

【附录一】源代码

# hospital.cpp

#include"hospital.h"

#define Max 1000000 //最多支持的挂号的人数

long renshu[5]= {0,0,0,0,0}; //存放各科室人数

long I=0; //数组下标

long L=0;

long Ghsr=0; //用于存放挂号的收入！

using namespace std;

struct Patient //定义患者结构体

{

char num[6]; //患者编号

char time[8] = ""; //挂号日期,初始化0

int sex = 0; //性别 1为男,2为女

int age = 0; //年龄

char room = '0'; //所挂号的科室

char jz[4] = "否"; //是否是急诊，默认值否

char zp; //专家门诊还是普通门诊

} P[Max],Q[Max]; //最多存放1000000个

struct Age //年龄段结构体

{

int e = 0; //儿童 enfant 1-14

int y = 0; //青年 young people 15-25

int m = 0; //中年 wrinkly 26-55

int o = 0; //老年 old folks 55以上

};

struct ghfl

{

int drrs = 0; //当日号人数

int yyrs = 0; //预约人数

int jzrs = 0; //急诊人数

} G;

struct Paixu //定义各科室就诊人数排序结构体

{

char name[6]; //科室名称

int money; //本科收入

struct Age a ; //Age 结构体类型的成元

} J[5]; //一共5个科室

void clear()

{

system("cls"); //清屏

system("date/t"); //显示日期

}

//--------------分科室函数------------------

void hospital::room(char room)

{

clear();

switch(room)

{

case '1':

renshu[0]++;

cout<<'\t'<<"\n\t请到第5就诊室！\n\n\t您的前面还有 " << renshu[0]-1 << " 人\n\n\t";

break;

case '2':

renshu[1]++;

cout<<'\t'<<"\n\t请到第4就诊室！\n\n\t您的前面还有 " << renshu[1]-1<< " 人\n\n\t";

break;

case '3':

renshu[2]++;

cout<<'\t'<<"\n\t请到第3就诊室！\n\n\t您的前面还有 " << renshu[2]-1<< " 人\n\n\t";

break;

case '4':

renshu[3]++;

cout<<'\t'<<"\n\t请到第2就诊室！\n\n\t您的前面还有 " << renshu[3]-1<< " 人\n\n\t";

break;

case '5':

renshu[4]++;

cout<<'\t'<<"\n\t请到第1就诊室！\n\n\t您的前面还有 " << renshu[4]-1<< " 人\n\n\t";

break;

}

}

//--------------挂当日号函数------------------

int hospital::gdrh()

{

char temp[1000] = ""; //为了避免输入多个字符时会产生的错误

if(input() == -1)

{

return 1; //如果在input里面选择了返回则返回

}

//tj(); //调用统计计算前面有几人

room(P[I-1].room); //分科室

G.drrs++; //挂当日号人数+1

system("pause"); //暂停

return 1;

}

//--------------查看挂号人数函数------------------

int hospital::ghrs()

{

char temp[100]="";

clear();

cout<<"\n\n\n\t\t\t各类挂号情况一览表\n";

cout << "\t==================================================\n";

cout << "\n\n\t" << "\t挂当日号的人数为----" << G.drrs;

cout << "\n\n\t" << "\t挂预约号的人数为----" << G.yyrs;

cout << "\n\n\t" << "\t挂急诊号的人数为----" << G.jzrs;

cout << "\n\n\n\t==================================================\n\t";

system("pause");

return(1);

}

//--------------挂急诊号函数------------------

int hospital::gjzh()

{

if(-1 == input())

{

return 1; //返回

}

clear(); //清屏

switch(P[I-1].room)

{

case '1':

renshu[0]++;

cout<<'\t'<<"\n\t请到第1急诊室！\n\n\t";

break;

case '2':

renshu[1]++;

cout<<'\t'<<"\n\t请到第2急诊室！\n\n\t";

break;

case '3':

renshu[2]++;

cout<<'\t'<<"\n\t请到第3急诊室！\n\n\t";

break;

case '4':

renshu[3]++;

cout<<'\t'<<"\n\t请到第4急诊室！\n\n\t";

break;

case '5':

renshu[4]++;

cout<<'\t'<<"\n\t请到第5急诊室！\n\n\t";

break;

}

G.jzrs++; //急诊号人数+1

strcpy( P[I-1].jz,"是");

system("pause");

return 1;

}

//--------------查看挂号收入函数------------------

int hospital::ghsr()

{

int i=0;

char temp[100]="";

char temp1='/';

clear();

cout<<"\n\n\n\t\t\t挂号情况一览表\n";

cout << "\t==================================================\n";

cout<<"\t今天的挂号费收入为: "<<Ghsr<<" 元";

cout << "\n\t==================================================\n";

for(i=0; i<5; i++)

cout<<"\n\n\t"<<J[i].name<<"的收入为: "<<J[i].money ;

cout << "\n\t==================================================\n\t";

system("pause");

return(1);

}

//--------------查看年龄比函数------------------

int hospital::nlb()

