**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**

1. **选题**

通讯录管理系统

1. **需求分析**

**用结构体和函数完成一个通讯录管理系统，此系统的具体功能要求如下：**

1. **添加联系人功能**

用户从键盘输入每个联系人的姓名，性别，年龄，电话，地址

1. **显示联系人功能**

将联系人信息显示在界面，同时实现将联系人保存至本地文件功能，将联系人信息汇总保存到一个txt文档，用户输入或删除联系人信息可随时保存到文档中

1. **查找联系人功能**

通过关键词联系人的姓名在通讯录中查找相关联系人的信息，并显示其姓名，性别，年龄，电话，地址

1. **修改联系人功能**

输入关键词联系人的姓名，判断该联系人是否在通讯录中留有信息，若有将对其信息进行修改，同时更新存储信息文档，若无，则无法修改

1. **删除联系人功能**

通过输入关键词联系人的姓名在通讯录中删除相关联系人的信息

1. **清空联系人功能**

将录入的通讯录信息全部删除，回归到初始状态

1. **清除本地的联系人文件功能**

将建立的输出流文档删除

1. **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据结构、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：联系人结构体；**

//定义一个联系人结构体

struct People\_Contact

{

string Name; //姓名

int Gender = 1; //性别

int Age = 0; //年龄

string Phone; //电话号码

string Add; //住址

};

//定义一个通信录结构体

struct Address\_your\_Books

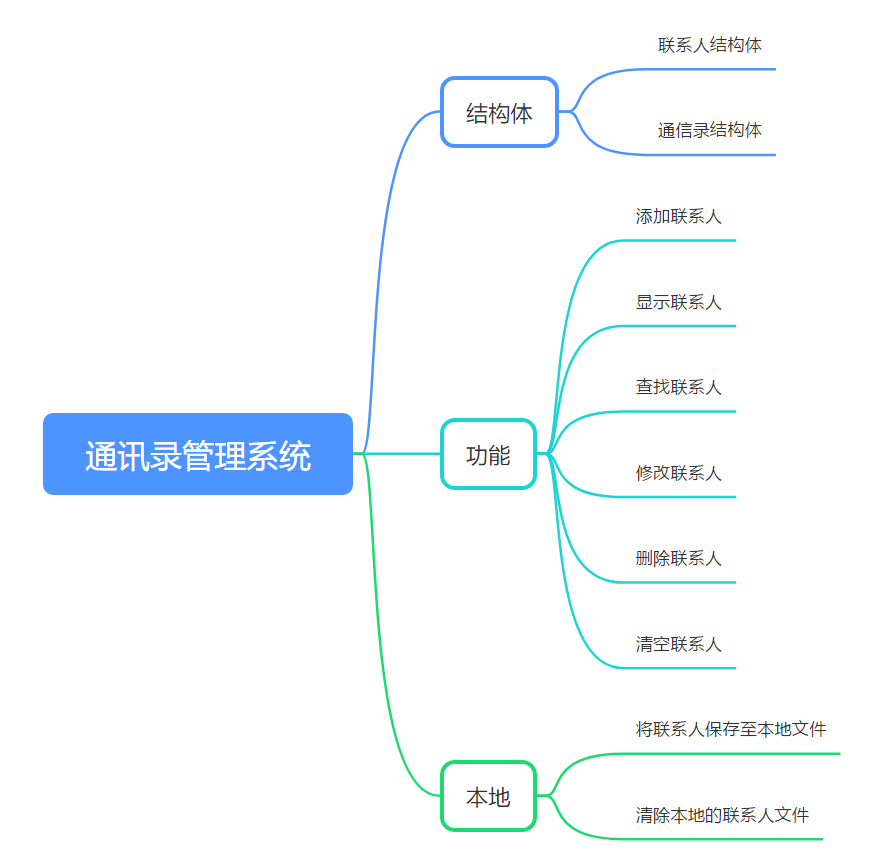
{

People\_Contact contactsPeople[Max]; //通信录联系人信息数组

int aCount = 0; //通信录中人员的个数

};

