目录

[一、目的 1](#_Toc518588725)

[二、课程设计环境： 1](#_Toc518588726)

[三、系统总体设计 1](#_Toc518588727)

[3.1 问题描述 1](#_Toc518588728)

[3.2 功能要求 1](#_Toc518588729)

[3.3 思路步骤 2](#_Toc518588730)

[3.4 类模块 2](#_Toc518588731)

[3.4.1 类名 2](#_Toc518588732)

[3.4.2类关系 2](#_Toc518588733)

[3.4.3 类成员函数 3](#_Toc518588734)

[3.5 结构设计 3](#_Toc518588735)

[3.5.1 系统流程框图： 3](#_Toc518588736)

[3.5.2 界面设计模块 3](#_Toc518588737)

[3.5.3 代码设计特点与创新： 4](#_Toc518588738)

[3.6 Star UML图形设计 4](#_Toc518588739)

[3.6.1 用例图 4](#_Toc518588740)

[3.6.2 时序图 5](#_Toc518588741)

[3.6.3 类图 6](#_Toc518588742)

[3.7 系统测试 6](#_Toc518588743)

[3.7.1 实验代码 6](#_Toc518588744)

[3.7.2 模块测试 26](#_Toc518588745)

[四、程序设计过程遇到的问题 31](#_Toc518588746)

[五、感想与体会 32](#_Toc518588747)

[附参考文献 32](#_Toc518588748)

**诊所信息管理系统设计**

# ****一、目的****

**1.课程设计目的：**

**（1）巩固学生面向对象程序设计知识，加强学生实际动手能力和提高学生综合素质。**

**（2）培养学生面向对象设计和开发能力，熟练掌握一种编译器的使用。**

**（3）通过设计和开发实际的应用小系统，提高学生的分析问题和解决问题的能力。**

2.实验目的：

完成小型诊所的简单信息管理程序，掌握类的设计以及继承和派生的应用。

# ****二、课程设计环境****

**软件：Visual Studio 2015、Star UML**

**硬件：手提电脑**

# ****三、系统总体设计****

## **3.1 问题描述**

**设计人类（Person类）和医生类（Doctor类），在此基础上，通过增加患者和账单，使它们公用于表示一家诊所的信息管理。**

## **3.2 功能要求**

**（1）在一条医生记录中，包括医生的专业说明（specialty），如内科医生（surgeon）、儿科医生（pediatrician） 、产科医生（obstetrician）及全科医生（general practitioner）。**

**（2）Doctor记录还含有诊费（off\_vist\_fee）。**

**（3）在一条患者记录中，包括该患者产生的药费(drug\_fee) ，患者的诊费（即医生的诊费）。**

**（4）在一条账单记录中，包括一条患者对象、该患者对应得主治医生、该患者产生的诊费和药费。**

**（5）应用程序能够显示出诊所中每个患者的信息和对应主治医生的信息。**

**（6）能够统计出所有患者的总费用。**

**（7）保存功能：可将当前系统中各类记录存入文件中，存入方式任意。**

**（8）读取功能：可将保存在文件中的信息读入到当前系统中，供用户进行使用。**

**（9）普通用户只能查询医生信息--不用登陆，要想维护该系统需要管理员登陆。【自己额外增加的功能】**

## **3.3 思路步骤**

**根据系统功能要求，设计步骤如下：**

**系统分析，建立该系统的框图和界面的设计；**

1. **分析系统中的每个实体及它们之间的关系；**
2. **根据系统功能描述，建立系统大致流程框图；**
3. **完成系统中各个类的描述；**
4. **各个类的实现；**
5. **实现系统的应用，完成代码；**
6. **功能调试与优化；**
7. **完成课程设计报告。**