{

long sum[5];

int i;

char temp[100]="";

clear();

cout<<"\n\n\t\t\t 各科室患者年龄层人数\n\n";

cout<<"\t\t儿 童\t 青 年\t 中 年\t 老 年\t 总 和\n\n";

cout<<"\t\t(1-14)\t(15-25)\t(26-55)\t(56以上)(1-100)\n";

cout<<" =============================================================\n\n";

for(i=0; i<5; i++) //求出各科室就诊的总人数

sum[i]=J[i].a.e+J[i].a.y+J[i].a.m+J[i].a.o ;

for(i=0; i<5; i++)

{

cout <<'\t'<<J[i].name<<'\t'<<J[i].a.e<<"\t "<<J[i].a.y<<"\t "<<J[i].a.m<<"\t "<<J[i].a.o<<"\t "<<sum[i];

cout<<"\n\n";

}

cout<<" =============================================================\n\t";

system("pause");

return(1);

}

//--------------查看各科就诊比例------------------

int hospital::jzrs()

{

int i=0;

char temp[100]="";

clear();

cout<<"\n\n\n\t\t\t各科就诊人数一览表\n";

cout << "\t==================================================";

for(i=0; i<5; i++)

cout<<"\n\n\t"<<i+1<<". "<<J[i].name<<"的就诊人数为: "<<renshu[i] ;

cout << "\n\t==================================================\n\t";

system("pause");

return(1);

}

//--------------挂号向导函数------------------

int hospital::ghxd()

{

clear(); //清屏

char gn, temp[1000]=""; //为了避免输入多个字符时会产生的错误

cout << "\n\n\t\t\t====挂号向导====\n\n";

cout << "\t\t以下仅供参考,如疑问请咨询医护人员！\n\n";

cout << "\t===科室============================包含内容===================\n\n" ;

cout << "\t 内科\t\t呼吸,消化,泌尿,心脏,免疫,血液,内分泌,神经,肾,等\n\n";

cout << "\t 外科\t\t感染,心胸,肝胆,外伤,微创,普外,肛肠,整形,等\n\n";

cout << "\t 骨科\t\t骨折,骨质增生,风湿,关节,四肢,骨神经,等\n\n";

cout << "\t 口腔科\t牙齿,牙龈,颌面,舌,淋巴,口腔溃疡,等\n\n";

cout << "\t 皮肤科\t癣类,疱疹,皮炎,皮肤瘙痒,癍类,痘类,等\n\n";

cout << "\t==============================================================\n\t";

system("pause");

return 1;

}

//--------------预约挂号功能函数------------------  
int hospital::yygh()  
{  
    char str\_1[5],str\_2[5],str[50],gn;                    //保存系统时间与预约不同天数及系统时间  
    int m=1,n=0;  
    if(-1==input())  
        return 1;  
    struct tm \*mytime;  
    time\_t LocalTime;  
    LocalTime=time(NULL);  
    mytime=localtime(&LocalTime);  
    strftime(str,sizeof(str),"%c",mytime);           //接收系统当前时间，格式09/04/15  
    do  
    {  
        clear();  
        if(m-n==0)  
        {  
            cout << "\n\n\n请挂当天号......\n\n";  
            system("pause");  
            return 1;  
        }  
  
        else  
        {  
            char date[8];  
            int flag=1;  
            while(flag==1)  
            {  
                   clear();  
                   memset(str\_1,'\0',sizeof(str\_1));  
                   memset(str\_2,'\0',sizeof(str\_2));  
  
                   cout << "==================================================\n";  
                   cout<< "\n\t注意：预约挂号只能预约七天以内\n\n"<<endl;  
  
                   cout<< "\t请输入日期(格式：月(00)/日(00)/年(00)：";  
                   cin>>date;  
                   strcpy(P[I].time, date);  
                   for(int j=0,r=0; j<strlen(date); j++)                   //保存不同天数  
                   {  
                        if(str[j]!=date[j])  
                        {  
                            if((j==3||j==4)&&(date[j]-'0')>=0&&(date[j]-'0')<=9)  
                            {  
                                str\_1[r]=str[j];  
                                str\_2[r]=date[j];  
                                r++;  
                                flag=0;  
                            }  
                           else  
                           {  
                                flag=1;  
                                clear();  
                                cout<<"\n\n\n请输入正确的日期格式......\n\n";  
                                system("pause");  
                                break;  
  
                           }  
                        }  
                   }  
               }  
           sscanf(str\_2,"%d",&m);                                   //字符串转换为数字  
           sscanf(str\_1,"%d",&n);  
        }  
    }  
          while(m-n>7||m-n<=0);  
         cout<< "\n\t请"<<m-n<<"天之后前来就诊，谢谢！\n\n\t";            //不同天数相减  
         G.yyrs++;  //预约挂号人数+1  
         switch(P[I-1].room)  
         {  
             case '1':  
                   renshu[0]++;  
                    break;  
             case '2':  
                   renshu[1]++;  
                   break;  
             case '3':  
                   renshu[2]++;  
                   break;  
             case '4':  
                    renshu[3]++;  
                   break;  
             case '5':  
                   renshu[4]++;  
                   break;  
          }  
        Q[L].zp=P[I-1].zp;  
        Q[L].age=P[I-1].age;  
        strcpy(Q[L].jz,P[I-1].jz);  
        strcpy(Q[L].num,P[I-1].num);  
        Q[L].room=P[I-1].room;  
        Q[L].sex=P[I-1].sex;  
        strcpy(Q[L].time, P[I-1].time);  
        L++;  
        system("pause");  
        return 1;  
}

