**南 京 师 范 大 学**

**《面向对象程序设计》**

**课程设计报告**



**编号及题目： 面向对象的同学通讯录系统**

**学 院：计算机与电子信息学院/人工智能学院**

**专 业：计算机科学与技术**

**姓 名： 黄艺杰**

**学 号： 21190630**

**任课教师： 曲维光**

**计算机与电子信息学院/人工智能学院**

**2021年 9 月 3 日**

1. 设计内容
2. 题目原文

通过建立班级电子通讯录，从而方便同学间通讯信息的共享，实现个人信息便捷的管理和维护，方便同学联系。

本系统的功能要求如下：

1. 该系统主要管理同学通讯录的相关信息。
2. 通讯录信息主要包括：姓名、院系、班级、手机、家庭电话、电子邮件、通讯录地址、邮编等内容。
3. 具有通讯录管理的基本操作：通讯录信息的添加、修改、删除和查询，查询功能应该支持多种不同方式查询，包括：根据单字段信息查询、多字段联合查询等方便功能。
4. 需求分析

设计相关类完成通讯录并具有以下管理功能：

添加：能添加学生的通讯信息（数据保存在文件中）

修改：对某位同学的通讯信息进行修改

删除：删除某位同学的通讯信息

查询：支持单字段和多字段查询某位同学通讯信息

退出：使用完成后安全退出系统

1. 系统环境

1.硬件环境：个人计算机

2.软件环境：Windows操作系统 Visual Studio

1. 设计思想描述

继承

添加、显示、修改、删除、查询、退出

手机、家庭电话、电子邮件、通讯录地址、邮编

姓名、院系、班级

学生类

通讯录类

管理类

面向对象的通讯录系统

程序设计思路图

如上图所示，系统通过管理类来使用各项功能，同时学生类和通讯录类定义不同的信息，并通过类的继承方式结合。

1. 程序结构

1.类设计

在此系统中，我设计了三个类，分别是管理类，学生类以及通讯录类

1. 管理类

管理类中主要是定义和实现系统的各个功能

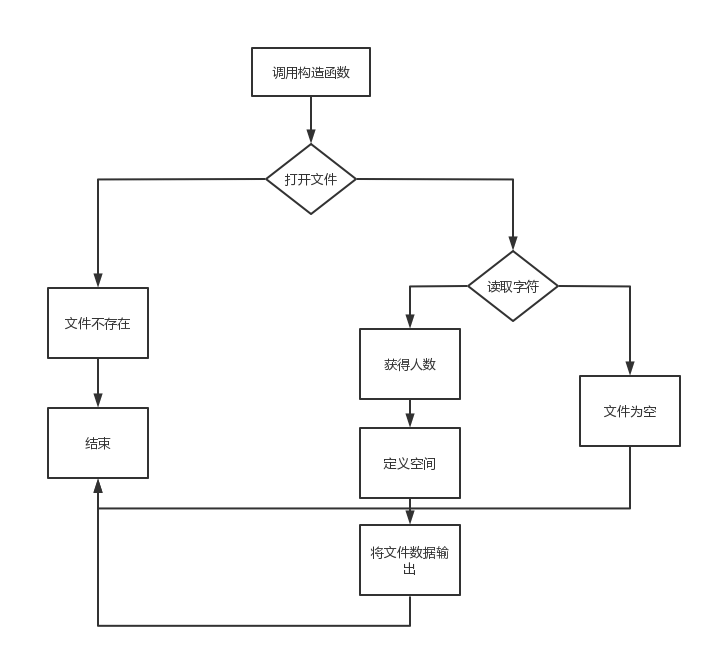
首先是头文件（BookManage.h)中的声明

1. #pragma once
2. #include<iostream>
3. #include<fstream>
4. #include"address.h"
5. #include"student.h"
6. **using** **namespace** std;
8. **class** BookManager    //通讯录管理类
9. {
10. **public**:
12. BookManager();   //构造函数
14. **void** show\_Menu();  //展示菜单
16. **void** Add\_Stu();   //添加学生信息
18. **void** Save();     //将信息保存进文件
20. **void** Exit\_System();  //退出系统
22. **int** number;    //记录人数
24. **bool** Is\_empty;   //判读文件是否为空
26. **void** init\_Stu();   //将文件信息输出
28. **void** show\_Stu();   //展示信息
30. **void** del\_Stu();    //删除学生信息
32. **void** mod\_Stu();   //修改学生信息
34. **void** find\_Stu();  //查找学生信息
36. **int** Is\_Exist(string name);  //判断某人是否存在
38. **int** get\_number();   //获取文件中人数
40. Student\*\* m\_StuArray;   //定义数据容器
42. ~BookManager();    //析构函数
44. };