## **3.4 类模块**

## **3.4.1 类名**

**Person人类：姓名、性别、年龄**

**Doctor医生类：姓名、性别、年龄、医生专业、医生诊费**

**Patient患者类：姓名、性别、年龄、患者药费、患者诊费**

**Bill账单类：姓名、性别、年龄、患者药费、患者诊费、主治医生姓名、主治医生性别、主治医生年龄、主治医生专业、主治医生诊费**

## **3.4.2类关系**

**Patient类继承于Person类**

**Doctor类继承于Person类**

**Bill类继承于Patient类**

## **3.4.3 类成员函数**

**Doctor医生类： D\_add（）【增加医生信息】**

**D\_select（）【查找某一医生信息】**

**D\_all（）【查看所有医生信息】**

**Patient患者类：P\_add（）【增加患者信息】**

**P\_select（）【查找某一患者信息】**

**P\_all（）【查看所有患者信息】**

**Bill账单类： B\_add（）【增加患者账单信息】**

**B\_select（）【查找某一患者账单信息】**

**B\_all（）【查看所有患者账单信息】】**

**B\_all（）【统计所有患者总费用】**

## **3.5 结构设计**

## **3.5.1 系统流程框图：**

诊所信息管理系统

管理员

普通用户

查看所有医生信息

按名查找医生

患者操作

医生操作

账单操作

统计所有患者总费用

查看所有账单信息

按患者名查找账单

增加患者信息

查看所有患者信息

按名查找患者信息

增加患者信息

查看所有医生信息

按名查找医生信息

增加医生信息

## **3.5.2 界面设计模块**

**1、登陆界面：1. 普通用户查看 2.管理员登陆 0.结束程序**

**2、普通用户查询界面：1.查询所有的医生信息 2.查找某一医生信息 3.返回上一层**

**3、管理员登陆后界面：1.医生操作 2.患者操作 3.患者账单操作 4.修改登陆密码 5.退出登陆**

**4、医生操作界面：1.查询所有的医生信息 2.按名查找医生 3.增加医生 4.返回上一层**

**5、患者操作界面：1.查询所有的患者信息 2.按名查找患者 3.增加患者信息 4.返回上一层**

**6、账单操作界面：1.查询所有的患者账单 2.查找某个患者账单 3.增加患者账单 4.计算所有患者的总费用 0.返回上一层**

## **3.5.3 代码设计特点与创新**

**1．用户界面**

**①在界面函数里写进了一个系统的清屏函数：system（”cls”）,这样能使界面更加友好，不会在前面出现一连串的命令和结果。**

**②界面函数独立出来，不放进类成员函数或主函数里，使代码阅读性更强和减少代码冗余。**

**2．代码**

**①在增加账单信息时，如果新增账单中患者的信息在患者文档没有重复的，则自动写入患者文档中**

## **3.6 Star UML图形设计**

## **3.6.1 用例图**

**用例图（Use Case Diagram）主要用来描述“用户、需求、系统功能单元”之间的关系。它展示了一个外部用户能够观察到的系统功能模型图。**

**从上图可知道：使用系统的为两类人：**

**普通用户。普通用户只能查看系统里的医生信息。**

**管理员。管理员能通过登陆进行系统维护，其中包括：**

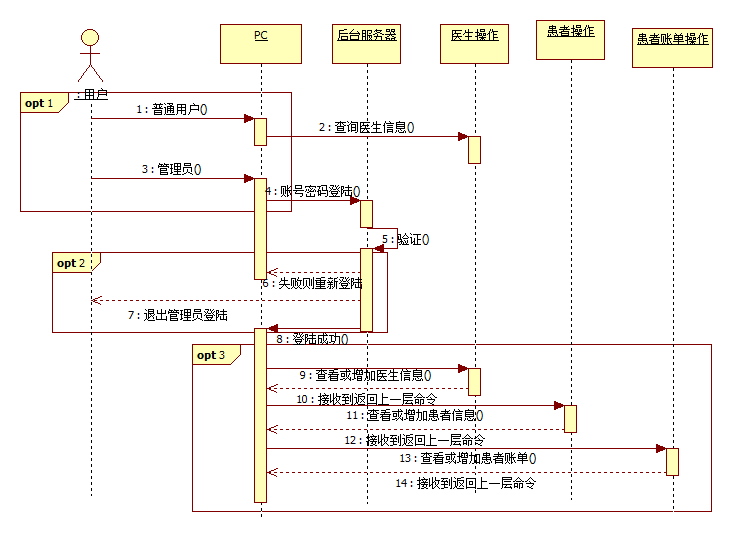
**医生操作：①查看所有医生信息 ②查找某一医生信息 ③增加医生信息**

**患者操作：①查看所有患者信息 ②查找某一患者信息 ③增加患者信息**

**患者账单操作：①查看所有账单信息 ②查找某一患者账单信息**

**③增加账单信息 ④统计所有患者总费用**

## **3.6.2 时序图**

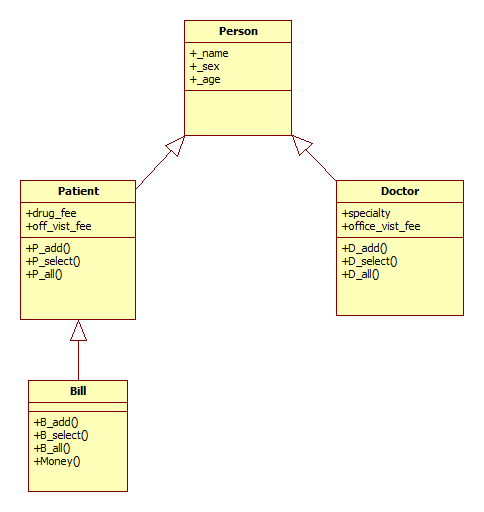


**时序图（Sequence Diagram）是显示对象之间交互的图，这些对象是按时间顺序排列的。时序图中显示的是参与交互的对象及其对象之间消息交互的顺序。**

**从图中我们能知道：系统执行的顺序如下：**

**系统开始要选择身份，如果选择普通用户，则可以直接查看医生信息。如果选择管理员，则要输入登陆账号和密码，后台验证账号密码错误则需重新登陆或退出管理员登陆，验证登陆成功则可以在医生操作、患者操作和患者账单操作里选择进行操作。**

## **3.6.3 类图**



**类图（Class Diagram）用于描述系统中所包含的类以及它们之间的相互关系，帮助简化对系统的理解。**

**从上图中我们能知道：**

**该系统有四个类，分别为Person类、Doctor类、Patient类和Bill类。它们之间的关系是：Doctor类和Patient类都继承于Person类，而Bill类继承于Patient类。**

## **3.7 系统测试**

## **3.7.1 实验代码**

**1.类模块、界面函数模块**

**void mainmenu();**

**void usermenu();**

**void menu();**

**void Dmenu();**

**void Pmenu();**

**void Bmenu();**

**void land(int& flag);**

**void changekey();**

**class Person//人类**

**{**

**public:**

**Person() {}**

**Person(string name, string sex, int age);**

**~Person() {}**

**protected:**

**string \_name;**

**string \_sex;**

**int \_age;**

**};**

**class Doctor:public Person//医生类**

**{**

**public:**

**Doctor() {}**

**Doctor(string Dname, string Dsex,int Dage,string Specialty, double vist\_fee);**

**~Doctor() {};**

**void D\_add();//增加医生**

**void D\_select();//选择查找某一名医生**

**void D\_all();//查看所有医生**

**protected:**

**string specialty;//医生专业**

**double off\_vist\_fee;//诊费**

**};**

**class Patient :public Person//患者类（继承于人类）**

**{**

**public:**

**Patient() {};**

**Patient(string name, string sex, int age, double Drug\_fee, double vist\_fee);**

**~Patient() {};**

**void P\_add();//增加患者**

**void P\_select();//查找某一患者**

**void P\_all();//查看所有患者**

**protected:**

**double drug\_fee;**

**double off\_vist\_fee;//诊费**

**};**

**class Bill :public Patient//账单类**

**{**

**public:**

**Bill() {}**

**Bill(string name, string sex, int age, double Drug\_fee, double vist\_fee, string Dname, string Dsex, int Dage, string Specialty, double Dvist\_fee);**

**~Bill() {}**

**void B\_add();//增加患者账单**

**void B\_select();//查找某一患者账单**

**void B\_all();//查看所有患者账单**

**void Money();//计算所有患者总费用**

**protected:**

**string D\_name;**

**string D\_sex;**

**int D\_age;**

**string specialty;**

**double Doff\_vist\_fee;**

**};**

**2.类模块及界面模块的函数实现：**

**#include <iostream>**

**#include <fstream>**

**#include<string>**

**#include<windows.h>**

**#include<iomanip>**

**#include"class.h"**

**using namespace std;**

**void mainmenu()**

**{**

**system("cls");**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "1.普通用户查看" << endl;**

**cout << "2.管理员登陆" << endl;**

**cout << "0.结束程序" << endl;**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**}**

**void usermenu()**

**{**

**system("cls");**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "1.查询所有的医生信息" << endl;**

**cout << "2.按名查找医生" << endl;**

**cout << "3.返回上一层" << endl;**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**}**

**void menu()**

**{**

**system("cls");**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "1.医生操作" << endl;**

**cout << "2.患者操作" << endl;**

**cout << "3.患者账单操作" << endl;**

**cout << "4.修改登陆密码" << endl;**

**cout << "5.退出登陆" << endl;**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**}**

**void Dmenu()**

**{**

**system("cls");**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "1.查询所有的医生信息" << endl;**

**cout << "2.按名查找医生" << endl;**

**cout << "3.增加医生" << endl;**

**cout << "4.返回上一层" << endl;**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**}**

**void Pmenu()**

**{**

**system("cls");**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "1.查询所有的患者信息" << endl;**

**cout << "2.按名查找患者" << endl;**

**cout << "3.增加患者信息" << endl;**

**cout << "4.返回上一层" << endl;**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**}**

**void Bmenu()**

**{**

**system("cls");**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "1.查询所有的患者账单" << endl;**

**cout << "2.查找某个患者账单" << endl;**

**cout << "3.增加患者账单" << endl;**

**cout << "4.计算所有患者的总费用" << endl;**

**cout << "0.返回上一层" << endl;**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**}**

**void land(int& flag)**

**{**

**system("cls");**

**string username;**

**string key;**

**string k[2];**

**fstream sfile("d:\\key.txt", ios::in);**

**sfile >> k[0] >> k[1];**

**cout << "请输入账号：";**

**cin >> username;**

**cout << "请输入密码：";**

**cin >> key;**

**if (k[0] == username && k[1] == key)**

**{**

**cout << endl;**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\*\*\*\*\*\*\*\*\*\* 登陆成功！ \*\*\*\*\*\*\*\*\*" << endl;**

**cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;**

**Sleep(1000);**

**flag = 1;**

**}**

**else**

**{**

**int in;**

**system("cls");**

**cout << "账号或密码输入错误！是否重新输入？" << endl;**

**cout << "1.重新输入" << endl;**

**cout << "2.退出" << endl;**

**cout << "请输入你的操作：" << endl;**

**cin >> in;**

**switch (in)**

**{**

**case 1: {**

**land(flag);**

**}break;**

**case 2: {}break;**

**default:**

**cout << "输入错误！请重新输入！" << endl;**

**Sleep(500);**

**break;**

**}**

**}**

**sfile.close();**

**}**

**void changekey()**

**{**

**string key;**

**string k[3];**

**fstream ifile("d:\\key.txt", ios::in);**

**ifile >> k[0]>>k[1];**

**ifile.close();**

**cout << "请输入新密码：";**

**cin >> key;**

**fstream ofile("d:\\key.txt", ios::out);**

**ofile << k[0] << " " << key << endl;**

**ofile.close();**

**fstream sfile("d:\\key.txt", ios::in);**

**sfile >> k[0] >> k[1];**

**if (k[1]==key)**

**cout << "修改密码成功！" << endl;**

**else**

**cout << "修改密码失败！" << endl;**

**ifile.close();**

**ofile.close();**

**sfile.close();**

**}**

**Person::Person(string name, string sex, int age) :\_name(name), \_sex(sex), \_age(age) {}**

**Doctor::Doctor(string Dname, string Dsex, int Dage, string Specialty, double vist\_fee):Person(Dname,Dsex,Dage), specialty(Specialty), off\_vist\_fee(vist\_fee) {}**

**void Doctor::D\_add()**

**{**

**char flag = '0';**

**int row = 0;**

**string D[100][10];**

**memset(D, 0, sizeof(D));//清空数组**

**fstream dfile("d:\\doctor.txt", ios::app);//在文件尾实现追加数据**

**fstream dsfile("d:\\doctor.txt", ios::in);//打开医生记录**

**if (!dfile)**

**cout << "不能打开doctor.txt文件！" << endl;**

**if (!dsfile)**

**cout << "不能打开doctor.txt文件！" << endl;**

**cout << "请输入医生姓名：";**

**cin >> \_name;**

**while (!dsfile.eof())**

**{**

**dsfile >> D[row][0] >> D[row][1] >> D[row][2] >> D[row][3] >> D[row][4];**

**if (D[row][0] == \_name)**

**break;**

**row++;**

**}**

**if (D[row][0] == \_name)**

**{**

**cout << "有重名医生，是否继续输入？Y/N" << endl;**

**cin >> flag;**

**if (flag == 'y' || flag == 'Y')**

**{**

**cout << "请输入医生性别：";**

**cin >> \_sex;**

**cout << "请输入医生年龄：";**

**cin >> \_age;**

**cout << "请输入医生专业：";**

**cin >> specialty;**

**cout << "请输入医生诊费：";**

**cin >> off\_vist\_fee;**

**dfile << \_name << " " << \_sex << " " << \_age << " " << specialty << " " << off\_vist\_fee << endl;**

**}**

**}**

**else**

**{**

**cout << "请输入医生性别：";**

**cin >> \_sex;**

**cout << "请输入医生年龄：";**

**cin >> \_age;**

**cout << "请输入医生专业：";**

**cin >> specialty;**

**cout << "请输入医生诊费：";**

**cin >> off\_vist\_fee;**

**dfile << \_name << " " << \_sex << " " << \_age << " " << specialty << " " << off\_vist\_fee << endl;**

**}**

**cout << "信息已录完，是否继续增加医生信息？Y/N" << endl;**

**cin >> flag;**

**if (flag == 'y' || flag == 'Y')**

**D\_add();**

**else**

**dfile.close();**

**}**

**void Doctor::D\_select()**

**{**

**char flag1;**

**fstream sfile("d:\\doctor.txt", ios::in);**

**if (!sfile)**

**cout << "不能打开doctor.txt文件！" << endl;**

**int row = 0;**

**string D[100][10];//定义一个string类型的二维数组, 用来保存从文本内读取的信息**

**memset(D, 0, sizeof(D));//清空数组**

**while (!sfile.eof())**

**{**

**sfile >> D[row][0] >> D[row][1] >> D[row][2] >> D[row][3] >> D[row][4];**

**row++;**

**}**

**string Dname;**

**int nameflag = 0;**

**cout << "请输入要查找的医生姓名：";**

**cin >> Dname;**

**for (int i = 0;i<row;i++)**

**{**

**if (D[i][0] == Dname)**

**{**

**cout << "医生姓名：" << D[i][0] << endl;**

**cout << "医生性别：" << D[i][1] << endl;**

**cout << "医生年龄：" << D[i][2] << endl;**

**cout << "医生专业：" << D[i][3] << endl;**

**cout << "医生诊费：" << D[i][4] << endl << endl;**

**nameflag = 1;**

**}**

**}**

**if (nameflag == 0)**

**cout << "记录中不存在该医生！" << endl;**

**cout << "是否继续查询?Y/N" << endl;**

**cin >> flag1;**

**if (flag1 == 'y' || flag1 == 'Y')**

**D\_select();**

**else**

**sfile.close();**

**}**

**void Doctor::D\_all()**

**{**

**fstream sfile("d:\\doctor.txt", ios::in);**

**if (!sfile)**

**cout << "不能打开doctor.txt文件！" << endl;**

**int row = 0;**

**string D[100][10];//定义一个string类型的二维数组, 用来保存从文本内读取的信息**

**memset(D, 0, sizeof(D));//清空数组**

**cout << left << setw(20)<<"医生姓名" << setw(20) << "医生性别" << setw(20) << "医生年龄" << setw(20) << "医生专业" << setw(20) << "医生诊费" << endl;**

**while (!sfile.eof())**

**{**

**sfile >> D[row][0] >> D[row][1] >> D[row][2] >> D[row][3] >> D[row][4];**

**cout << left << setw(20) << D[row][0] << setw(20) << D[row][1] << setw(20) << D[row][2] << setw(20) << D[row][3] << setw(20) << D[row][4] << endl;**

**row++;**

**}**

**//for(int i=0;i<row;i++)**

**// cout << D[i][0] << "\t" << D[i][1] << "\t" << D[i][2] << "\t" << D[row][3] << "\t" << D[row][] << endl;**

**sfile.close();**

**}**

**Patient::Patient(string name, string sex, int age, double Drug\_fee, double vist\_fee) :Person(\_name, \_sex, \_age), drug\_fee(Drug\_fee), off\_vist\_fee(vist\_fee) {}**