//--------------统计功能函数------------------

void hospital::tj()

{

long i=0,m;

struct Paixu t;

m=I;

strcpy(J[0].name,"内科"); //J结构体数组赋初值

strcpy(J[1].name,"外科");

strcpy(J[2].name,"骨科");

strcpy(J[3].name,"皮肤科");

strcpy(J[4].name,"口腔科");

for(i=0; i<5; i++) //各科室各年龄段人数初值为0

{

J[i].a.e=0;

J[i].money = 0;

J[i].a.y=0;

J[i].a.m=0;

J[i].a.o=0;

}

Ghsr = 0;

for(i=0; i<m; i++)

{

if(P[i].zp=='Y')Ghsr+=10,J[P[i].room-'1'].money+=10; //统计挂号收入:专家

else if(P[i].zp == 'N') Ghsr+=5,J[P[i].room-'1'].money+=5; //统计挂号收入:普通

if (P[i].age<=14) //统计各科室各年龄段人数

J[P[i].room-'1'].a.e++;

else if (P[i].age <=25)

J[P[i].room-'1'].a.y++;

else if(P[i].age <=55)

J[P[i].room-'1'].a.m++;

else J[P[i].room-'1'].a.o++;

}

}

//--------------输入功能函数------------------

int hospital::input()

{

char room2='/',cls1='0',zp2,sex2,qr;

int age2=0,i=0;

char room1[1000]="";

char zp1[1000]="";

char sex1[1000]="";

char age1[1000]="";

char cls[1000]="";

char temp[6]="";

char temp1[6]="";

while(I<Max) //重复输入记录

{

strcpy(room1,"");

strcpy(zp1,"");

strcpy(sex1,"");

strcpy(age1,"");

strcpy(cls,"");

strcpy(temp,"");

strcpy(temp1,"");

room2='/';

sex2='/';

zp2='/';

cls1='0';

age2=0;

qr='/';

while((room2<'1' || room2 > '5') && room2!='r' )

{

clear();

cout<<"\n\n\t\t\t\t请选择科室\n\n";

cout << "\n\t=============================================================\n";

cout<<"\n\t\t1.内科";

cout<<"\t\t2.外科";

cout<<"\t\t3.骨科\n\n";

cout<<"\t\t4.皮肤科";

cout<<"\t5.口腔科";

cout<<"\tr.返回!";

cout << "\n\n\t=============================================================\n";

cout<< "\n\t请选择所要挂科室的编号:";

cin>>room1; //输入放在临时变量temp中

if (strlen(room1)==1) room2=tolower(room1[0]);

}

if (room2=='r') return(-1);

while(zp2!='Y'&&zp2!='N') //选择专家门诊吗

{

clear();

cout<<"\n\n\n普通门诊: 5元 \n专家门诊: 10元\n\n\n您是否要挂专家门诊(Y/N):";

cin>>zp1;

if (strlen(zp1)==1)

zp2=toupper(zp1[0]); //变成大写,以避免系统区分大小写

}

while(sex2!='1'&&sex2!='2') //选择性别1为男,2为女

{

clear();

cout<<"\n\n\n\t性 别 (1.男 2.女):";

cin>>sex1;

if (strlen(sex1)==1)

sex2=sex1[0];

}

while(age2<=0||age2>150) //输入的年龄在1--150之间

{

clear();

cout<<"\n\n\n\t年 龄:";

cin>>age1;

if (strlen(age1)<=3)

age2=atoi(age1); //将字符型的age1转换成整型赋给age2;

}

ltoa(I+1,temp,10); //计算当前患者编号

if(strlen(temp)<6)

for(i=0; i<6-strlen(temp); i++)

strcat(temp1,"0");

strcat(temp1,temp);

// 输出患者信息以确认输入是否正确

while(cls1!='1'&&cls1!='2'&&cls1!='r')

{

clear();

cout<<"\n\n\n\t\t\t您的输入的信息如下,请确认是否正确!\n\n\n\n";

cout<<"\t编 号\t 性 别\t年 龄\t所挂科室\t专家门诊\t挂号费\n\n";

cout<<'\t'<<temp1; //编号

if (sex2=='1')

cout<<'\t'<<" 男"; //性别

else if(sex2 =='2')

cout<<'\t'<<" 女";

cout<<'\t'<<age2; //年龄

switch(room2) //科室

{

case '1':

cout<<'\t'<<"内 科";

break;

case '2':

cout<<'\t'<<"外 科";

break;

case '3':

cout<<'\t'<<"骨 科";

break;

case '4':

cout<<'\t'<<"皮肤科";

break;

case '5':

cout<<'\t'<<"口腔科";

break;