然后在BookManage.cpp文件中实现各个功能

首先是构造函数

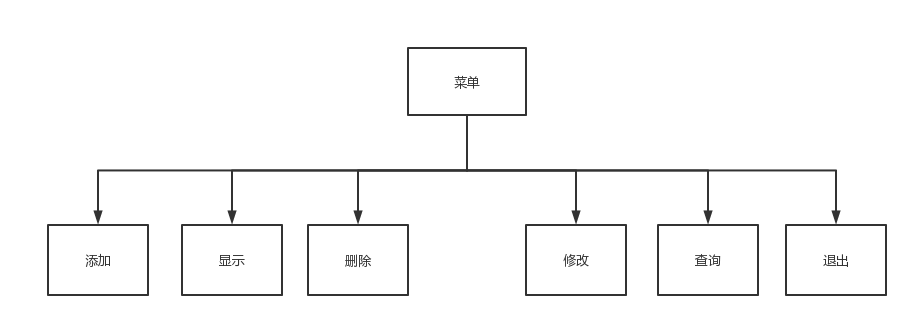
1. #define Filename addresslist.txt    //用宏定义，方便修改文件名
2. BookManager::BookManager()    //构造函数
3. {
4. //初始化文件
5. ifstream fp;
6. fp.open("Filename",ios::in);   //读的方式打开文件
7. **if** (!fp.is\_open())    //当文件不存在
8. {
9. //cout << "文件不存在" << endl;
10. **this**->number = 0;   //初始化人数为0
11. **this**->m\_StuArray = NULL;
12. **this**->Is\_empty = **true**;
13. fp.close();
14. **return**;  //返回
15. }
16. **char** ch;
17. fp >> ch;
18. **if** (fp.eof())   //当文件为空
19. {
20. //cout << "文件为空！" << endl;
21. **this**->number = 0;   //初始化人数为0
22. **this**->m\_StuArray = NULL;
23. **this**->Is\_empty = **true**;
24. fp.close();
25. **return**;   //返回
26. }
28. **int** sum = **this**->get\_number();
29. //cout << "学生人数是：" << sum << "人" << endl;
30. **this**->number = sum;
32. **this**->m\_StuArray = **new** Student \* [sum];
33. **this**->init\_Stu();  //将文件的数据输出
35. }



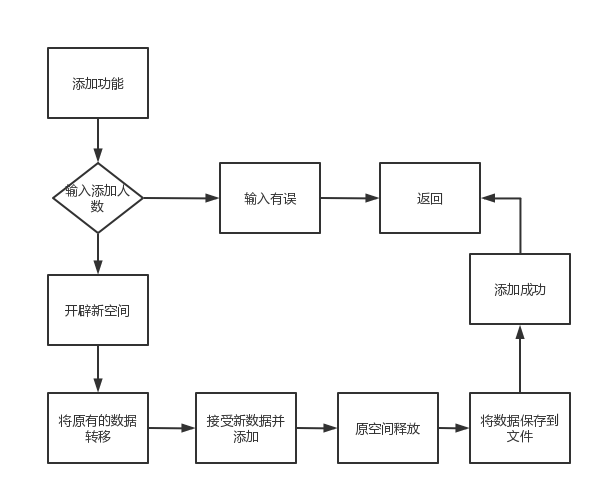
后续是一些获取人数、展示菜单和判断某人是否存在的函数

1. **int** BookManager::get\_number()   //获取人数
2. {
3. ifstream fp;
4. fp.open("Filename", ios::in);  //读文件
5. string Name, College, Class, Tel, Phone,
6. Email, address, Post;
7. **int** num = 0;   //记录数量
8. **while** (fp >> Name && fp >> College && fp >> Class && fp >> Tel && fp >> Phone &&
9. fp >> Email && fp >> address && fp >> Post)    //数据输出
10. {
11. num++;
12. }
13. **return** num;
14. }

17. **void** BookManager::init\_Stu()   //将原来的数据初始化
18. {
19. ifstream fp;
20. fp.open("Filename", ios::in);  //读文件
21. string Name, College, Class, Tel, Phone,
22. Email, address, Post;
23. **int** i = 0;
24. **while** (fp >> Name && fp >> College && fp >> Class && fp >> Tel && fp >> Phone &&
25. fp >> Email && fp >> address && fp >> Post)
26. {
27. Student\* stu = NULL;
28. stu = **new** Student(Name, College, Class, Tel, Phone,
29. Email, address, Post);
30. **this**->m\_StuArray[i] = stu;
31. i++;
32. }
33. fp.close();
34. }
35. **void** BookManager::show\_Menu()           //展示菜单
36. {
37. cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
38. cout << "\*\*\*\*\*\* 欢迎使用同学通讯录系统！\*\*\*\*\*\*" << endl;
39. cout << "\*\*\*\*\*\*\*\*\* 1.添加同学信息  \*\*\*\*\*\*\*\*\*\*\*" << endl;
40. cout << "\*\*\*\*\*\*\*\*\* 2.显示同学信息  \*\*\*\*\*\*\*\*\*\*\*" << endl;
41. cout << "\*\*\*\*\*\*\*\*\* 3.删除同学信息  \*\*\*\*\*\*\*\*\*\*\*" << endl;
42. cout << "\*\*\*\*\*\*\*\*\* 4.修改同学信息  \*\*\*\*\*\*\*\*\*\*\*" << endl;
43. cout << "\*\*\*\*\*\*\*\*\* 5.查询同学信息  \*\*\*\*\*\*\*\*\*\*\*" << endl;
44. cout << "\*\*\*\*\*\*\*\*\* 6.退出通讯系统  \*\*\*\*\*\*\*\*\*\*\*" << endl;
45. cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
46. cout << endl;
47. }
49. **int** BookManager::Is\_Exist(string name)   //判断某人是否存在
50. {
51. **int** flag = -1;
53. **for** (**int** i = 0; i < **this**->number; i++)
54. {
55. **if** (**this**->m\_StuArray[i]->m\_Name == name)
56. {
57. flag = i;                //将位置保存
58. **break**;
59. }
60. }
61. **return** flag;                 //返回位置
62. }