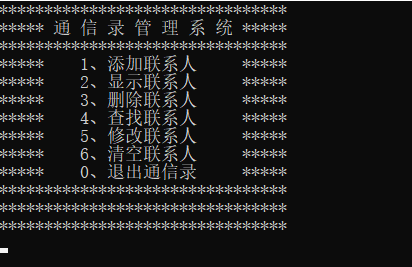
1. **项目总体框架**



1. **项目设计思路**
2. 了解正常通讯录的所有功能，确定代码需要完成的所有功能
3. 通过定义结构体，确定通讯录部分的属性
4. 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
5. 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
6. **运行环境**

Visual Studio 2019

1. **界面设计**
2. **界面设计：如图所示；**

****

1. **交互设计：通过键盘输入实现交互**
2. **系统实现**

源代码：

#include<iostream>

#include<string>

#include<fstream>

using namespace std;

const int Max = 1000;//通讯录最大人数

//定义一个联系人结构体

struct People\_Contact

{

string Name; //姓名

int Gender = 1; //性别

int Age = 0; //年龄

string Phone; //电话号码

string Add; //住址

};

//定义一个通信录结构体

struct Address\_your\_Books

{

People\_Contact contactsPeople[Max]; //通信录联系人信息数组

int aCount = 0; //通信录中人员的个数

};

//添加联系人

void Add\_People(Address\_your\_Books\* your\_Book)

{

//判断是否满员

if (your\_Book->aCount == Max)

{

cout << "通信录已满，无法添加！" << endl;

return;

}

else

{

//姓名

string name;

cout << "请输入姓名： " << endl;

cin >> name;

your\_Book->contactsPeople[your\_Book->aCount].Name = name;

//性别

int gender;

cout << "请输入性别： " << endl;

cout << "1 ---- 男 " << endl;

cout << "2 ---- 女 " << endl;

//判断是否输入正确序号，正确则记录，否则重新输入

while (true)

{

cin >> gender;

if (gender == 1 || gender == 2)

{

your\_Book->contactsPeople[your\_Book->aCount].Gender = gender;

break;

}

cout << "输入有误请重新输入！" << endl;

}

//年龄

int age;

cout << "请输入年龄： " << endl;

cin >> age;

your\_Book->contactsPeople[your\_Book->aCount].Age = age;

//电话

string phone;

cout << "请输入电话： " << endl;

cin >> phone;

your\_Book->contactsPeople[your\_Book->aCount].Phone = phone;

//地址

string add;

cout << "请输入地址： " << endl;

cin >> add;

your\_Book->contactsPeople[your\_Book->aCount].Add = add;

//个数加一

your\_Book->aCount += 1;

cout << "添加成功！" << endl;

system("pause");

//清屏

system("cls");

}

}

//显示联系人

void Show\_People(Address\_your\_Books\* your\_Book)

{

//判断有无联系人

if (your\_Book->aCount == 0)

{

cout << "暂无联系人，请先添加一下!" << endl;

}

else

{

//显示所有联系人

for (int i = 0; i < your\_Book->aCount; i++)

{

cout << " 姓名：" << your\_Book->contactsPeople[i].Name;

cout << " 性别：" << (your\_Book->contactsPeople[i].Gender != 2 ? "男" : "女");

cout << " 年龄：" << your\_Book->contactsPeople[i].Age;

cout << " 电话：" << your\_Book->contactsPeople[i].Phone;

cout << " 住址：" << your\_Book->contactsPeople[i].Add;

cout << endl;

}

}

//在data.txt中写入所有的联系人

ofstream Write;

//在E盘中自动生成data.txt文件储存联系人信息

Write.open("E:\\data.txt");

for (int i = 0; i < your\_Book->aCount; i++)

{

Write << " 姓名：" << your\_Book->contactsPeople[i].Name;

Write << " 性别：" << (your\_Book->contactsPeople[i].Gender != 2 ? "男" : "女");

Write << " 年龄：" << your\_Book->contactsPeople[i].Age;