**void Patient::P\_add()**

**{**

**char flag = '0';**

**int row = 0;**

**string P[100][10];**

**memset(P, 0, sizeof(P));//清空数组**

**fstream dfile("d:\\patient.txt", ios::app);**

**fstream dsfile("d:\\patient.txt", ios::in);//打开患者记录**

**if (!dfile)**

**cout << "不能打开patient.txt文件！" << endl;**

**if (!dsfile)**

**cout << "不能打开patient.txt文件！" << endl;**

**cout << "请输入患者姓名：";**

**cin >> \_name;**

**while (!dsfile.eof())**

**{**

**dsfile >> P[row][0] >> P[row][1] >> P[row][2] >> P[row][3] >> P[row][4];**

**if (P[row][0] == \_name)**

**break;**

**row++;**

**}**

**if (P[row][0] == \_name)**

**{**

**cout << "有重名患者，是否继续输入？Y/N" << endl;**

**cin >> flag;**

**if (flag == 'y' || flag == 'Y')**

**{**

**cout << "请输入患者性别：";**

**cin >> \_sex;**

**cout << "请输入患者年龄：";**

**cin >> \_age;**

**cout << "请输入患者医药费：";**

**cin >> drug\_fee;**

**cout << "请输入患者诊费（即医生诊费）：";**

**cin >> off\_vist\_fee;**

**dfile << \_name << " " << \_sex << " " << \_age << " " << drug\_fee << " " << off\_vist\_fee << endl;**

**}**

**}**

**else**

**{**

**cout << "请输入患者性别：";**

**cin >> \_sex;**

**cout << "请输入患者年龄：";**

**cin >> \_age;**

**cout << "请输入患者医药费：";**

**cin >> drug\_fee;**

**cout << "请输入患者诊费（即医生诊费）：";**

**cin >> off\_vist\_fee;**

**dfile << \_name << " " << \_sex << " " << \_age << " " << drug\_fee << " " << off\_vist\_fee << endl;**

**}**

**cout << "信息已录完，是否继续增加患者信息？Y/N" << endl;**

**cin >> flag;**

**if (flag == 'y' || flag == 'Y')**

**P\_add();**

**dfile.close();**

**dsfile.close();**

**}**

**void Patient::P\_select()**

**{**

**char flag1;**

**fstream sfile("d:\\patient.txt", ios::in);**

**if (!sfile)**

**cout << "不能打开patient.txt文件！" << endl;**

**int row = 0;**

**string P[100][10];//定义一个string类型的二维数组, 用来保存从文本内读取的信息**

**memset(P, 0, sizeof(P));//清空数组**

**while (!sfile.eof())**

**{**

**sfile >> P[row][0] >> P[row][1] >> P[row][2] >> P[row][3] >> P[row][4];**

**row++;**

**}**

**string Pname;**

**int nameflag = 0;**

**cout << "请输入要查找的患者姓名：";**

**cin >> Pname;**

**for (int i = 0;i<row;i++)**

**{**

**if (P[i][0] == Pname)**

**{**

**cout << "患者姓名：" << P[i][0] << endl;**

**cout << "患者性别：" << P[i][1] << endl;**

**cout << "患者年龄：" << P[i][2] << endl;**

**cout << "患者产生的药费：" << P[i][3] << endl;**

**cout << "患者诊费（即医生诊费）：" << P[i][4] << endl << endl;**

**nameflag = 1;**

**}**

**}**

**if (nameflag == 0)**

**cout << "记录中不存在该患者！" << endl;**

**cout << "是否继续查询?Y/N" << endl;**

**cin >> flag1;**

**if (flag1 == 'y' || flag1 == 'Y')**

**P\_select();**

**else**

**sfile.close();**

**}**

**void Patient::P\_all()**

**{**

**fstream sfile("d:\\patient.txt", ios::in);**

**if (!sfile)**

**cout << "不能打开patient.txt文件！" << endl;**

**int row = 0;**

**string P[100][10];//定义一个string类型的二维数组, 用来保存从文本内读取的信息**

**memset(P, 0, sizeof(P));//清空数组**

**cout << left << setw(20) << "患者姓名" << setw(20) << "患者性别" << setw(20) << "患者年龄" << setw(20) << "患者药费" << setw(20) << "患者诊费" << endl;**

**while (!sfile.eof())**

**{**

**sfile >> P[row][0] >> P[row][1] >> P[row][2] >> P[row][3] >> P[row][4];**

**cout << left << setw(20) << P[row][0] << setw(20) << P[row][1] << setw(20) << P[row][2] << setw(20) << P[row][3] << setw(20)<< P[row][4] << endl;**

**row++;**

**}**

**sfile.close();**

**}**

**Bill::Bill(string name, string sex, int age, double Drug\_fee, double vist\_fee, string Dname, string Dsex, int Dage, string Specialty, double Dvist\_fee):Patient(name, sex, age, Drug\_fee, vist\_fee), D\_name(Dname),D\_sex(Dsex),D\_age(Dage),specialty(Specialty),Doff\_vist\_fee(Dvist\_fee){}**

**void Bill::Money(void)**

**{**

**double sum = 0.0;**

**fstream patientfile("d:\\bill.txt", ios::in);;**

**if (!patientfile)**

**{**

**cout << "打开文件失败！" << endl;**

**}**

**string B[100][10];**

**int row = 0;**

**while (!patientfile.eof())**

**{**

**patientfile >> B[row][0] >> B[row][1] >> B[row][2] >> B[row][3] >> B[row][4] >> B[row][5] >> B[row][6] >> B[row][7] >> B[row][8] >> B[row][9];**

**row++;**

**}**

**for (int i = 0; i<row; i++)**

**{**

**sum = sum + atoi(B[i][3].c\_str()) + atoi(B[i][4].c\_str());**

**}**

**cout << "所有患者的总费用为：" << sum << "元" << endl;**

**}**

**void Bill::B\_add()**

**{**

**char flag = '0';**

**int row = 0;**

**string B[100][10];**

**memset(B, 0, sizeof(B));//清空数组**

**fstream bfile("d:\\bill.txt", ios::app);**

**fstream bsfile("d:\\bill.txt", ios::in);//打开账单记录**

**if (!bfile)**

**cout << "不能打开bill.txt文件！" << endl;**

**if (!bsfile)**

**cout << "不能打开patient.txt文件！" << endl;**

**cout << "请输入患者姓名：";**

**cin >> \_name;**

**while (!bsfile.eof())**

**{**

**bsfile >> B[row][0] >> B[row][1] >> B[row][2] >> B[row][3] >> B[row][4] >> B[row][5] >> B[row][6] >> B[row][7] >> B[row][8] >> B[row][9];**