接下来是添加信息的实现，对于用数组指针应用参考[4]。

1. **void** BookManager::Add\_Stu()    //添加功能
2. {
3. cout << "请输入添加的学生数：" << endl;
4. **int** sum = 0;
5. cin >> sum;
6. **if** (sum > 0)
7. {
8. **int** newNum = sum + number;    //计算加入后的人数
9. Student\*\* newSpace = **new** Student \* [newNum];  //开辟新的空间  参考代码
10. **if** (**this**->m\_StuArray != NULL)
11. {
12. **for** (**int** i = 0; i < **this**->number; i++)
13. {
14. newSpace[i] = **this**->m\_StuArray[i];
15. }
16. }
17. //添加新数据
18. **for** (**int** i = 0; i < sum; i++)
19. {
20. string Name, College, Class, Tel, Phone,
21. Email, address, Post;    //定义相关数据
22. cout << "请按顺序输入第" << i + 1
23. << "个同学的姓名、院系、班级、手机、家庭电话、电子邮件、家庭地址、邮编：" << endl;
24. cin >> Name >> College >> Class >> Tel >> Phone >> Email >> address >> Post;  //输入相关数据
26. Student\* add = NULL;
27. add = **new** Student(Name, College, Class, Tel, Phone, Email, address, Post);
28. newSpace[**this**->number + i] = add;   //在原有的基础上继续添加
30. }
31. **delete**[] **this**->m\_StuArray;    //释放原有空间
32. **this**->Is\_empty = **false**;      //文件不为空
33. **this**->m\_StuArray = newSpace;
34. **this**->number = newNum;         //更新学生人数
35. cout << "添加成功！" << endl;
36. **this**->Save();      //将数据导入文件
37. }
38. **else**
39. {
40. cout << "输入有误！" << endl;
41. }
42. //按任意键清屏回到上级目录
43. system("pause");
44. system("cls");
45. }



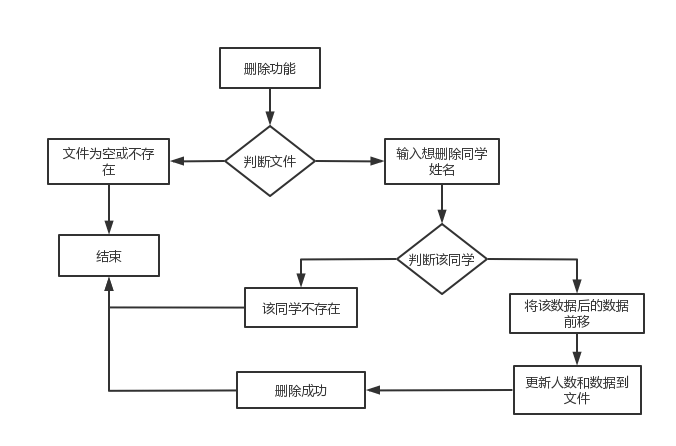
显示功能

当判断文件不为空时，将文件内容输出

1. **void** BookManager::show\_Stu()
2. {
3. **if** (**this**->Is\_empty)
4. cout << "文件为空或不存在！" << endl;
5. **else**
6. {
7. **for** (**int** i = 0; i < number; i++)
8. {
9. **this**->m\_StuArray[i]->showStudent();
10. **this**->m\_StuArray[i]->showAddress();
11. }
12. }
13. system("pause");
14. system("cls");
15. }

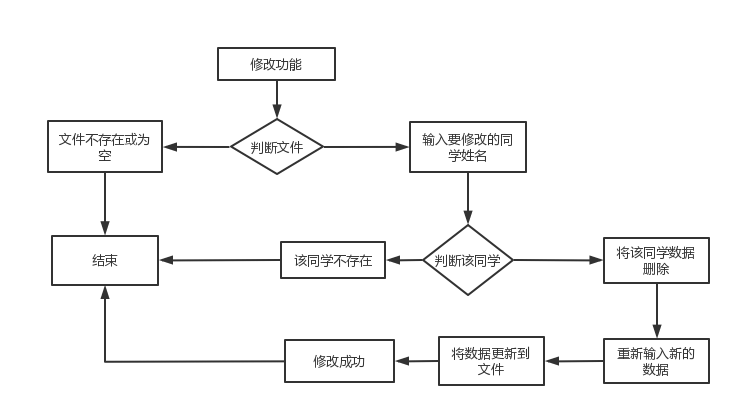
删除功能，对于重名现象进一步考虑院系

1. **void** BookManager::del\_Stu()
2. {
3. **if** (**this**->Is\_empty) cout << "文件不存在或为空！" << endl;
4. **else**
5. {
6. cout << "请输入你想删除的名字：" << endl;
7. string name;
8. **int** a[10];
9. **int** num = 0;
10. cin >> name;
11. **for** (**int** i = 0; i < number; i++)
12. {
13. **if** (**this**->m\_StuArray[i]->m\_Name == name)
14. {
15. cout << "你想删除，" << name << "的信息如下：" << endl;
16. **this**->m\_StuArray[i]->showStudent();
17. **this**->m\_StuArray[i]->showAddress();
18. a[num] = i;
19. num++;
20. }
21. }
23. **if** (num != 0)   //如果该同学存在
24. {
25. **if** (num > 1)
26. {
27. cout << "请输入你想删除人的院系：" << endl;
28. string col;
29. cin >> col;
30. **for** (**int** i = 0; i < num; i++)
31. {
32. **if** (**this**->m\_StuArray[a[i]]->m\_College == col)
33. {
34. **for** (**int** j = a[i]; j < **this**->number - 1; j++)
35. {
36. **this**->m\_StuArray[j] = **this**->m\_StuArray[j + 1];  //将此数据后的数据前移
37. }
38. **this**->number--;        //更新人数
39. **this**->Save();        //将删除后的信息更新到文件中
40. cout << "删除成功！" << endl;
41. system("pause");
42. system("cls");
43. }
44. }
46. }
47. **else** **if** (num == 1)
48. {
49. **for** (**int** i = a[0]; i < **this**->number - 1; i++)
50. {
51. **this**->m\_StuArray[i] = **this**->m\_StuArray[i + 1];  //将此数据后的数据前移
52. }
53. **this**->number--;        //更新人数
54. **this**->Save();        //将删除后的信息更新到文件中
55. cout << "删除成功！" << endl;
56. system("pause");
57. system("cls");
58. }
59. }
60. **else** cout << "未找到该同学！" << endl;
61. }
62. }