}

if (zp2=='Y') //是否为专家门诊

cout<<"\t\t"<<" 是";

else cout<<"\t\t"<<"不 是";

if (zp2=='Y') //挂号费用

cout<<"\t\t"<<"10";

else cout<<"\t\t"<<"5";

cout<<"\n\n================================================================================";

cout<<"\n\n\t\t1.重新填写!\t\t2.信息正确!";

cout<<"\n\n\n请输入您要选择菜单项(1,2):";

cin>>cls;

if (strlen(cls)==1)

cls1=tolower(cls[0]); //变为小写字母

}

if (cls1=='2')

{

P[I].age =age2;

P[I].room=room2;

strcpy(P[I].num,temp1);

P[I].sex=sex2-'0';

P[I].zp=zp2;

I++;

return 0;

}

} //重复输入记录(完)

}

//--------------挂号功能菜单函数------------------

int hospital::gh\_menu()

{

int qu[5],i,fh; //qu数组用于接收返回值 i数值下标 fh返回的开关

char gn, temp[1000]=""; //为了避免输入多个字符时会产生的错误，例：123，-2等等

//调用统计功能函数求出所需要的值

do //控制返回操作

{

for(i=0; i<5; i++) qu[i]=2; //返回时重新赋初值

fh=0;

i=0;

gn='0';

strcpy(temp,"");

clear();

while(gn!='1'&&gn!='2'&&gn!='3'&&gn!='4'&&gn!='q'&&gn!='r') //只能输入1、2或者3

{

clear();

cout<<"\n\n\n\t\t\t===请选择您需要的功能===\n";

cout << "\n\t=============================================================\n";

cout<<"\n\t1.挂当日号\t2.预约挂号\t3.挂急诊号\t4.挂号向导\n";

cout <<"\n\t\t\tr.返 回!\tq.退 出!\n";

cout << "\n\t=============================================================\n";

cout<<"\t请输编号(1,2,3,4,r,q):";

cin>>temp; //输入放在临时变量temp中

if (strlen(temp)==1) gn=tolower(temp[0]);//输入多个字符时跳过

}

switch(gn)

{

case '1':

qu[0]=gdrh();

break; //挂当日号

case '2':

qu[1]=yygh();

break; //预约挂号

case '3':

qu[2]=gjzh();

break; //挂急诊号

case '4':

qu[3]=ghxd();

break; //挂号向导

case 'r':

return(1); //表示选择了"返回"

case 'q':

qu[4]=0; //正常退出程序

}

for(i=0; i<5; i++)

if (qu[i]==0) //已经选择了"退出"

{

system("cls");

cout<<"\n\n\n\n\t\t\t您确定要退出本系统吗?(Y/N)";

cin>>temp;

if (tolower(temp[0])=='y')

{

f\_write(); //写文件

system("cls");

cout<<"\n\n\n\t\t\t您已成功退出系统!\n\n\n\n\n\n";

exit(0);

}

else

{

fh=1;

break;

}

}

else //选择了"返回"

if (qu[i]==1)

{

fh=1;

break;

}

}

while(fh==1);

return(-1);

}

int hospital::show()

{

char n, temp[1000]=""; //为了避免输入多个字符时会产生的错误，例：123，-2等等

while(TRUE)

{

n = '0'; //每次循环初始化值

while(n!='1'&&n!='2'&&n!='3'&&n!='4'&&n!='5'&&n!='6'&&n!='7'&&n!='r')

{

clear();

cout << "\n\t==============================================================\n\n";

cout<< "\t1.所有名单\t2.内科名单\t3.外科名单\n";

cout<< "\t4.骨科名单\t5.皮肤科名单\t6.口腔科名单\n";

cout<< "\t7.预约名单";

cout << "\tr.返回！\n";

cout << "\n\t==============================================================\n\n";

cout<<"\t请输入选择(1,2,3,4,5,6,7,r)：";

cin>>temp;

if (strlen(temp)==1 ) n=tolower(temp[0]);//输入多个字符时转化为小写

}

if(n == 'r')

{

return 1; //返回

}

clear();

cout << "\n\t==============================================================\n";

cout<< "\t患者编号"<<"\t性别"<<"\t科室"<<"\t急诊"<<"\t专家门诊\n";

switch(n)

{

case '1': //打印所有病人名单

{

for(int i=0; i<I; i++)

{

if(P[i].room != '0')

{

cout<< "\t"<<P[i].num<<" \t"<<P[i].sex<<"\t"<<P[i].room<<"\t"<<P[i].jz<<"\t"<<P[i].zp<<endl;

}

}

break;

}

case '2': //打印内科名单

{

for(int i=0; i<I; i++)

{

if(P[i].room == '1')

cout<< "\t"<<P[i].num<<" \t"<<P[i].sex<<"\t"<<P[i].room<<"\t"<<P[i].jz<<"\t"<<P[i].zp<<endl;

}

break;

}

case '3': //打印外科名单

{

for(int i=0; i<I; i++)

{

if(P[i].room=='2')

cout<< "\t"<<P[i].num<<" \t"<<P[i].sex<<"\t"<<P[i].room<<"\t"<<P[i].jz<<"\t"<<P[i].zp<<endl;

}

break;