**if (B[row][0] == \_name)**

**break;**

**row++;**

**}**

**if (B[row][0] == \_name)**

**{**

**cout << "账单中有重名患者，是否继续输入？Y/N" << endl;**

**cin >> flag;**

**if (flag == 'y' || flag == 'Y')**

**{**

**cout << "请输入患者性别：";**

**cin >> \_sex;**

**cout << "请输入患者年龄：";**

**cin >> \_age;**

**cout << "请输入患者医药费：";**

**cin >> drug\_fee;**

**cout << "请输入患者诊费（即医生诊费）：";**

**cin >> off\_vist\_fee;**

**cout << "请输入患者的主治医生名字：";**

**cin >> D\_name;**

**cout << "请输入患者的主治医生性别：";**

**cin >> D\_sex;**

**cout << "请输入患者的主治医生年龄：";**

**cin >> D\_age;**

**cout << "请输入患者的主治医生专业：";**

**cin >> specialty;**

**cout << "请输入患者的主治医生诊费：";**

**cin >> Doff\_vist\_fee;**

**bfile << \_name << " " << \_sex << " " << \_age << " " << drug\_fee << " " << off\_vist\_fee << " " << D\_name << " " << D\_sex << " " << D\_age << " " << specialty << " " << Doff\_vist\_fee << endl;**

**}**

**}**

**else**

**{**

**cout << "请输入患者性别：";**

**cin >> \_sex;**

**cout << "请输入患者年龄：";**

**cin >> \_age;**

**cout << "请输入患者医药费：";**

**cin >> drug\_fee;**

**cout << "请输入患者诊费：";**

**cin >> off\_vist\_fee;**

**cout << "请输入患者的主治医生名字：";**

**cin >> D\_name;**

**cout << "请输入患者的主治医生性别：";**

**cin >> D\_sex;**

**cout << "请输入患者的主治医生年龄：";**

**cin >> D\_age;**

**cout << "请输入患者的主治医生专业：";**

**cin >> specialty;**

**cout << "请输入患者的主治医生诊费：";**

**cin >> Doff\_vist\_fee;**

**bfile << \_name << " " << \_sex << " " << \_age << " " << drug\_fee << " " << off\_vist\_fee << " " << D\_name << " " << D\_sex << " " << D\_age << " " << specialty << " " << Doff\_vist\_fee << endl;**

**}**

**bfile.close();**

**bsfile.close();**

**//写入patient.txt中**

**int row1 = 0;**

**string P[100][10];**

**memset(P, 0, sizeof(P));//清空数组**

**fstream dfile("d:\\patient.txt", ios::app);**

**fstream dsfile("d:\\patient.txt", ios::in);//打开患者记录**

**if (!dfile)**

**cout << "不能打开patient.txt文件！" << endl;**

**if (!dsfile)**

**cout << "不能打开patient.txt文件！" << endl;**

**while (!dsfile.eof())**

**{**

**dsfile >> P[row1][0] >> P[row1][1] >> P[row1][2] >> P[row1][3] >> P[row1][4];**

**if (P[row1][0] == \_name)**

**break;**

**row++;**

**}**

**if (P[row1][0] == \_name && P[row1][1] == \_sex && atoi(P[row1][2].c\_str()) == \_age)**

**{}**

**else**

**{**

**dfile << \_name << " " << \_sex << " " << \_age << " " << drug\_fee << " " << off\_vist\_fee << endl;**

**}**

**dfile.close();**

**dsfile.close();**

**cout << "是否继续增加患者账单信息？Y/N" << endl;**

**cin >> flag;**

**if (flag == 'y' || flag == 'Y')**

**B\_add();**

**bfile.close();**

**bsfile.close();**

**}**

**void Bill::B\_select()**

**{**

**char flag2;**

**fstream sfile("d:\\bill.txt", ios::in);**

**if (!sfile)**

**cout << "不能打开bill.txt文件！" << endl;**

**int row = 0;**

**string B[100][10];//定义一个string类型的二维数组, 用来保存从文本内读取的信息**

**memset(B, 0, sizeof(B));//清空数组**

**while (!sfile.eof())**

**{**

**sfile >> B[row][0] >> B[row][1] >> B[row][2] >> B[row][3] >> B[row][4] >> B[row][5] >> B[row][6] >> B[row][7] >> B[row][8] >> B[row][9];**

**row++;**

**}**

**string Bname;**

**int nameflag = 0;**

**cout << "请输入要查找的患者姓名：";**

**cin >> Bname;**

**for (int i = 0;i<row;i++)**

**{**

**if (B[i][0] == Bname)**

**{**

**cout << "患者姓名：" << B[i][0] << endl;**

**cout << "患者性别：" << B[i][1] << endl;**

**cout << "患者年龄：" << B[i][2] << endl;**

**cout << "患者产生的药费：" << B[i][3] << endl;**

**cout << "患者产生的诊费：" << B[i][4] << endl;**

**cout << "主治医生姓名：" << B[i][5] << endl;**

**cout << "主治医生性别：" << B[i][6] << endl;**

**cout << "主治医生年龄：" << B[i][7] << endl;**

**cout << "主治医生专业：" << B[i][8] << endl;**

**cout << "主治医生诊费：" << B[i][9] << endl << endl;**

**nameflag = 1;**

**}**

**}**

**if (nameflag == 0)**

**cout << "账单中不存在该患者！" << endl;**

**cout << "是否继续查询?Y/N" << endl;**

**cin >> flag2;**

**if (flag2 == 'y' || flag2 == 'Y')**

**B\_select();**

**sfile.close();**

**}**

**void Bill::B\_all()**

**{**

**fstream sfile("d:\\bill.txt", ios::in);**

**if (!sfile)**

**cout << "不能打开bill.txt文件！" << endl;**

**int row = 0;**

**string B[100][10];//定义一个string类型的二维数组, 用来保存从文本内读取的信息**

**memset(B, 0, sizeof(B));//清空数组**

**cout << left << setw(20) << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << setw(4) << "患者" << setw(26) << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << setw(18) << "------------------" << setw(8) << "主治医生" << setw(20) << "--------------------" << endl;**

**cout << left << setw(10) << "姓名" << setw(10) << "性别" << setw(10) << "年龄" << setw(10) << "药费(元)" << setw(10) << "诊费(元)"<<"|"<< setw(10) << "姓名" << setw(10) << "性别" << setw(10) << "年龄" << setw(10) << "专业" << setw(10) << "诊费(元)" << endl;**

**while (!sfile.eof())**

**{**

**sfile >> B[row][0] >> B[row][1] >> B[row][2] >> B[row][3] >> B[row][4] >> B[row][5] >> B[row][6] >> B[row][7] >> B[row][8] >> B[row][9];**

**cout << left << setw(10) << B[row][0] << setw(10) << B[row][1] << setw(10) << B[row][2] << setw(10) << B[row][3] << setw(10) << B[row][4] << "|" << setw(10) << B[row][5] << setw(10) << B[row][6] << setw(10) << B[row][7] << setw(10) << B[row][8] << setw(10) << B[row][9] << endl;**