修改功能，若发现重名进一步考虑院系

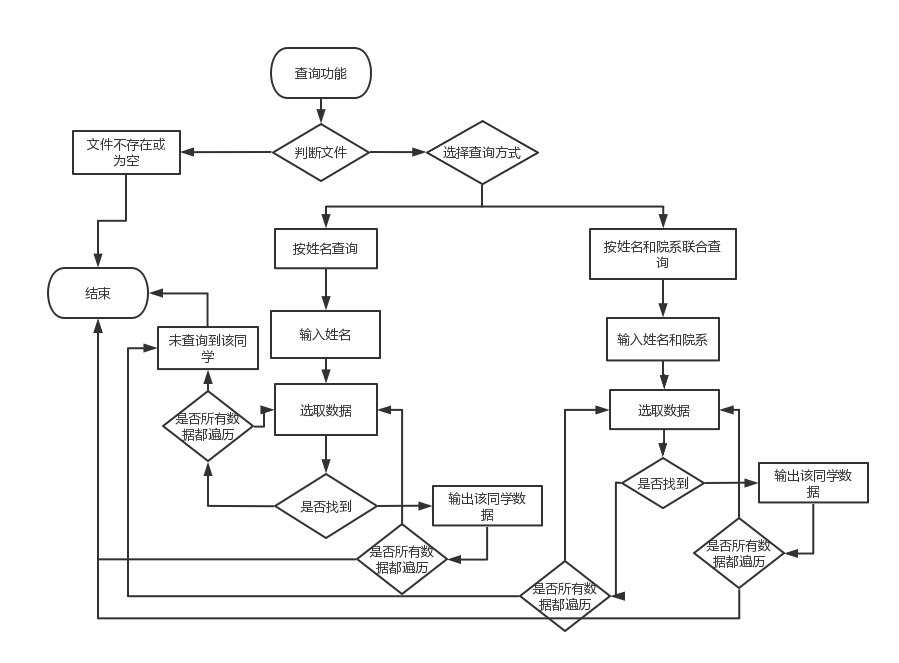
1. **void** BookManager::mod\_Stu()
2. {
3. **if** (**this**->Is\_empty)   //文件为空
4. {
5. cout << "文件不存在或为空！" << endl;
6. }
7. **else**
8. {
9. cout << "请输入要修改的同学姓名：" << endl;
10. string name;
11. cin >> name;
12. **int** num = 0;
13. **int** a[10];
14. **for** (**int** i = 0; i < number; i++)
15. {
16. **if** (**this**->m\_StuArray[i]->m\_Name == name)
17. {
18. cout << "你想修改，" << name << "的信息如下：" << endl;
19. **this**->m\_StuArray[i]->showStudent();
20. **this**->m\_StuArray[i]->showAddress();
21. a[num] = i;
22. num++;
23. }
24. }
25. **if** (num != 0)   //如果该同学存在
26. {
27. **if** (num > 1)
28. {
29. cout << "请输入你想修改人的院系：" << endl;
30. string col;
31. cin >> col;
32. **for** (**int** i = 0; i < num; i++)
33. {
34. **if** (**this**->m\_StuArray[a[i]]->m\_College == col)
35. {
36. **delete** **this**->m\_StuArray[a[i]];    //将要修改的人先删除
37. string College, Class, Tel, Phone,
38. Email, address, Post;    //定义相关数据
39. cout << "请按顺序输入" << name
40. << "同学的新的院系、班级、手机、家庭电话、电子邮件、家庭地址、邮编：" << endl;
41. cin >> College >> Class >> Tel >> Phone >> Email >> address >> Post;  //输入相关数据
43. Student\* stu = NULL;
44. stu = **new** Student(name, College, Class, Tel, Phone,
45. Email, address, Post);
46. **this**->m\_StuArray[a[i]] = stu;
47. cout << "修改成功！" << endl;
48. **this**->Save();      //保存到文件中
49. }
50. }
52. }
53. **else** **if** (num == 1)
54. {
55. **delete** **this**->m\_StuArray[a[0]];    //将要修改的人先删除
56. string College, Class, Tel, Phone,
57. Email, address, Post;    //定义相关数据
58. cout << "请按顺序输入" << name
59. << "同学的新的院系、班级、手机、家庭电话、电子邮件、家庭地址、邮编：" << endl;
60. cin >> College >> Class >> Tel >> Phone >> Email >> address >> Post;  //输入相关数据
62. Student\* stu = NULL;
63. stu = **new** Student(name, College, Class, Tel, Phone,
64. Email, address, Post);
65. **this**->m\_StuArray[a[0]] = stu;
66. cout << "修改成功！" << endl;
67. **this**->Save();      //保存到文件中
68. }
69. }
70. **else** cout << "未找到该同学！" << endl;
71. }
72. system("pause");
73. system("cls");
74. }



查询功能

按姓名或按姓名和院系查询

1. **void** BookManager::find\_Stu()
2. {
3. **if** (**this**->Is\_empty)   //文件为空
4. {
5. cout << "文件不存在或为空！" << endl;
6. }
7. **else**
8. {
9. cout << "请选择查找方式：" << endl;
10. cout << "1.按照姓名查询" << endl;
11. cout << "2.按照姓名和学院查询" << endl;
12. **int** choice = 0;
13. cin >> choice;
14. **if** (choice == 1)     //单一字段查询
15. {
16. string name;
17. **int** flag = 0;
18. cout << "请输入要查询的姓名：" << endl;
19. cin >> name;
20. **for** (**int** i = 0; i < number; i++)
21. {
22. **if** (**this**->m\_StuArray[i]->m\_Name == name)
23. {
24. cout << "查找成功，" << name << "的信息如下：" << endl;
25. **this**->m\_StuArray[i]->showStudent();
26. **this**->m\_StuArray[i]->showAddress();
27. flag = 1;
28. }
29. }
30. **if** (flag == 0)
31. cout << "未查询到此人的信息！" << endl;
32. }
33. **else** **if** (choice == 2)    //多字段联合查询
34. {
35. string name,college;
36. **int** flag = 0;
37. cout << "请输入要查询的姓名和院系：" << endl;
38. cin >> name>>college;
39. **for** (**int** i = 0; i < number; i++)
40. {
41. **if** (**this**->m\_StuArray[i]->m\_Name == name&&**this**->m\_StuArray[i]->m\_College==college)
42. {
43. cout << "查找成功，" << name << "的信息如下：" << endl;
44. **this**->m\_StuArray[i]->showStudent();
45. **this**->m\_StuArray[i]->showAddress();
46. flag = 1;
47. }
48. }
49. **if** (flag == 0)
50. cout << "未查询到此人的信息！" << endl;
51. }
52. **else** cout << "输入有误！" << endl;
53. }
54. system("pause");
55. system("cls");
56. }



