

课程设计报告书

课程设计题目：医院病历管理系统

学生：叶霆振 卢婧 王珂

班级：电气 2004 班 电气 2007 班

指导教师：黄庆凤 江敏

2021 年 6 月 9 日

一 . 选题

医院病历管理系统

二 . 需求分析

用类和函数完成一个医院病历管理系统，此系统的具体功能要求如下：

1. 添加患者信息功能

用户从键盘输入每个患者的姓名，性别，年龄，电话，病房，症状等

2. 显示患者信息功能

将患者信息显示在界面，同时实现将患者相关信息保存至本地文件的功能，将信息汇总保存到一个 txt 文档，用户输入或删除患者的相关信息可随时保存到文档中

3. 查找患者信息功能

通过关键词患者的姓名在医院病历管理系统中查找相关患者的信息，并显示其姓名，性别，年龄，电话，病房，症状等

4. 修改患者信息功能

输入关键词患者的姓名，判断该患者是否在病例系统中留有信息。若有，则对其信息进行修改，同时更新存储信息文档，若无，则不进行修改

5. 删除患者相关信息功能

通过输入关键词患者的姓名，在病历系统中删除相关患者的信息

6. 清空患者信息功能

将录入的患者信息全部删除，回归到初始状态

7. 清除本地存有患者信息的文件功能

将建立的输出流文档清空

三 . 系统设计

在理解需求和分析需求的基础上，确定为了实现这些功能，目标程序所需要的数结构、算法和程序流程。然后，对已经确定的数据结构、算法等进一步细化出所有的子功能，确定出模块间的接口，描述每个模块的处理过程，确定输入输出数据的内外部形式。

1. 数据结构设计：病历信息类与病历库类；

//病历信息

```
class Patient_Cases_Database
```

```
{
```

```
public:
```

```
string Name;    //患者姓名
```

```
int Gender = 1;    //患者性别(默认为男)
```

```
int Age = 0;    //患者年龄
```

```
string Phone;    //患者电话号码
```

```
string Location;    //患者病床
```

```
string Symptom;    //患者症状
```

```
};
```

//医院病历库

```
class Hospital_Cases
```

```
{
```

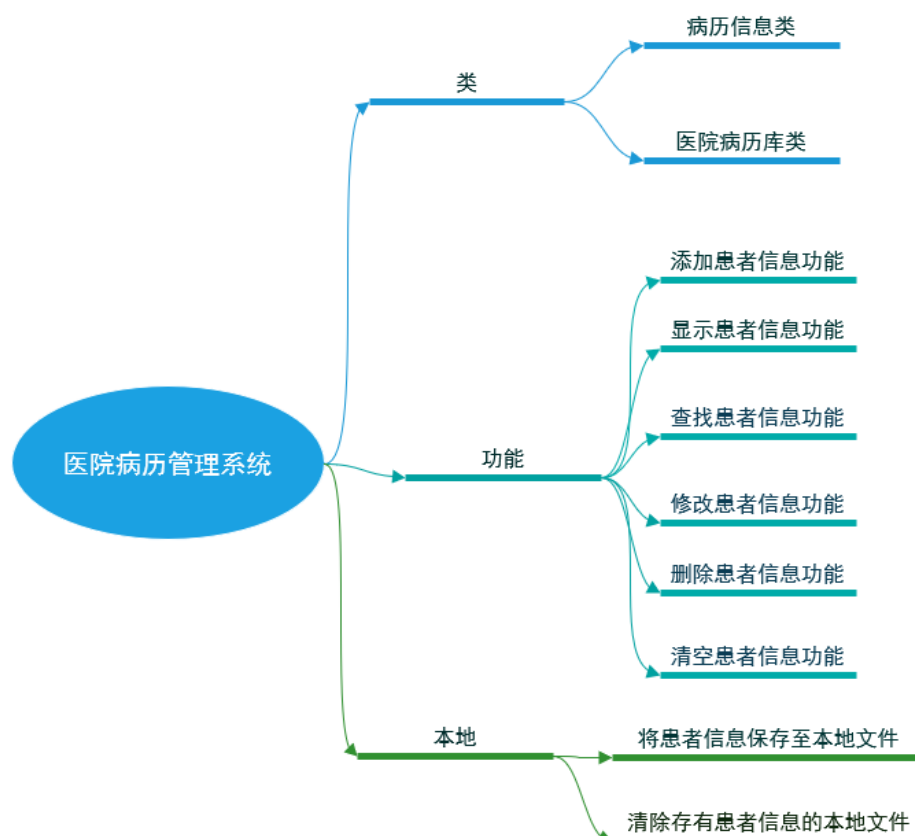
```
public:
```

```
    Patient_Cases_Database Patient_case[Max];    //医院病历库数组
```

```
    int Cases_Count = 0;    //医院病历库中病历的个数
```

```
};
```