}

case '4': //打印骨科名单

{

for(int i=0; i<I; i++)

{

if(P[i].room=='3')

cout<< "\t"<<P[i].num<<" \t"<<P[i].sex<<"\t"<<P[i].room<<"\t"<<P[i].jz<<"\t"<<P[i].zp<<endl;

}

break;

}

case '5': //打印皮肤科名单

{

for(int i=0; i<I; i++)

{

if(P[i].room=='4')

cout<< "\t"<<P[i].num<<" \t"<<P[i].sex<<"\t"<<P[i].room<<"\t"<<P[i].jz<<"\t"<<P[i].zp<<endl;

}

break;

}

case '6': //打印口腔科名单

{

for(int i=0; i<I; i++)

{

if(P[i].room=='5')

cout<< "\t"<<P[i].num<<" \t"<<P[i].sex<<"\t"<<P[i].room<<"\t"<<P[i].jz<<"\t"<<P[i].zp<<endl;

}

break;

}

case '7': //打印口腔科名单

{

for(int i=0; i<L; i++)

{

if(Q[i].room != '0')

cout<< "\t"<<Q[i].num<<" \t"<<Q[i].sex<<"\t"<<Q[i].room<<"\t"<<Q[i].jz<<" \t"<<Q[i].zp<<" \t"<<Q[i].time<<endl;

}

break;

}

}

cout << "\n\t==============================================================\n\n";

cout << "\n\t";

system("pause"); //暂停

}

}

//--------------统计功能菜单函数------------------

int hospital::tj\_menu()

{

int qu[6],i,fh; //qu数组用于接收返回值 i数值下标 fh返回的开关

char gn, temp[1000]=""; //为了避免输入多个字符时会产生的错误，例：123，-2等等

tj(); //调用统计函数

do //控制返回操作

{

for(i=0; i<5; i++) qu[i]=2; //返回时重新赋初值

fh=0;

i=0;

gn='0';

strcpy(temp,"");

clear();

while(gn!='1'&&gn!='2'&&gn!='3'&&gn!='4'&&gn!='5'&&gn!='q'&&gn!='r') //只能输入1、2或者3

{

clear();

cout<<"\n\n\n\t\t\t===请选择您需要的功能===\n";

cout << "\n\t==============================================================\n\n";

cout<<"\t1.查看挂号情况\t2.各科就诊人数\t3.患者年龄段\t4.挂号收入情况\n\n\t5.打印病人名单";

cout <<"\tr.返 回!\tq.退 出!\n";

cout << "\n\t==============================================================\n";

cout<<"\n\t请输入编号(1,2,3,4,5,6,r,q):";

cin>>temp; //输入放在临时变量temp中

if (strlen(temp)==1) gn=tolower(temp[0]);//输入多个字符时跳过

}

switch(gn) //被调用的各函数只起输出作用各函数都返回1

{

case '1':

qu[0]=ghrs();

break; //调用查看挂号情况函数

case '2':

qu[1]=jzrs();

break; //调用各科就诊人数函数

case '3':

qu[2]=nlb();

break; //调用各科就诊年龄比函数

case '4':

qu[3]=ghsr();

break; //调用查看收入函数

case '5':

qu[4]=show();//调用打印函数

break;

case 'r':

return(1); //表示选择了"返回"

case 'q':

qu[5]=0; //正常退出程

}

for(i=0; i<6; i++)

if (qu[i]==0) //已经选择了"退出"

{

system("cls");

cout<<"\n\n\n\n\t\t\t您确定要退出本系统吗?(Y/N)";

cin>>temp;

if (tolower(temp[0])=='y')

{

f\_write();

system("cls");

cout<<"\n\n\n\t\t\t您已成功退出系统!\n\n\n\n\n\n";

exit(0);

}

else

{

fh=1;

break;

}

}

else //选择了"返回"

if (qu[i]==1)

{

fh=1;

break;

}

}

while(fh==1);

return(-1);

}

//--------------主菜单函数------------------

void hospital::menu()

{

f\_read();

{

int qu[3],i,fh; //qu数组用于接收返回值 i数值下标 fh返回的开关

char gn, temp[1000]=""; //为了避免输入多个字符时会产生的错误

do //控制返回操作

{

for(i=0; i<3; i++) qu[i]=2; //选择返回 时重新赋初值

fh=0;

i=0;

gn='0';

strcpy(temp,"");

clear(); //清屏

while(gn!='1'&&gn!='2'&&gn!='3'&&gn!='q') //只能输入1、2或者q

{

clear();

cout<<"\n\n\n\t\t\t===欢迎使用医院候诊管理系统===\n";

cout << "\n\t=============================================================\n\n";

cout<<"\n\t1.挂号功能\t2.统计功能\t3.查找与退挂\tq.退出\n";

cout << "\n\n\t=============================================================\n";

cout<<"\n\t请输编号(1,2,q):";

cin>>temp; //输入放在临时变量temp中

if (strlen(temp)==1) gn=tolower(temp[0]);//输入多个字符时跳过

}

switch(gn)

