpoint << "Birthday(Y--M--D)"
        << setw(8) << "     Hire Date(Y--M--D)" << endl;

    inhospital.read(reinterpret_cast<char*>(&m), sizeof(Doctor));
    while (inhospital && !inhospital.eof())
    {
        if (m.getid() != 0)
        {
            cout << left << setw(10) << m.getid() << setw(11) << m.getfirst()
                << setw(16) << m.getlast() << setw(9) << (m.getsex() == 1 ? "Male" :
"Female") << setw(12);
            switch (m.getjob())
            {
            case 11:
```

```
{
    cout << "呼吸内科";
    break;
}
case 12:
{
    cout << "消化内科";
    break;
}
case 13:
{
    cout << "神经内科";
    break;
}
case 14:
{
    cout << "心血管内科";
    break;
}
case 21:
{
    cout << "普通外科";
    break;
}
case 22:
{
    cout << "神经外科";
    break;
}
case 23:
{
    cout << "心胸外科";
    break;
}
```

```
case 24:
{
    cout << "骨外科";
    break;
}
case 31:
{
    cout << "耳鼻喉科";
    break;
}
case 32:
{
    cout << "眼科";
    break;
}
case 33:
{
    cout << "  口腔科";
    break;
}
case 41:
{
    cout << "中医保健科";
    break;
}
case 42:
{
    cout << "中医按摩科";
    break;
}
case 43:
{
    cout << "中医全科";
    break;
```

```
            }
            default:break;
            }
            cout << setw(6) << right << fixed
                << showpoint << m.getbyear() << "--" << m.getbmonth() << "--" <<
m.getbday()
                << setw(15) << m.getsyear() << "--" << m.getsmonth() << "--" <<
m.getsday() << endl;

        }
        inhospital.read(reinterpret_cast<char*>(&m), sizeof(Doctor));
    }
    inhospital.close();
}
void cleandocdata()
{
    cout << "Input the code: ";
    int code;
    cin >> code;
    if (code != 111111)
    {
        cout << "U don't have the right.\n";
        return;
    }
    if (code == 111111)
    {
        fstream inoutDoctor("Hospital.txt", ios::out | ios::binary);
        if (!inoutDoctor)
        {
            cerr << "File could not be opened." << endl;
            exit(EXIT_FAILURE);
        }
        inoutDoctor.close();
        cout << "ALL data is DELETED.\n\n";
```

```
	}
}
void setVectordoc()//把每次打开以后的文件已有 id 存入 doc
{
	ifstream inDoctor("Hospital.txt", ios::in | ios::binary);
	if (!inDoctor)
	{
		ofstream doctor("Hospital.txt");//用来判断文件是否存在，不存在则创建
	}
	Doctor m;
	inDoctor.read(reinterpret_cast<char*>(&m), sizeof(Doctor));
	while (inDoctor && !inDoctor.eof())
	{
		if (m.getid() != 0)
		{
			doc.push_back(m.getid());
			sort(doc.begin(), doc.end());//删除 doc 中的重复元素
			vector<int>::iterator ite = unique(doc.begin(), doc.end());
			doc.erase(ite, doc.end());
		}
		inDoctor.read(reinterpret_cast<char*>(&m), sizeof(Doctor));
	}
}

int main()
{
	int n = 0;
	cout << "如果您想处理医生的信息，请输入 1.\n" << "如果您想处理患者的信
息，请输入 2。" << endl;
	cin >> n;
	if (n == 1)
	{
		setVectordoc();
```

```
        cout << "\n\n\n\n\n                              " << HSname << endl;
        cout << "                                                               --
--" << "Located in " << HSaddress << endl;
        cout << "The hospital was founded in " << HSyear << "--" << HSmonth << "-
-" << HSday << endl;
        cout << "\t\t ★**★** Welcome to the Hospital Information Management
System **★**★" << endl;
        cout << "\n";
        cout << "\t\t                  ☆********★**********☆***********★
********☆\n";
        cout   <<   "\t\t                                                    ★
★\n";
        cout   <<   "\t\t                                                    ☆
☆\n";
        cout << "\t\t                  ★       1.Add                   2.Change
★\n";
        cout << "\t\t                  ☆       3.Delete                  4.Print
☆\n";
        cout   <<   "\t\t                                                    ☆
☆\n";
        cout   <<   "\t\t                                                    ★
★\n";
        cout << "\t\t                  ☆★☆★☆★**********☆***********★
☆★☆★☆\n";

        int choose = 0;
        cin >> choose;
        system("cls");
        while (choose != -1)
        {
            switch (choose)
            {
            case 1:
            {
```