Write << " 电话：" << your\_Book->contactsPeople[i].Phone;

Write << " 住址：" << your\_Book->contactsPeople[i].Add;

Write << endl;

}

Write.close();

system("pause");

system("cls");

}

//检测联系人是否存在

int Test\_People(Address\_your\_Books\* your\_Book, string name)

{

for (int i = 0; i < your\_Book->aCount; i++)

{

if (your\_Book->contactsPeople[i].Name == name)

{

return i;

}

}

return -1;

}

//查找联系人

void Find\_People(Address\_your\_Books\* your\_Book)

{

//记录用户输入的姓名

string name;

cout << "请输入您要查找的用户姓名" << endl;

cin >> name;

//判断联系人是否存在

int index = Test\_People(your\_Book, name);

if (index > -1)

{

//存在

cout << "姓名：" << your\_Book->contactsPeople[index].Name << endl;

cout << "性别：" << your\_Book->contactsPeople[index].Gender << endl;

cout << "年龄：" << your\_Book->contactsPeople[index].Age << endl;

cout << "电话：" << your\_Book->contactsPeople[index].Phone << endl;

cout << "住址：" << your\_Book->contactsPeople[index].Add << endl;

}

else

{

//不存在

cout << "查无此人" << endl;

}

system("pause");

system("cls");

}

//修改联系人

void Change\_People(Address\_your\_Books\* your\_Book)

{

//记录用户输入的姓名

string name;

cout << "请输入您要修改的用户姓名" << endl;

cin >> name;

//判断联系人是否存在

int index = Test\_People(your\_Book, name);

if (index > -1)

{

//存在

//姓名

string name;

cout << "请输入修改后的姓名： " << endl;

cin >> name;

your\_Book->contactsPeople[index].Name = name;

//性别

int gender;

cout << "请输入修改后的性别： " << endl;

cout << "1 ---- 男 " << endl;

cout << "2 ---- 女 " << endl;

while (true)

{

cin >> gender;

if (gender == 1 || gender == 2)

{

your\_Book->contactsPeople[index].Gender = gender;

break;

}

cout << "输入有误请重新输入！" << endl;

}

//年龄

int age;

cout << "请输入修改后的年龄： " << endl;

cin >> age;

your\_Book->contactsPeople[index].Age = age;

//电话

string phone;

cout << "请输入修改后的电话： " << endl;

cin >> phone;

your\_Book->contactsPeople[index].Phone = phone;

//地址

string add;

cout << "请输入修改后的地址： " << endl;

cin >> add;

your\_Book->contactsPeople[index].Add = add;

cout << "修改成功!" << endl;

//更改本地文档

Show\_People(your\_Book);

}

else

{

//不存在

cout << "查无此人" << endl;

}

system("pause");

system("cls");

}

//删除联系人

void Delete\_People(Address\_your\_Books\* your\_Book)

{

//记录用户输入的姓名

string name;

cout << "请输入您要删除的用户姓名：" << endl;

cin >> name;

//判断联系人是否存在

int index = Test\_People(your\_Book, name);

//存在

if (index > -1)

{

for (int i = index; i < your\_Book->aCount - 1; i++)

{

//往前移位操作

People\_Contact temp = your\_Book->contactsPeople[i];

your\_Book->contactsPeople[i] = your\_Book->contactsPeople[i + 1];

your\_Book->contactsPeople[i + 1] = temp;

}

//减少一个记录

your\_Book->aCount--;

//更改本地文档

Show\_People(your\_Book);

}

else//不存在

{

cout << "查无此人" << endl;

}

system("pause");

system("cls");

}

//清空联系人

void Clear(Address\_your\_Books\* your\_Book)

{

string choose = "N";

cout << "您确定要清空通信录吗？(Y 确认/N 取消)" << endl;

while (true)

{

cin >> choose;

if (choose == "Y" || choose == "N")

{

break;

}

else

{

cout << "输入有误请重新输入!!" << endl;

}

}

if (choose == "Y")

{

your\_Book->aCount = 0;

cout << "清空成功!!" << endl;