**row++;**

**}**

**sfile.close();**

**}**

**3.主函数：**

**#include <iostream>**

**#include<string>**

**#include<windows.h>**

**#include"class.h"**

**using namespace std;**

**int main()**

**{**

**int n; //用来选择的变量**

**int flag;**

**Doctor a;**

**Patient b;**

**Bill c;**

**while (1)**

**{//while 1**

**flag = 0;**

**mainmenu();**

**cout << "请输入你的操作：";**

**cin >> n;**

**switch (n)**

**{//switch 1**

**case 1: {**

**while (1)**

**{//while 2**

**usermenu();**

**cout << "请输入你的操作：";**

**cin >> n;**

**switch (n)**

**{//switch 2**

**case 1: {**

**a.D\_all();**

**system("pause");**

**}break;**

**case 2: {**

**a.D\_select();**

**}break;**

**case 3: {}break;**

**default:**

**cout << "输入错误！请重新输入！" << endl;**

**Sleep(500);**

**break;**

**}//end switch 2**

**if (n == 3)**

**break;**

**}break;//end while 2**

**}break;**

**case 2: {**

**land(flag);**

**if (flag == 1)**

**{**

**while (1)**

**{//whlie 3**

**menu();**

**cout << "请输入你的操作：";**

**cin >> n;**

**switch (n)**

**{//switch 3**

**case 1: {**

**while (1)**

**{//while 4**

**Dmenu();**

**cout << "请输入你的操作：";**

**cin >> n;**

**switch (n)**

**{//switch 4**

**case 1: {**

**a.D\_all();**

**system("pause");**

**}break;**

**case 2: {**

**a.D\_select();**

**}break;**

**case 3: {**

**a.D\_add();**

**}break;**

**case 4: { }break;**

**default:**

**cout << "输入错误！请重新输入！" << endl;**

**Sleep(500);//执行挂起一段时间//500ms=0.5s，也就是等待一段时间在继续执行;在头文件windows.h中**

**break;**

**}//end switch 4**

**if (n == 4)**

**break;**

**}//end while 4**

**}break;**

**case 2: {**

**while (1)**

**{//while 5**

**Pmenu();**

**cout << "请输入你的操作：";**

**cin >> n;**

**switch (n)**

**{//switch 5**

**case 1: {**

**b.P\_all();**

**system("pause");**

**}break;**

**case 2: {**

**b.P\_select();**

**}break;**

**case 3: {**

**b.P\_add();**

**}break;**

**case 4: { }break;**

**default:**

**cout << "输入错误！