退出功能

安全退出系统。

1. **void** BookManager::Exit\_System()  //退出系统
2. {
3. cout << "欢迎下次使用！" << endl;
4. system("pause");
5. exit(0);
6. }

最后是一个将数据保存至文件和析构函数

1. **void** BookManager::Save()
2. {
3. ofstream out;
4. out.open("Filename", ios::out);   //用输出方式打开文件
5. **for** (**int** i = 0; i < **this**->number; i++)   //将数据写入文件
6. {
7. out << **this**->m\_StuArray[i]->m\_Name << " "
8. << **this**->m\_StuArray[i]->m\_College << " "
9. << **this**->m\_StuArray[i]->m\_Class << " "
10. << **this**->m\_StuArray[i]->m\_Tel << " "
11. << **this**->m\_StuArray[i]->m\_Phone << " "
12. << **this**->m\_StuArray[i]->m\_Email << " "
13. << **this**->m\_StuArray[i]->m\_Address << " "
14. << **this**->m\_StuArray[i]->m\_Post << " " << endl;
15. }
16. out.close();
17. }
19. BookManager::~BookManager()   //析构函数
20. {
21. **if** (**this**->m\_StuArray != NULL)
22. {
23. **delete**[] **this**->m\_StuArray;        //用new开辟的堆区数据delete释放
24. **this**->m\_StuArray = NULL;
25. }
26. }
27. 通讯录类头文件

在通讯录类中定义相关变量和虚函数

1. #pragma once
2. #include<iostream>
3. #include<string>
4. **using** **namespace** std;
6. //创建通讯录抽象类
7. **class** Address
8. {
9. **public**:
10. string m\_Tel;   //手机号
11. string m\_Phone;  //家庭电话
12. string m\_Email;   //电子邮件
13. string m\_Address;  //家庭地址
14. string m\_Post;    //邮编
16. //显示通讯信息
17. **virtual** **void** showAddress() = 0;    //抽象类
18. //显示学生信息
19. **virtual** **void** showStudent() = 0;    //抽象类
21. };
22. 学生类

学生类继承通讯录类

student.h文件：

1. #pragma once
2. #include<iostream>
3. #include"address.h"
4. **using** **namespace** std;
6. //创建学生类
7. **class** Student :**public** Address   //继承通讯录类
8. {
9. **public**:
10. Student(string Name, string College, string Class, string Tel,string Phone,
11. string Email, string Address, string Post );   //构造函数
13. //显示通讯信息
14. **virtual** **void** showAddress() ;    //抽象类
15. //显示学生信息
16. **virtual** **void** showStudent() ;    //抽象类
18. string m\_Name;     //姓名
19. string m\_College;   //院系
20. string m\_Class;    //班级
21. ~Student();
22. };

student.cpp文件，进行赋值和输出功能。

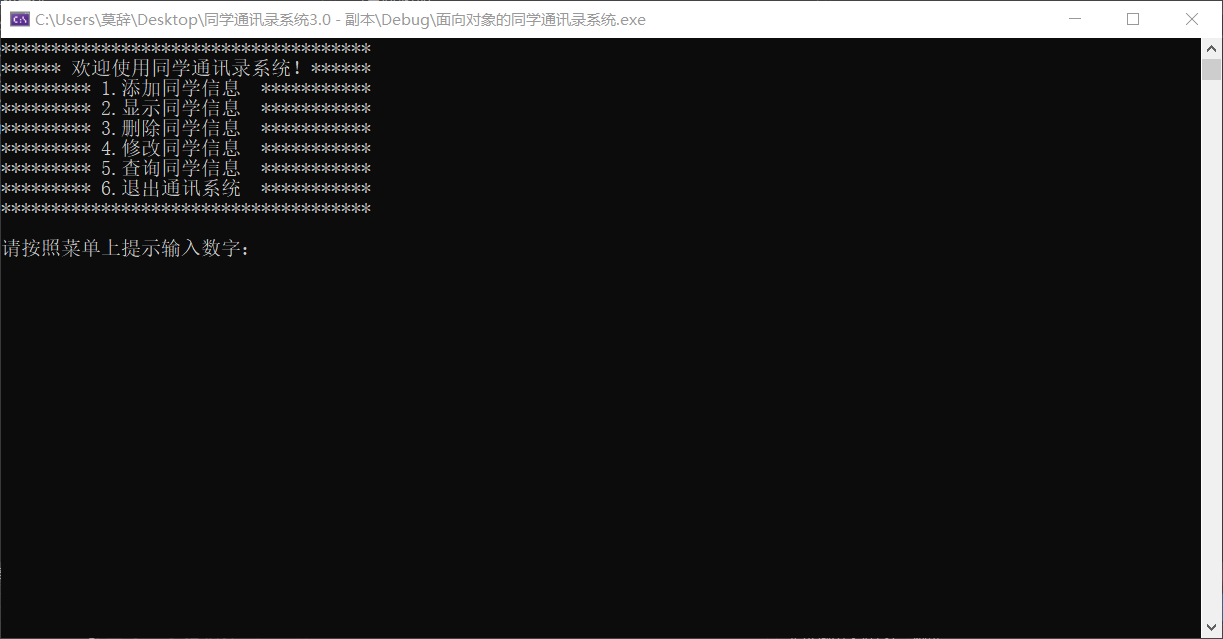
1. #include"student.h"
3. Student::Student(string Name, string College, string Class, string Tel, string Phone,
4. string Email, string Address, string Post)   //构造函数
5. {
6. m\_Name = Name;
7. m\_College = College;
8. m\_Class = Class;
9. m\_Tel = Tel;
10. m\_Phone = Phone;
11. m\_Email = Email;
12. m\_Address = Address;
13. m\_Post = Post;
14. }
15. //显示通讯信息
16. **void** Student::showAddress()
17. {
18. cout << "  手机：" << m\_Tel
19. << "  家庭电话：" << m\_Phone
20. << "  电子邮箱：" << m\_Email
21. << "  家庭地址：" << m\_Address
22. << "  邮编：" << m\_Post << endl;
23. }
24. //显示学生信息
25. **void** Student::showStudent()
26. {
27. cout << "姓名：" << m\_Name<< "  院系：" << m\_College<< "  班级：" << m\_Class;
28. }
29. Student::~Student()
30. {
31. }