//清空E盘中data.txt文件

ofstream Delete;

Delete.open("E:\\data.txt");

Delete.close();

}

else

{

cout << "操作已取消!!" << endl;

}

system("pause");

system("cls");

}

//显示主菜单

void Show\_Menu()

{

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

cout << "\*\*\*\*\* 通 信 录 管 理 系 统 \*\*\*\*\*" << endl;

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

cout << "\*\*\*\*\* 1、添加联系人 \*\*\*\*\*" << endl;

cout << "\*\*\*\*\* 2、显示联系人 \*\*\*\*\*" << endl;

cout << "\*\*\*\*\* 3、删除联系人 \*\*\*\*\*" << endl;

cout << "\*\*\*\*\* 4、查找联系人 \*\*\*\*\*" << endl;

cout << "\*\*\*\*\* 5、修改联系人 \*\*\*\*\*" << endl;

cout << "\*\*\*\*\* 6、清空联系人 \*\*\*\*\*" << endl;

cout << "\*\*\*\*\* 0、退出通信录 \*\*\*\*\*" << endl;

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

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

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

}

int main()

{

//记录用户输入的信息

int select = 0;

//定义一个通信录结构体

Address\_your\_Books your\_Book;

//设置当前人数为零

your\_Book.aCount = 0;

while (true)

{

//显示主菜单

Show\_Menu();

cin >> select;

switch (select)

{

case 1://1、添加联系人

Add\_People(&your\_Book); //地址传递

break;

case 2://2、显示联系人

Show\_People(&your\_Book); //地址传递

break;

case 3://3、删除联系人

Delete\_People(&your\_Book);

break;

case 4://4、查找联系人

Find\_People(&your\_Book);

break;

case 5://5、修改联系人

Change\_People(&your\_Book);

break;

case 6://6、清空联系人

Clear(&your\_Book);

break;

case 0://0、退出通信录

cout << "欢迎下次使用!" << endl;

system("pause");

return 0;

break;

default:

break;

}

}

system("pause");//按任意键继续

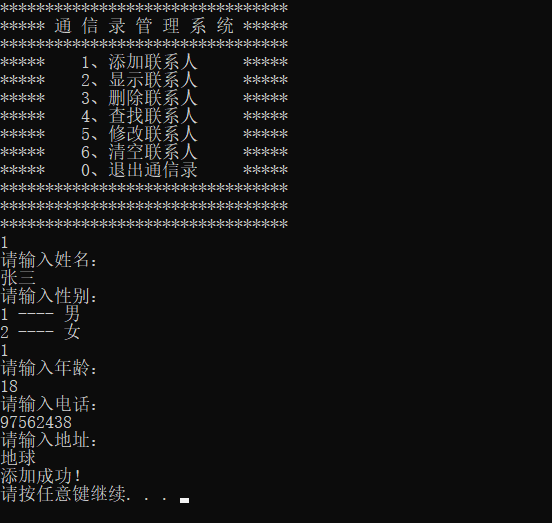
return 0;

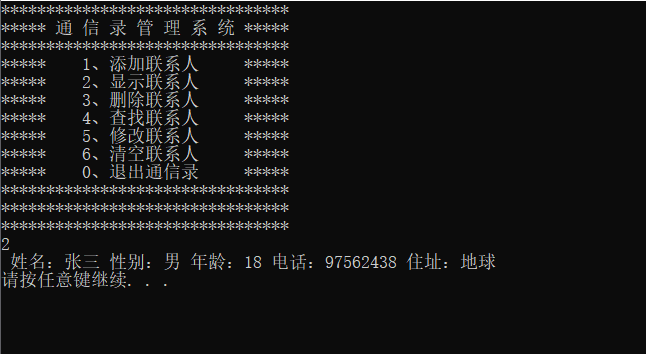
}

1. **测试与调试**
2. **遇到的问题及解决方法**

对通讯录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、修改联系人函数中，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Show\_People()；函数解决问题
3. **修改后运行结果**





1. **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（通讯录管理系统）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