请重新输入<< endl;**

**Sleep(500);**

**break;**

**}//end switch 5**

**if (n == 4)**

**break;**

**}//end while 5**

**}break;**

**case 3: {**

**while (1)**

**{//while 6**

**Bmenu();**

**cout << "请输入你的操作：";**

**cin >> n;**

**switch (n)**

**{//switch 6**

**case 1: {**

**c.B\_all();**

**system("pause");**

**}break;**

**case 2: {**

**c.B\_select();**

**}break;**

**case 3: {**

**c.B\_add();**

**}break;**

**case 4: {**

**c.Money();**

**system("pause");**

**}break;**

**case 0: { }break;**

**default:**

**cout << "输入错误！请重新输入！" << endl;**

**Sleep(500);//执行挂起一段时间500ms=0.5s，也就是等待//一段时间在继续执行;在头文件windows.h中**

**break;**

**}//end switch 6**

**if (n == 0)**

**break;**

**}//end while 6**

**}break;**

**case 4: {**

**changekey();**

**system("pause");**

**}break;**

**case 5: { }break;**

**default:**

**cout << "输入错误！请重新输入！" << endl;**

**Sleep(500);**

**break;**

**}**

**if (n == 5)**

**break;**

**}**

**}//end while 3**

**}break;**

**case 0: {}break;**

**default:**

**cout << "输入错误！请重新输入！" << endl;**

**Sleep(500);**

**break;**

**}//end switch 1**

**if (n == 0)**

**break;**

**}//end while 1**

**return 0;**

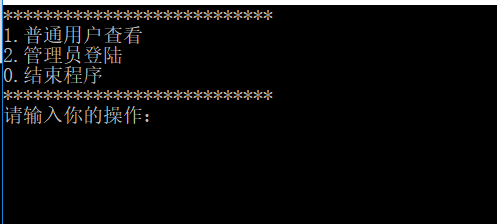
**}**

## **3.7.2 模块测试**

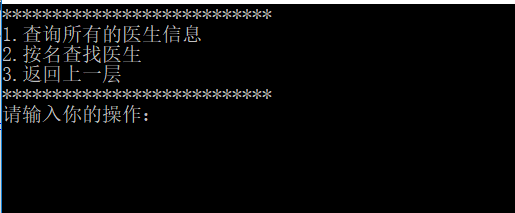
**测试步骤：**

**1．界面测试**

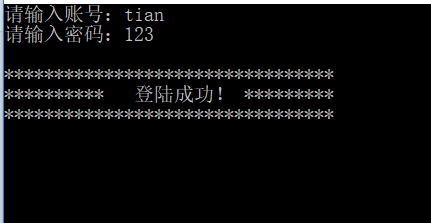
**（1）系统登陆界面如下图**



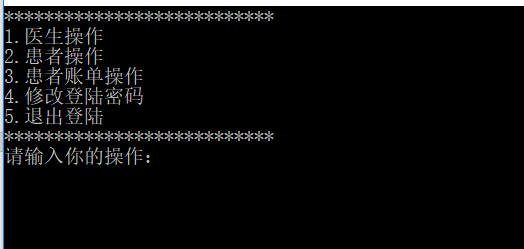
1. **普通用户查看界面如下图：**



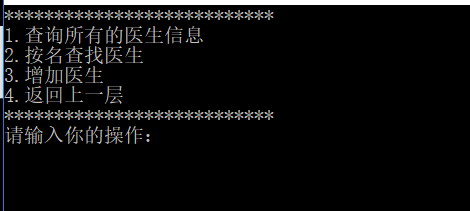
1. **管理员登陆界面如下图：**



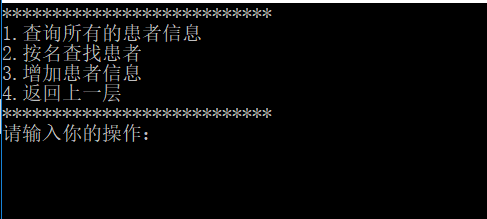
**（2）管理员操作界面如下图**



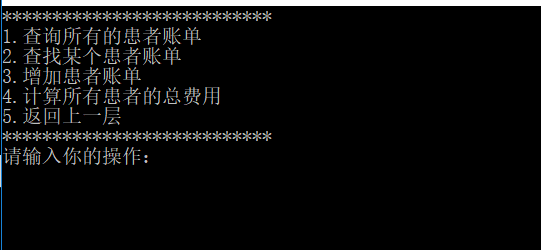
**1）医生操作界面如下图：**



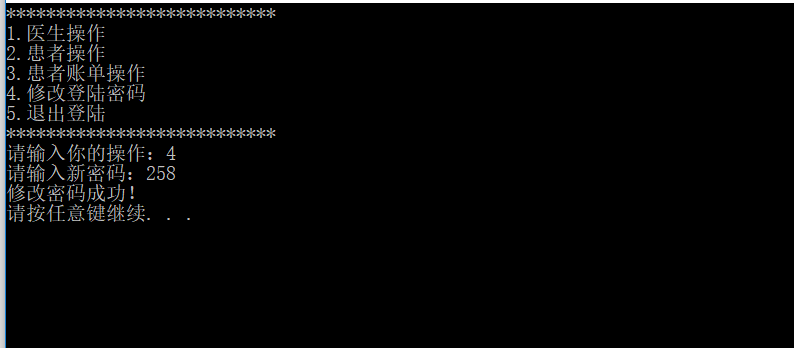
**2）患者操作界面如下图：**



**3）患者账单操作界面如下图：**



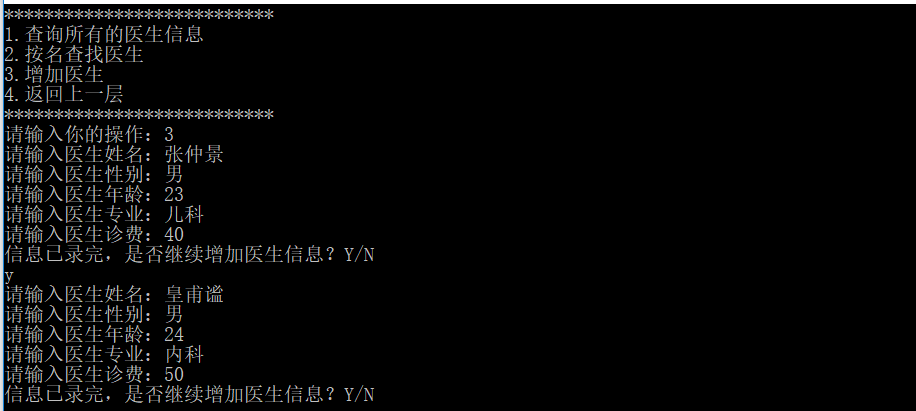
**4）修改登陆密码界面如下图：**



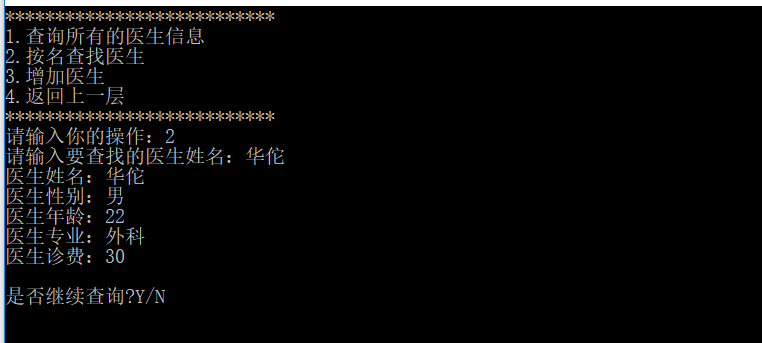
**2.功能操作测试**

**医生：**

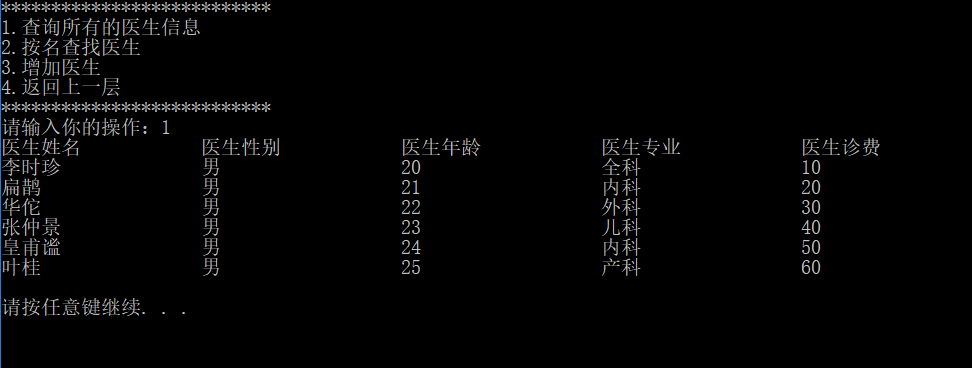