2. 项目总体框架



3. 项目设计思路

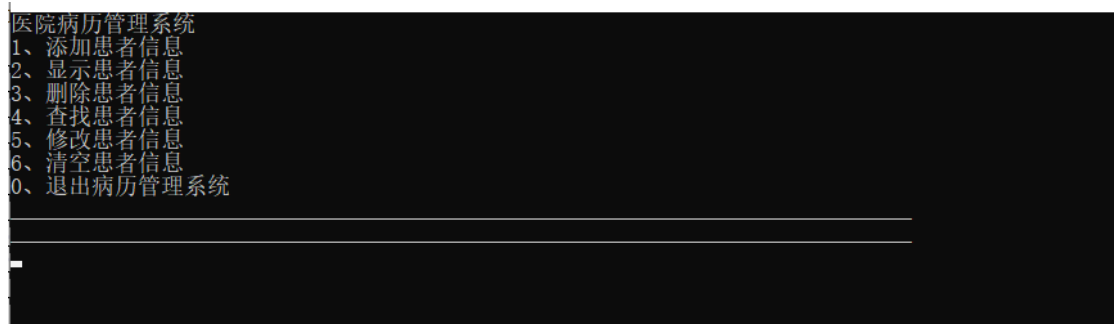
- (1) 了解通用病历系统的所有功能，确定代码需要完成的所有功能
- (2) 通过定义类，确定病历系统部分的属性
- (3) 按照所需功能依次编写不同函数的代码，并将函数逐一测试，查找 Bug
- (4) 实现所有功能后，扩展将程序中的联系人存储到本地 .txt 文件中的功能，实现关闭程序后仍能查看所有联系人

4. 运行环境

Visual Studio 2019

四 . 界面设计

1. 界面设计：如图所示；



```
医院病历管理系统
1、添加患者信息
2、显示患者信息
3、删除患者信息
4、查找患者信息
5、修改患者信息
6、清空患者信息
0、退出病历管理系统
```

2. 交互设计：

通过键盘输入指令，实现程序与用户的交互

五 . 系统实现

备注：由于代码行数过多，仅选取关键函数、功能函数的声明和类的定义进行展示

源代码：

```
#include<iostream>
#include<string>
#include<fstream>//用于写入本地文档
using namespace std;

const int Max = 1000; //医院住院最大人数
```

//病历信息

class Patient_Cases_Database

```
{
public:
    string Name;      //患者姓名
    int Gender = 1;    //患者性别(默认为男)
    int Age = 0;       //患者年龄
    string Phone;      //患者电话号码
    string Location;   //患者病床
    string Sympyom;    //患者症状
};
```

//医院病历库

class Hospital_Cases

```
{
public:
    Patient_Cases_Database Patient_case[Max];    //医院病历库数组
    int Cases_Count = 0;                        //医院病历库中病历的个数
};
```