2.主函数设计

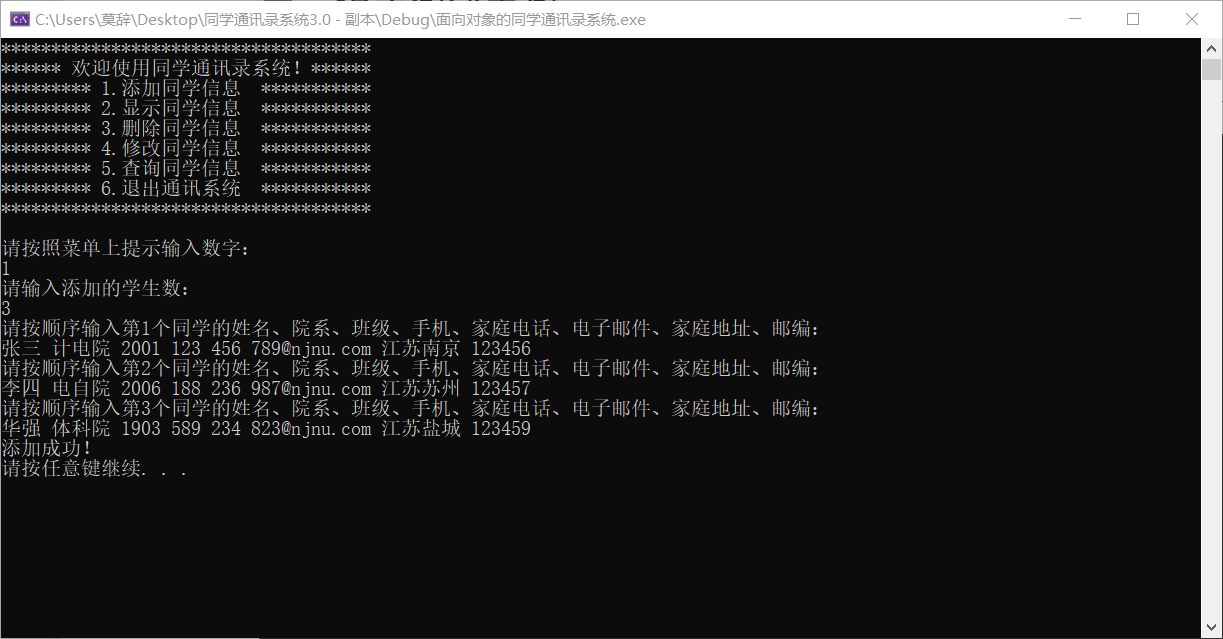
在主函数中，我实例化管理类对象来实现功能选择，不同数字调用不同函数。

1. #include<iostream>
2. #include"bookManager.h"
3. #include"address.h"
4. #include"student.h"
5. **using** **namespace** std;
7. **int** main()
8. {
9. BookManager bm;           //实例化管理对象
10. **int** choice = 0;            //用户的功能选择
12. **while** (**true**)
13. {
14. bm.show\_Menu();          //调用菜单
15. cout << "请按照菜单上提示输入数字：" << endl;
16. cin >> choice;    //输入你想使用的功能所代表的数字
17. **switch** (choice)
18. {
19. **case** 1:       //添加信息
20. bm.Add\_Stu();
21. **break**;
22. **case** 2:       //显示信息
23. bm.show\_Stu();
24. **break**;
25. **case** 3:       //删除信息
26. bm.del\_Stu();
27. **break**;
28. **case** 4:       //修改信息
29. bm.mod\_Stu();
30. **break**;
31. **case** 5:       //查询信息
32. bm.find\_Stu();
33. **break**;
34. **case** 6:       //退出系统
35. bm.Exit\_System();     //调用退出系统函数
36. **break**;
37. **default**:
38. system("cls");    //输入其他数字清屏
39. **break**;
40. }
41. }
43. system("pause");
44. **return** 0;
45. }
46. 程序测试案例