```
			DoctorAdd(); break;
		}
		case 2:
		{
			DoctorChange(); break;
		}
		case 3:
		{
			DoctorDelete(); break;
		}
		case 4:
		{
			DoctorPrint(); break;
		}
		case 5:
		{
			cleandocdata(); break;//
		}
		default: break;
		}
		cout << endl << endl << endl;
		cout << "\t\t ★ ** ★ ** Welcome to the Hospital Information
Management System **★**★" << endl;
		cout << "\t\t                ☆********★**********☆***********★
********☆\n";
		cout    <<    "\t\t                                                    ★
★\n";
		cout    <<    "\t\t                                                    ☆
☆\n";
		cout << "\t\t          ★        1.Add                    2.Change
★\n";
		cout << "\t\t            ☆        3.Delete                   4.Print
☆\n";
		cout << "\t\t                 ☆                         5.clean  data
```

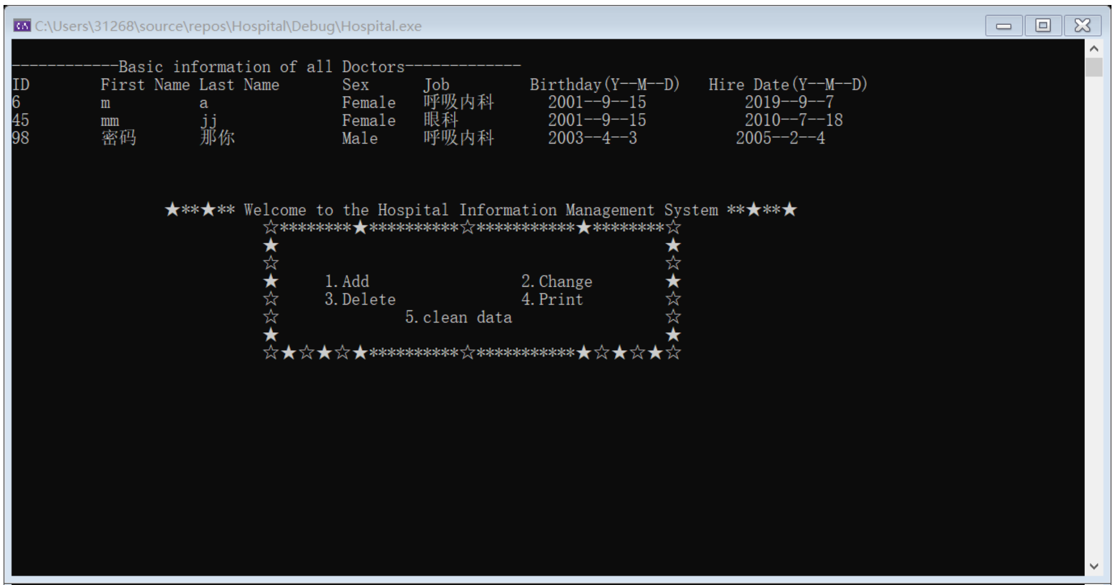
```
☆\n";
            cout    <<    "\t\t                                             ★
★\n";
            cout << "\t\t           ☆★☆★☆★*********☆**********★
☆★☆★☆\n";
            cin >> choose;
            system("cls");
        }
        return 0;
        system("pause");
    }
    else
    {
        PatientDataBase a;
        int n, q;
        char s;
        for (;;)
        {
            system("cls");

            cout << "\n\n\n\n\n                          " << HSname << endl;
            cout                              <<                              "
----" << "Located in " << HSaddress << endl;
            cout << "The hospital was founded in " << HSyear << "--" << HSmonth
<< "--" << HSday << endl;
            cout  <<  "\t\t  ★ ** ★ **  Welcome  to  the  Hospital  Information
Management System **★**★" << endl;
            cout << "\t\t           ☆*******★*********☆**********★
*******☆\n";
            cout << "                    1  删除所有病例信息        " <<
endl;
            cout << "                    2  添加病历记录            " <<
endl;
            cout << "                    3  单个删除指定病历信息    " <<
```

```
endl;
        cout << "                    4   单个修改指定病例信息    " <<
endl;
        cout << "                    5   查找指定病例信息        " <<
endl;
        cout << "                    6   显示病历信息            " <<
endl;
        cout << "                    7   导出病历信息            " <<
endl;
        cout << "                    0   退出程序                " <<
endl;
        cout << "\t\t          ☆★☆★☆★*********☆**********★
☆★☆★☆\n";