{

case '1':

qu[0]=gh\_menu();

break; //调用挂号功能函数

case '2':

qu[1]=tj\_menu();

break; //调用统计功能的界面函数

case '3':

qu[2]=cz\_menu(); //调用查找函数

break;

case 'q':

qu[3]=0; //正常退出程序

}

for(i=0; i<4; i++)

if (qu[i]==0) //已经选择了"退出"

{

system("cls");

cout<<"\n\n\n\n\t\t\t您确定要退出本系统吗?(Y/N)";

cin>>temp;

if (tolower(temp[0])=='y')

{

f\_write(); //写文件

system("cls");

cout<<"\n\n\n\t\t\t您已成功退出系统!\n\n\n\n\n\n";

exit(0);

}

else

{

fh=1;

break;

}

}

else //选择了"返回"

if (qu[i]==1)

{

fh=1;

break;

}

}

while(fh==1);

}

}

//--------------文件写出函数------------------

void hospital::f\_write()

{

int NUM = 50;

tj();

FILE \*fp1,\*fp2, \*fp3, \*fp4, \*fp6;

fp2 = fopen("全局变量.txt", "wb");

fp1 = fopen("患者名单.txt", "wb"); //只写的方式打开

fp3 = fopen("预约名单.txt", "wb");

fp4 = fopen("挂号人数.txt", "wb");

//fp5 = fopen("排序结构体.txt", "wb");

fp6 = fopen("各科人数.txt", "wb");

fprintf(fp2, "%ld\n%ld\n", I, L); //保存全局变量

fwrite(Q,sizeof(struct Patient),L, fp3); //保存预约名单

fwrite(&G,sizeof(struct ghfl),1, fp4); //保存患者信息

//fwrite(J,sizeof(struct Paixu),5, fp5); //保存患者信息

fwrite(renshu,sizeof(renshu[0]),5, fp6); //保存各科人数

fwrite(P,sizeof(struct Patient)+2,I, fp1); //保存患者信息

//fclose(fp5);

fclose(fp4);

fclose(fp3);

fclose(fp1);

fclose(fp2);

fclose(fp6);

cout << "\n\n\n\t\t\t"<<"名单保存成功,1秒后自动退出\n\n\n";

for(int i=0; i<NUM; i++)

{

Sleep(20);

cout <<"\t\t\t退出中..."<< i \* 100.0 /NUM <<"%\r";

}

}

//--------------文件读入函数------------------

void hospital::f\_read()

{

int NUM = 50;

FILE \*fp1, \*fp2, \*fp3, \*fp4, \*fp6;

fp2 = fopen("全局变量.txt", "rb");

fp1 = fopen("患者名单.txt", "rb"); //只读的方式打开

fp3 = fopen("预约名单.txt", "rb");

fp4 = fopen("挂号人数.txt", "rb");

//fp5 = fopen("排序结构体.txt", "rb");

fp6 = fopen("各科人数.txt", "rb");

fscanf(fp2, "%ld%ld", &I, &L); //读取全局变量

fread(Q,sizeof(struct Patient),L, fp3); //读取预约名单

fread(&G,sizeof(struct ghfl),1, fp4); //读取患者信息

//fread(J,sizeof(struct Paixu),5, fp5); //读取患者信息

fread(renshu,sizeof(renshu[0]),5, fp6); //读取各科人数

fread(P,sizeof(struct Patient)+2,I, fp1); //读取患者信息

//fclose(fp5);

fclose(fp4);

fclose(fp3);

fclose(fp1);

fclose(fp2);

fclose(fp6);

cout << "\n\n\n\t\t\t" << "文件读取成功,1秒后自动进入系统!\n\n";

for(int i=0; i<NUM; i++)

{

Sleep(20);

cout <<"\t\t\t载入中..."<< i \* 100.0 /NUM <<"%\r";

}

}

//--------------模糊查找-----------------

int hospital::serch()

{

char gn='0',age1[100]="", temp2[1000]=""; //为了避免输入多个字符时会产生的错误

int age2=0;

while(gn!='1'&&gn!='2'&&gn!='3'&&gn!='4'&&gn!='5')

{

clear();

cout<<"\n\n\t\t系统将按照科室和年龄来查找\n\n\n";

cout<<"\t请输入您所挂科室<1,2,3,4,5>：";

cin>>temp2;

if (strlen(temp2)==1)

gn=tolower(temp2[0]);//输入多个字符时跳过

}

while(age2<=0||age2>150) //输入的年龄在1--150之间

{

clear();

cout<<"\n\n\n请输入您的年龄：";

cin>>age1;

if (strlen(age1)<=3)

age2=atoi(age1); //将字符型的age1转换成整型赋给age2;

}

int flag=1;

for(int j=0; j<=I; j++)

{

if(gn==P[j].room&&age2==P[j].age)

{

if(flag==1)

{

clear();

cout << "\n\t\t\t系统查找到以下信息：\n";

cout << "\n\t==============================================================\n";

cout << "\t患者编号"<<"\t性别"<<"\t科室"<<"\t急诊"<<"\t专家门诊\n";

flag=0;