**（1）医生信息增加如下图：**

****

**（2）查找某一医生信息如下图：**

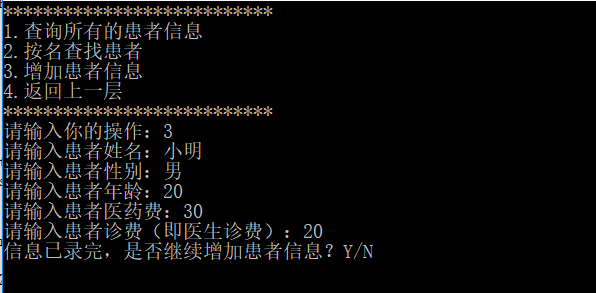


**（3）查询所有医生信息如下图：**

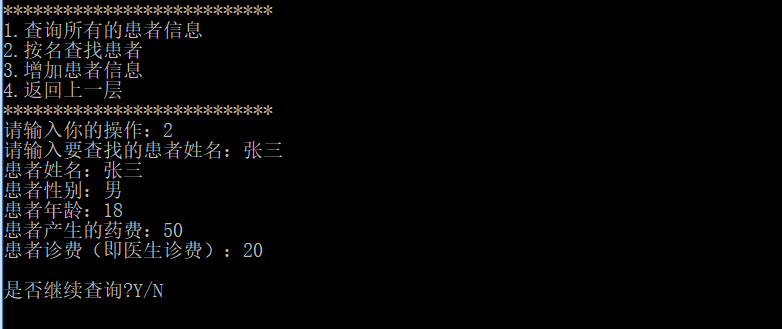


**患者：**

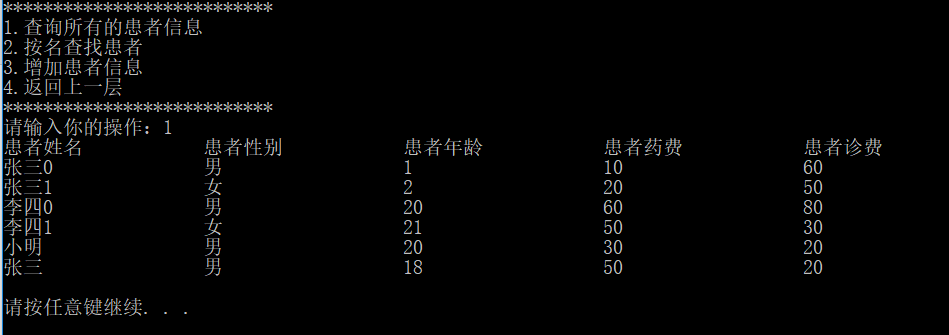
1. **增加患者信息如下图：**



1. **查找某一患者信息如下图：**

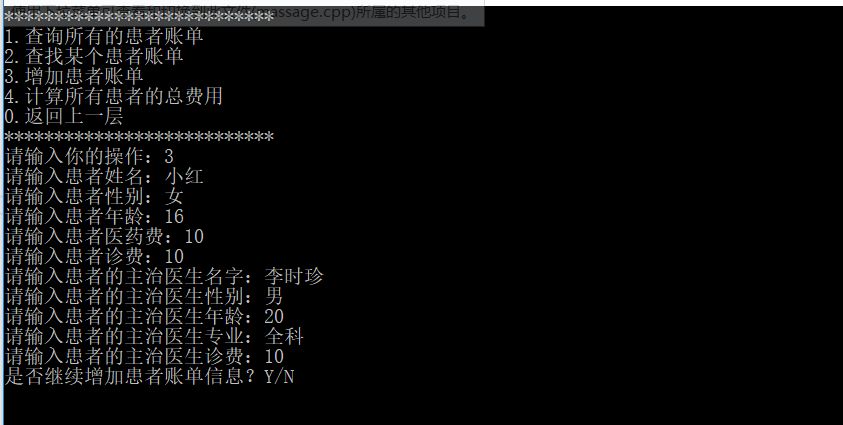


1. **查询所有患者信息如下图：**

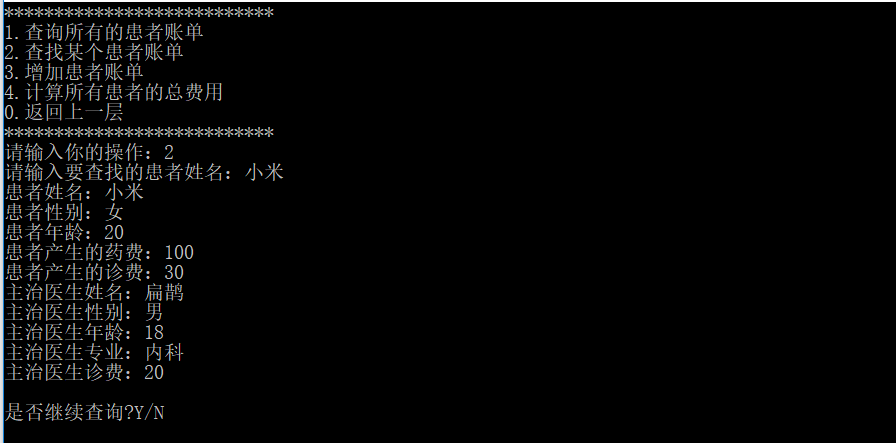


**账单：**

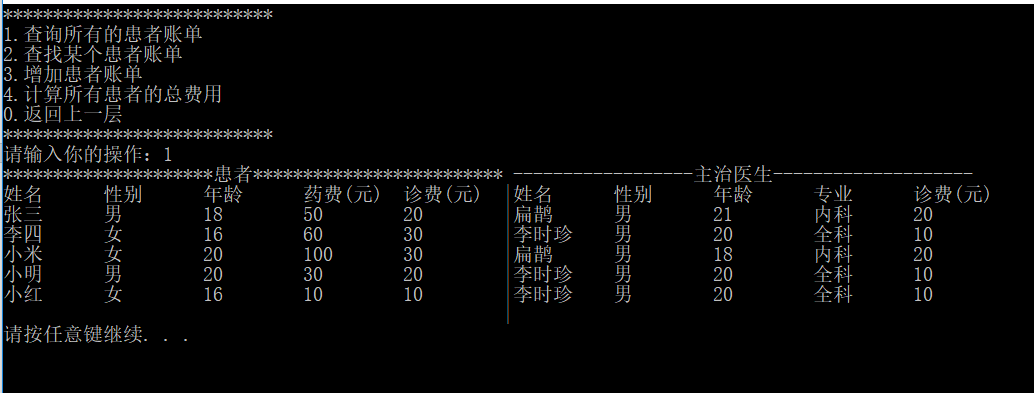
* 1. **增加患者账单如下图：**



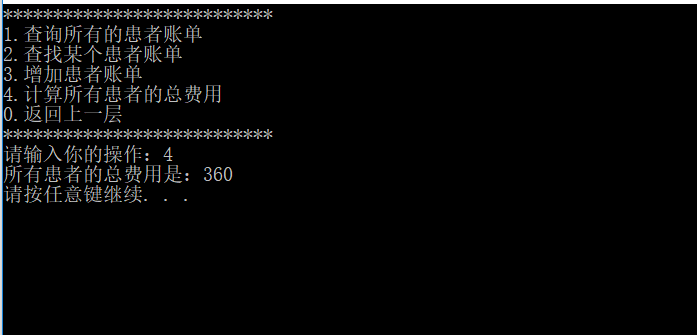
* 1. **查找某一患者账单信息如下图：**



**③查看所有账单信息如下图：**



**④统计所有患者的总费用如下图：**



# ****四、程序设计过程遇到的问题****

**1. 对文件的读写操作不熟悉，刚开始对同一文件同时进行读写操作，发现无论怎么改里面的代码都达不到要求。**

**解决方法：再次阅读了c++程序设计第九章的关于文件读写的内容，发现不能同时对同一文件进行读和写，然后我就先读文件，关闭文件再进行写操作。**

**2.在进入账单操作界面时，当选择返回上一层时，本来应该是返回到管理员操作界面的，结果直接退出登陆了。**

**分析并解决问题：通过反复检查代码，发现当给变量“n”输入一个值“5”选择返回时，程序退出switch（）选择语句，接着退出while循环，然后就和上一层的退出登陆的按键“5”冲突了，结果便直接退出登陆，只要选择另一个值，例如“0”来作为返回上一层的输入值即可。**

**3.主程序一开始想到的是用if…else语句的嵌套来实现功能的选择，但发现可读性不高，代码复杂。**

**解决方法：通过分析思考，最终决定用switch语句来实现功能的选择。**

**4.调试程序代码的时候，刚开始发现操作界面每次输入的命令和结果都会保留在界面上，还有显示的数据格式很不美观。**

**解决方法：通过上网查资料和阅读《c++程序设计》，找到了一系统清屏函数system（”cls”）和格式函数setw（）。**

**5.写代码的过程中尤其要注意变量类型，有时会不小心写错了，就会出现一些不易察觉的bug。。**

## **五、感想与体会**

**经过了为期两周的c++课程设计，不敢说已经完全掌握了c++这门语言，但至少学会了运用这门语言去实现一些简单的应用程序设计。**

**从刚抽到课题，理清思路，查阅资料，学习StarUML作图，到代码的开发，最后再调试并完善代码。我觉得这一过程中最重要的不是实验的结果，而是从无到有的一种个人的自主学习、自主思考发现并解决问题能力的培养。通过学习，发现问题并解决问题的过程中，我对c++有了更加深刻的理解，也在一定程度上扩展了我的知识面和提高了我的开发能力和动手能力。同时，也让我深刻地了解到我在发现问题这一方面的能力还有待提高，因为在写此次课设代码过程中发现的问题往往要仔细品读许多遍才能找到解决方法，这也从侧面反映了我对知识掌握地并不是很牢固。**

**我知道，无论是c++还是其他方面的知识，还有很多是我不懂的，这就需要我不断地去学习、去掌握应用。**

# ****附参考文献****

1. **吴乃陵,c++程序设计，高等教育出版社，第二版，2006年**
2. [**http://www.docin.com/p-1943503976.html**](http://www.docin.com/p-1943503976.html)
3. [**https://jingyan.baidu.com/article/e8cdb32b5cdf9137052bad93.html**](https://jingyan.baidu.com/article/e8cdb32b5cdf9137052bad93.html)
4. [**https://www.cnblogs.com/biehongli/p/6052421.html**](https://www.cnblogs.com/biehongli/p/6052421.html)