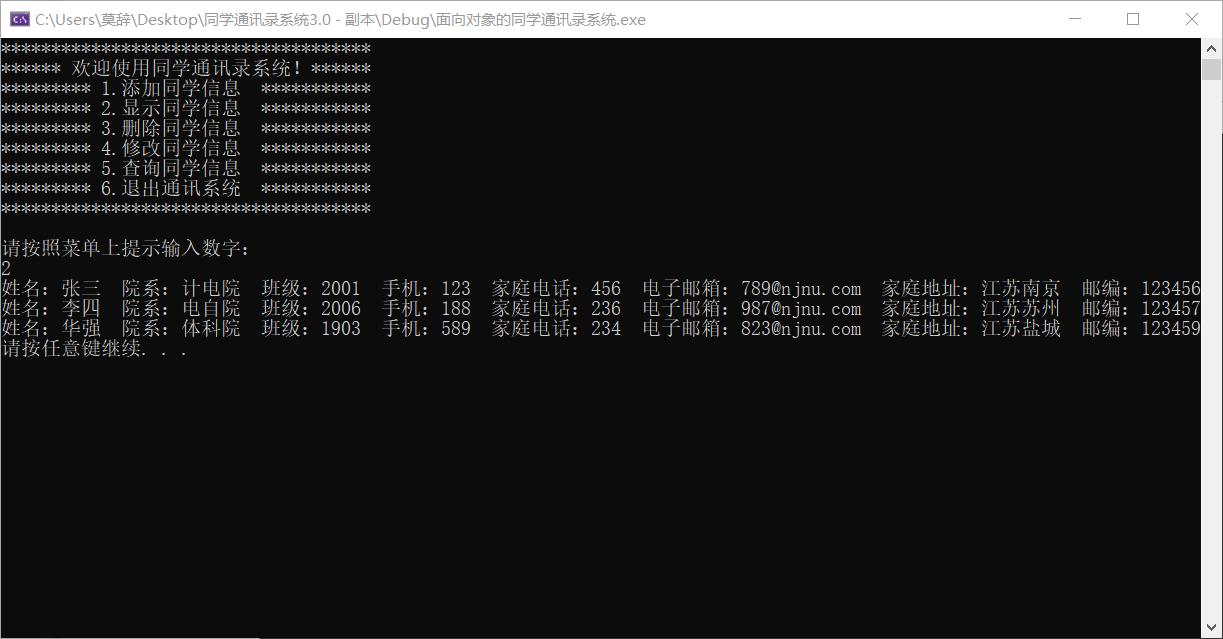
首先，我们在Visual Studio上运行程序，弹出如下页面



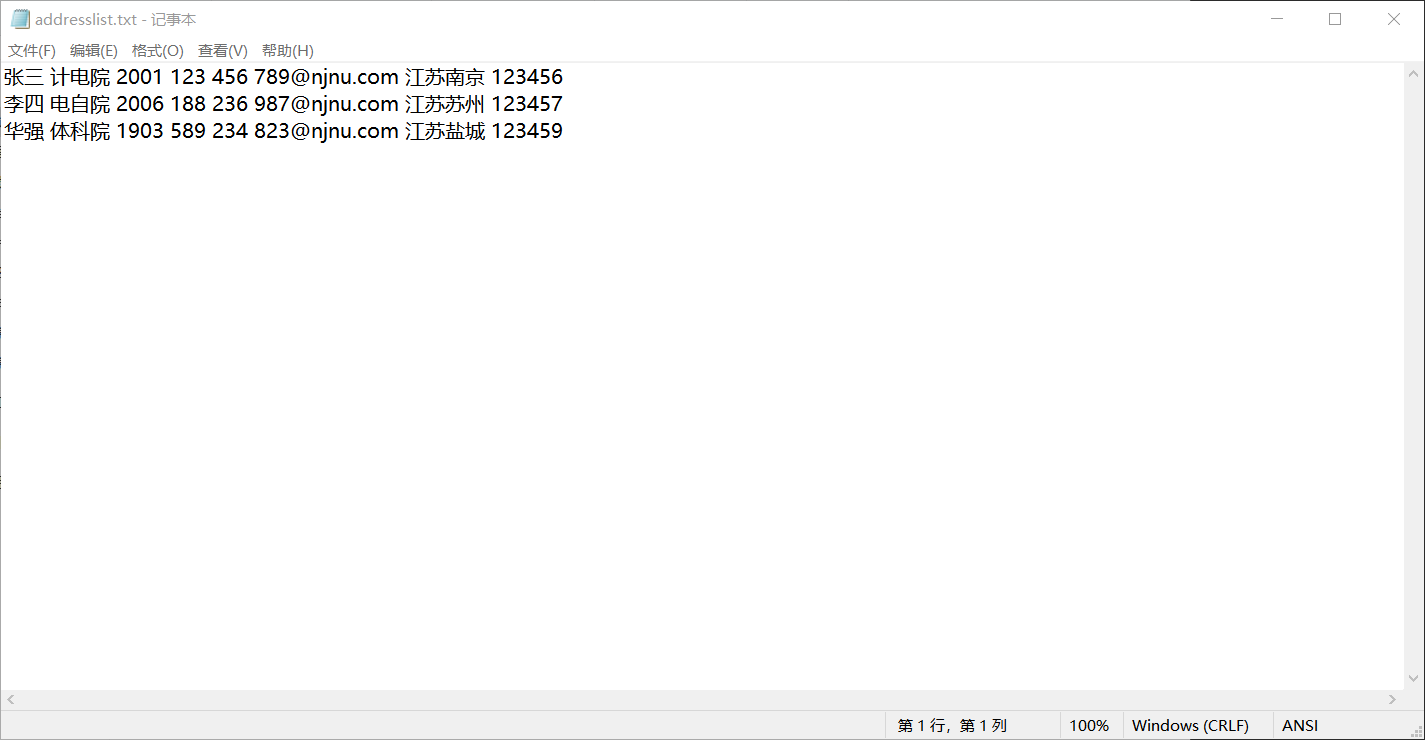
我们输入1，选择添加信息，并按照提示输入信息，得到添加成功页面。