}

cout<< "\t"<<P[j].num<<" \t"<<P[j].sex<<"\t"<<P[j].room<<"\t"<<P[j].jz<<"\t"<<P[j].zp<<endl;

}

}

char ch='0',temp1[10]="";

do

{

cout << "\n\n\n\t\t以上是否有您的信息<Y/N>:";

cin>>temp1;

ch=tolower(temp1[0]);

}

while(ch!='y'&&ch!='n');

switch(ch)

{

case 'y':

{

char temp[10];

long i=-1;

while(i <= 0) //输入的编号必须大于0

{

clear();

cout<<"\n\n\n\t请输入您的诊号，我们将为您打印您的诊单:";

cin>>temp;

i = atoi(temp); //将字符型的age1转换成整型赋给age2;

}

if(i > I)

{

clear();

cout << "\n\n\n\t\t抱歉,不存在这个诊号！\n\n\t";

system("pause");

return 1;

}

cout << "\n\t==============================================================\n";

cout << "\t患者编号"<<"\t性别"<<"\t科室"<<"\t急诊"<<"\t专家门诊\n";

cout << "\t"<<P[i-1].num<<" \t"<<P[i-1].sex<<"\t"<<P[i-1].room<<"\t"<<P[i-1].jz<<"\t"<<P[i-1].zp<<endl;

break;

}

case 'n':

cout << "\n\n\t建议您到服务窗口去查找!\n\n\n\t";

break;

}

system("pause"); //暂停

return 1;

}

//--------------按诊号函数------------------

//形参没有实际用处 只是用来重载

int hospital::serch(int i)

{

clear();

long n=-1;

char temp[100]="";

while(n <= 0) //输入的编号必须大于0

{

clear();

cout<<"\n\n\n\t请输入您的诊号:";

cin>>temp;

n = atoi(temp); //将字符型的age1转换成整型赋给age2;

}

if(n > I)

{

clear();

cout << "\n\n\n\t\t抱歉,不存在这个单号！\n\n\t";

system("pause");

return 1;

}

clear();

cout << "\n\n\n\n";

for(int i=0; i<50; i++)

{

Sleep(10);

cout <<"\t\t\t正在打印您的诊单..."<< i \* 100.0 /50 <<"%\r";

}

clear();

cout << "\n\t==============================================================\n";

cout << "\t患者编号"<<"\t性别"<<"\t科室"<<"\t急诊"<<"\t专家门诊\n";

cout << "\t"<<P[n-1].num<<" \t"<<P[n-1].sex<<"\t"<<P[n-1].room<<"\t"<<P[n-1].jz<<"\t"<<P[n-1].zp<<"\n\n\t";

system("pause"); //暂停

return 1;

}

//--------------查找菜单函数------------------

int hospital::cz\_menu()

{

int qu[5],i,fh; //qu数组用于接收返回值 i数值下标 fh返回的开关

char gn, temp[1000]=""; //为了避免输入多个字符时会产生的错误，例：123，-2等等

do //控制返回操作

{

for(i=0; i<5; i++) qu[i]=2; //返回时重新赋初值

fh=0;

i=0;

gn='0';

strcpy(temp,"");

clear();

while(gn!='1'&&gn!='2'&&gn!='3'&&gn!='4'&&gn!='q'&&gn!='r') //只能输入1、2或者3

{

clear();

cout<<"\n\n\n\t\t\t===请选择您需要的功能===\n";

cout << "\n\t==============================================================\n\n";

cout<<"\t1.按诊号查找\t2.模糊查找\t3.退订号 \t4.清除数据\n\n";

cout <<"\tr.返 回!\tq.退 出!\n";

cout << "\n\t==============================================================\n";

cout<<"\n\t请输入编号(1,2,3,4,r,q):";

cin>>temp; //输入放在临时变量temp中

if (strlen(temp)==1) gn=tolower(temp[0]);//输入多个字符时跳过

}

switch(gn) //被调用各函数都返回1

{

case '1':

qu[0]=serch(1);

break; //按诊号查找 ,设置形参只是用来重载,没有实际用处

case '2':

qu[1]=serch();

break; //模糊查找

case '3':

qu[2]=delete1(); //删除单号

break;

case '4':

qu[3]=delete\_file(); //清除数据

case 'r':

return(1); //表示选择了"返回"

case 'q':

qu[4]=0; //正常退出程

}

for(i=0; i<5; i++)

if (qu[i]==0) //已经选择了"退出"

{

system("cls");

cout<<"\n\n\n\n\t\t\t您确定要退出本系统吗?(Y/N)";

cin>>temp;

if (tolower(temp[0])=='y')

{

f\_write();

system("cls");

cout<<"\n\n\n\t\t\t您已成功退出系统!\n\n\n\n\n\n";

exit(0);

}

else

{

fh=1;

break;

}

}

else //选择了"返回"

if (qu[i]==1)

{

fh=1;

break;

}

}

while(fh==1);

return(-1);

}

//--------------有形参的delete1函数------------------

int hospital::delete1(long i)

{

if(i > I)

{

clear();

cout << "\n\n\n\t\t抱歉,不存在这个单号！";

system("pause");

return 1;

}

//处理ren数数组

switch(P[i-1].room)