        cin >> n;
        switch (n)
        {
        case 1:a.clear();
            system("pause"); break;
        case 2:a.PatientAdd();
            system("pause");
            break;
        case 3:a.PatientDelete();
            system("pause");
            break;
        case 4:a.PatientChange();
            system("pause");
            break;
        case 5:cout << "请选择查询方式(0:ID,1:医生签名)" << endl;
            cin >> q;
            if (q)
                a.queryDS();
            else
                a.queryID();
            system("pause");
            break;
        case 6:a.disp();
            system("pause"); break;
        case 7:a.writetofile();
            system("pause"); break;
        case 0:if (a.quit() != 'y')
            continue;
                else break;
        }
        if (n == 0) break;
    }
    cout << "按任意键退出程序，谢谢使用" << endl;
    a.~PatientDataBase();
    exit(0);
}
}
```

Result:

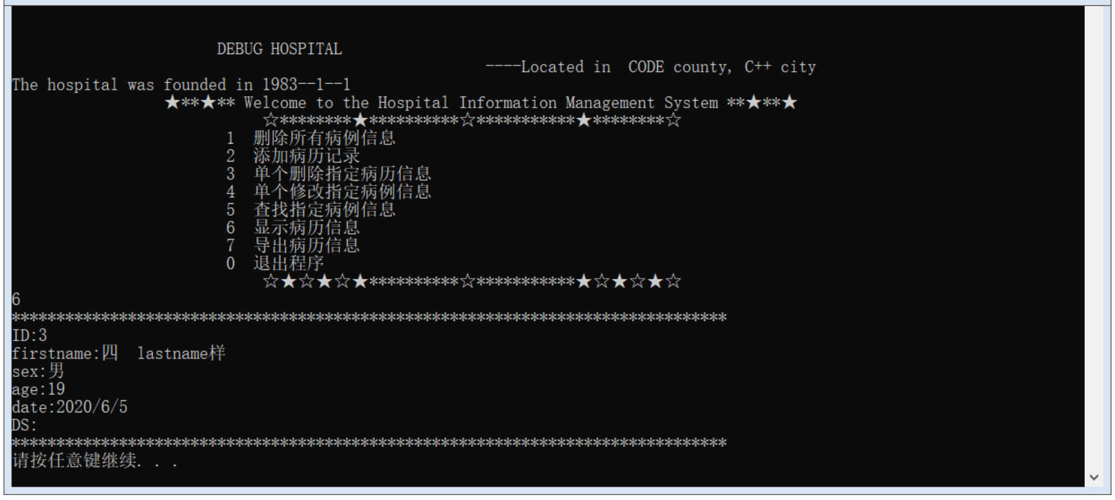


C:\Users\31268\source\repos\Hospital\Debug\Hospital.exe

```
-----------Basic information of all Doctors-------------
ID        First Name Last Name       Sex      Job         Birthday(Y--M--D)   Hire Date(Y--M--D)
6         m          a               Female   呼吸内科      2001--9--15          2019--9--7
45        mm         jj              Female   眼科          2001--9--15          2010--7--18
98        密码       那你            Male     呼吸内科      2003--4--3          2005--2--4



                 ★**★** Welcome to the Hospital Information Management System **★**★
                            ☆********★**********☆***********★********☆
                            ★                                          ★
                            ☆                                          ☆
                            ★      1.Add                 2.Change      ★
                            ☆      3.Delete              4.Print       ☆
                            ☆               5.clean data               ☆
                            ★                                          ★
                            ☆★☆★☆★**********☆***********★☆★☆★☆
```



```
                       DEBUG HOSPITAL
                                                        ----Located in  CODE county, C++ city
The hospital was founded in 1983--1--1
                 ★**★** Welcome to the Hospital Information Management System **★**★
                            ☆********★**********☆***********★********☆
                        1  删除所有病例信息
                        2  添加病历记录
                        3  单个删除指定病历信息
                        4  单个修改指定病例信息
                        5  查找指定病例信息
                        6  显示病历信息
                        7  导出病历信息
                        0  退出程序
                            ☆★☆★☆★**********☆***********★☆★☆★☆
6
********************************************************************************
ID:3
firstname:四  lastname样
sex:男
age:19
date:2020/6/5
DS:
********************************************************************************
请按任意键继续. . .
```

Chapter 5　Discussion

(1) This time it took too much time to implement the logic.
(2)Instead of integrating the Doctor class and the Patient class, it would be better to be a derived class.
(3)The output design has some imperfections, such as the default hire D ate being later than birthday.

Chapter 6　Conclusion

Through this big homework, we have a deeper understanding of the kno wledge, but also more proficient in object-oriented programming.

But at the same time, we encountered a lot of problems in the process, t hrough solving these problems, we also carried out some corresponding r eflection and summary.

Reference:

[1] 9.18, , 《C++ How to Program, Ninth Edition》, 2019
[2] 14.14, 《C++ How to Program, Ninth Edition》, 2019
[3]《C++自动创建文件夹》， CSDN, 2015
[4] 《病例信息管理系统》，CSDN—Jeremy Zhangsiyu