//添加患者病历

void Add_Cases(Hospital_Cases* your_case) {省略}

//显示患者病历

void Show_Cases(Hospital_Cases* your_case) {省略}

//删除患者病历

void Delete_Cases(Hospital_Cases* your_case) {省略}

//查找患者

void Find_Cases(Hospital_Cases* your_case) {省略}

//修改患者信息

void Change_Cases(Hospital_Cases* your_case) {省略}

//清空患者

void Clear(Hospital_Cases* your_case) {省略}

//检测患者是否存在

```
int Test_Cases(Hospital_Cases* your_case, string name)
{
    for (int i = 0; i < your_case->Cases_Count; i++)
    {
        if (your_case->Patient_case[i].Name == name)
```

```
    {
        return i;
    }
}
return -1;
}

//将患者信息存在本地
void Write_Cases(Hospital_Cases* your_case)
{
    //在 data.txt 中写入所有的患者
    ofstream Write;
    //在 E 盘中自动生成 data.txt 文件储存患者信息
    Write.open("E:\\data.txt");
    for (int i = 0; i < your_case->Cases_Count; i++)
    {
        Write << " 患者姓名: " << your_case->Patient_case[i].Name;
        Write << " 患者性别: " << (your_case->Patient_case[i].Gender != 2 ? "男" : "女");
        Write << " 患者年龄: " << your_case->Patient_case[i].Age;
        Write << " 患者电话: " << your_case->Patient_case[i].Phone;
        Write << " 患者病房: " << your_case->Patient_case[i].Location;
        Write << " 患者症状: " << your_case->Patient_case[i].Symptom;
        Write << endl;
    }
    Write.close();
}

//显示主菜单
void Show_Menu()
{
    cout << "医院病历管理系统" << endl;
    cout << "1、添加患者信息" << endl;
    cout << "2、显示患者信息" << endl;
    cout << "3、删除患者信息" << endl;
    cout << "4、查找患者信息" << endl;
    cout << "5、修改患者信息" << endl;
    cout << "6、清空患者信息" << endl;
    cout << "0、退出病历管理系统" << endl;
    cout << "
    << endl;
    cout << "
    << endl;
}
```

```
int main()
{
    //记录用户输入的信息
    int select = 0;
    //定义一个病例库类
    Hospital_Cases your_Case;
    //设置当前人数为零
    your_Case.Cases_Count = 0;

    while (true)
    {
        //显示主菜单
        Show_Menu();

        cin >> select;
        switch (select)
        {
            case 1://添加患者信息
                Add_Cases(&your_Case);
                break;
            case 2://显示患者信息
                Show_Cases(&your_Case);
                break;
            case 3://删除患者信息
                Delete_Cases(&your_Case);
                break;
            case 4://查找患者信息
                Find_Cases(&your_Case);
                break;
            case 5://修改患者信息
                Change_Cases(&your_Case);
                break;
            case 6://清空患者信息
                Clear(&your_Case);
                break;
            case 0://退出病历管理系统
                cout << "欢迎下次使用!" << endl;
                system("pause");
                return 0;
                break;
            default:
                break;
        }
    }
}
```

```
system("pause");//按任意键继续  
return 0;  
}
```

六. 测试与调试

1. 遇到的问题及解决方法

对病历管理系统各项功能进行测试，出现了如下问题并进行了如下修改：

- (1) 在 .exe 程序运行时，多次步骤后程序界面较为混乱，后引入 system("cls") 清屏解决
- (2) 删除、修改患者信息函数中，不知如何能对本地的 data.txt 文件进行删除、修改单个联系的人操作，之后通过调用 Write_Cases() 函数解决问题

2. 修改后运行结果

```
医院病历管理系统  
1、添加患者信息  
2、显示患者信息  
3、删除患者信息  
4、查找患者信息  
5、修改患者信息  
6、清空患者信息  
0、退出病历管理系统
```

```
1  
请输入患者姓名：  
张三  
请输入患者性别：  
1 --- 男  
2 --- 女  
1  
请输入患者年龄：  
24  
请输入患者电话：  
13467349023  
请输入患者病房：  
302  
请输入患者症状：  
肺炎  
添加成功！  
请按任意键继续. . .
```

```
医院病历管理系统  
1、添加患者信息  
2、显示患者信息  
3、删除患者信息  
4、查找患者信息  
5、修改患者信息  
6、清空患者信息  
0、退出病历管理系统
```

```
2  
患者姓名：张三 患者性别：男 患者年龄：24 患者电话：13467349023 患者病房：302 患者症状：肺炎  
请按任意键继续. . .
```


七. 心得与体会

在此次项目设计中,从技术上,对C++语言有了更加深入的了解,对程序框架有了更为清楚的认识,提高了编程水平,同时完成了对课程的复习以及对所学知识的检验,培养了思考与解决问题的能力;在团队合作上,通过交流与讨论,我们确定了程序的算法流程,核心功能和数据结构等,通过合作解决了出现的问题,同时在交流中我们收获了很多灵感,弥补了知识上的不足,实现了融会贯通,同时锻炼了沟通能力和协调能力,受益匪浅。

声明: 我将我的 C++ 课程设计项目 (医院病历管理系统) 授权给 (C++ 程序设计基础) MOOC 教学团队使用, 允许其在互联网以及所需要的教学中传播、宣传和展示。

签名: 叶霆振 卢婧 王珂