{

case '1':

renshu[0]--;

break;

case '2':

renshu[1]--;

break;

case '3':

renshu[2]--;

break;

case '4':

renshu[3]--;

break;

case '5':

renshu[4]--;

break;

}

//处理Patient结构体

P[i-1].age = '0';

P[i-1].zp = '0';

P[i-1].room = '0';

strcpy(P[i-1].num, "0");

//处理ghfl结构体

if(strcmp( P[i-1].time ,"") != 0)

{

G.yyrs--;

}

else if(strcmp( P[i-1].jz ,"是") == 0)

{

G.jzrs--;

}

else

{

G.drrs--;

}

clear();

cout << "\n\n\n\t\t单号已经退订成功!\n\t";

system("pause");

return 1;

}

//--------------重载delete1函数------------------

int hospital::delete1()

{

clear();

long i = -1; //初始化为-1

char temp[1000]=""; //为了避免输入多个字符时会产生的错误，例：123，-2等等

while(i <= 0) //输入的编号必须大于0

{

clear();

cout<<"\n\n\n\t请输入要退挂的诊号:";

cin>>temp;

i = atoi(temp); //将字符型的age1转换成整型赋给age2;

}

if(i > I)

{

clear();

cout << "\n\n\n\t\t抱歉,不存在这个单号！\n\n";

system("pause");

return 1;

}

//处理ren数数组

switch(P[i-1].room)

{

case '1':

renshu[0]--;

break;

case '2':

renshu[1]--;

break;

case '3':

renshu[2]--;

break;

case '4':

renshu[3]--;

break;

case '5':

renshu[4]--;

break;

}

//处理Patient结构体

P[i-1].age = '0';

P[i-1].zp = '0';

P[i-1].room = '0';

strcpy(P[i-1].num, "0");

//处理ghfl结构体

if(strcmp( P[i-1].time ,"") != 0)

{

G.yyrs--;

}

else if(strcmp( P[i-1].jz ,"是") == 0)

{

G.jzrs--;

}

else

{

G.drrs--;

}

clear();

cout << "\n\n\n\t\t单号已经退订成功!\n\n";

system("pause");

return 1;

}

//------------删除本地文件--------------------

int hospital::delete\_file()

{

FILE \*fp1, \*fp2, \*fp3, \*fp4, \*fp6;

char temp[1000]=""; //为了避免输入多个字符时会产生的错误，例：123，-2等等

system("cls");

cout<<"\n\n\t\t清除完数据之后将会退出程序，请手动重启程序！";

cout<<"\n\n\n\n\t\t\t您确定要清除全部数据吗?(Y/N)";

cin>>temp;

if (tolower(temp[0])=='y') //选择了是

{

fp2 = fopen("全局变量.txt", "wb");

fp1 = fopen("患者名单.txt", "wb"); //只读的方式打开

fp3 = fopen("预约名单.txt", "wb");

fp4 = fopen("挂号人数.txt", "wb");

fp6 = fopen("各科人数.txt", "wb");

remove("全局变量.txt");

remove("患者名单.txt");

remove("预约名单.txt");

remove("挂号人数.txt");

remove("各科人数.txt");

fclose(fp4);

fclose(fp3);

fclose(fp1);

fclose(fp2);

fclose(fp6);

system("cls");

cout << "\n\n\n\t\t本地数据已经全部清除!\n\n";

exit(1);

}

else //选择了否

{

return 1;

}

cout << "\n\n\n\t\t本地数据已经全部清除!\n\n";

system("pause");

return 1;

}

# main.cpp

#include"hospital.h"

//-----------主函数--------------

int main()

{

system("color 3F");

hospital hp; //定义一个对象

hp.menu();

return 0;

}

# hospital.h

#ifndef HOSPITAL\_H\_INCLUDED

#define HOSPITAL\_H\_INCLUDED

#include <windows.h>

#include <ctime>

#include <cstdio>

#include <cstring>

#include <iostream>

#include <stdio.h>

#include <stdlib.h>

class hospital //定义医院类

{

public :

int delete\_file(); //删除本地文件

int delete1(); //重载删除函数

int delete1(long i);//删除函数

int cz\_menu(); //查找删除函数菜单

int serch(int i); //重载serch函数实现两种不同查找方式

int serch(); //查找删除函数

void f\_write(); //写文件函数，用于在程序关闭之前保存数据到本地

void f\_read(); //读文件函数，用于在程序打开之前读取数据要程序中

int input(); //输入函数

void menu(); //主菜单函数

void tj(); //统计功能函数

int gh\_menu(); //挂号功能选择菜单

int tj\_menu(); //统计功能选择菜单函数

int gjzh(); //挂急诊号

int yygh(); //预约挂号函数

int ghxd(); //挂号向导函数

int ghrs(); //挂号情况统计函数

int jzrs(); //各科就诊人数统计函数

int nlb(); //年龄人数统计函数

int ghsr(); //挂号收入情况

int gdrh(); //挂当日号函数

int show(); //展示病人名单

void room(char room); //分诊室函数

};

#endif // HOSPITAL\_H\_INCLUDED