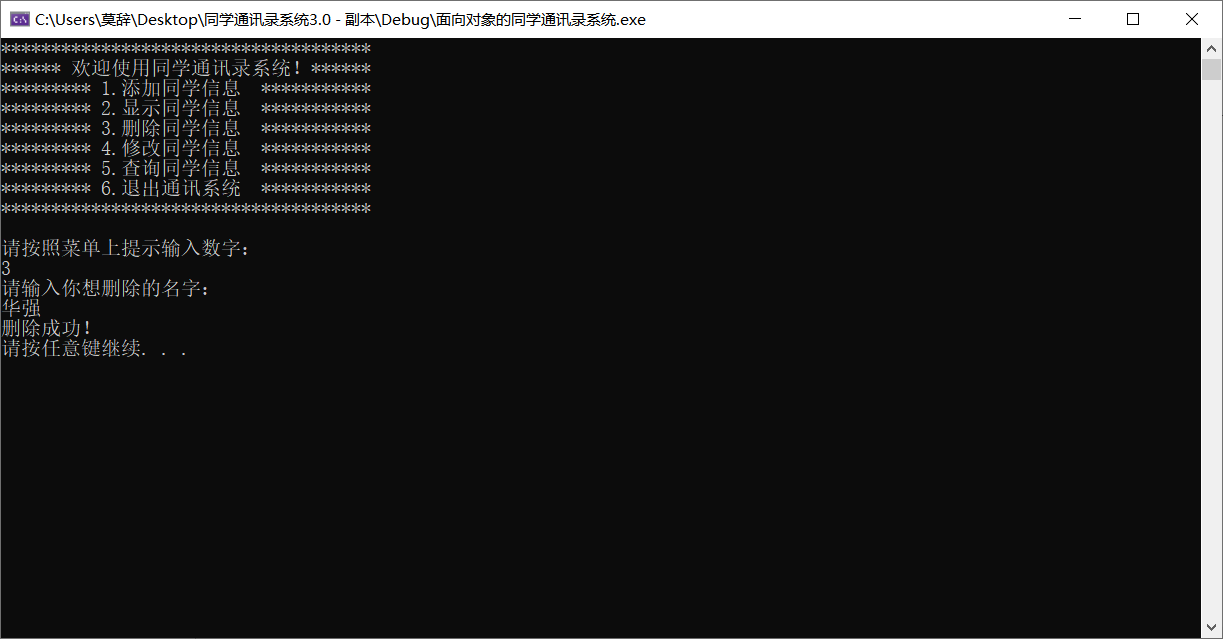
按任意键后回到上级目录，我们输入2，选显示功能，得到我们输入的信息。

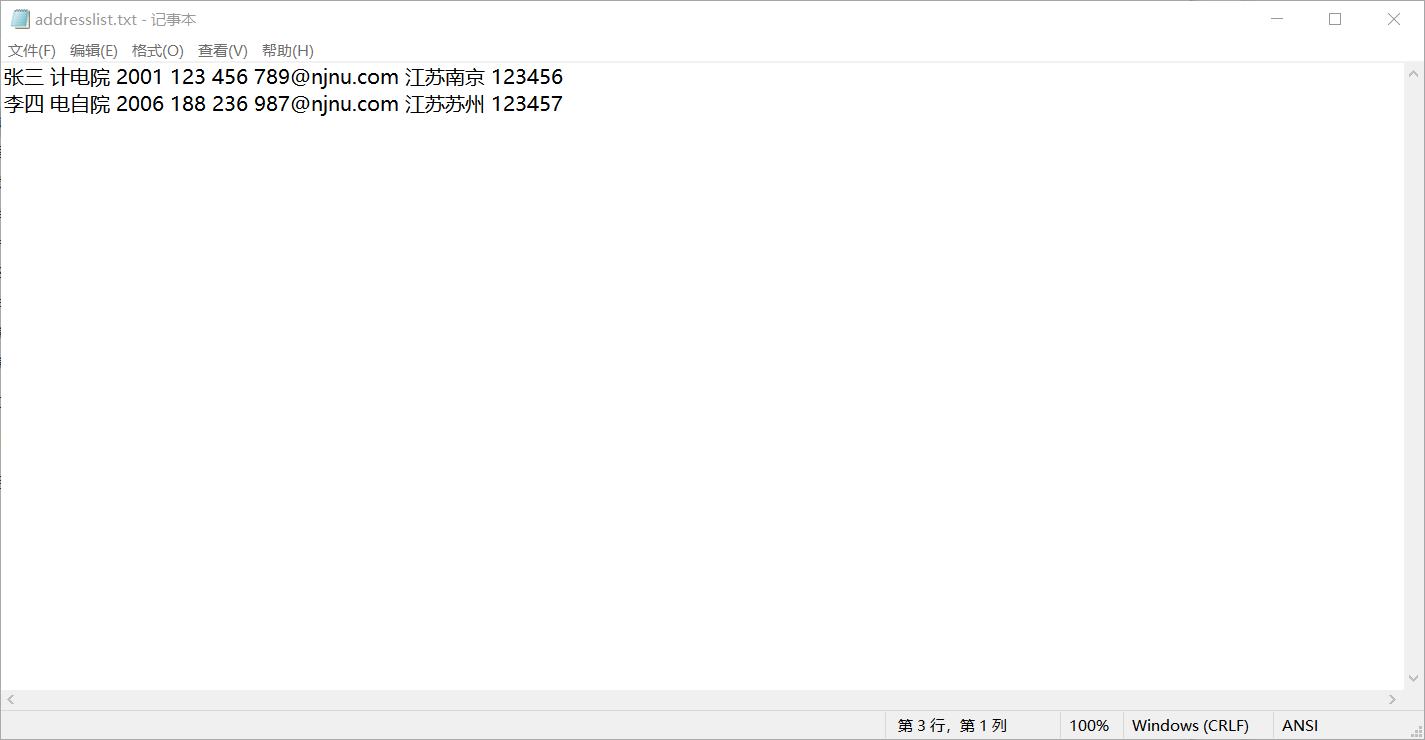


我们打开文件夹，发现一个名为"addresslist.txt"的文档，打开就是我们所输入的信息。



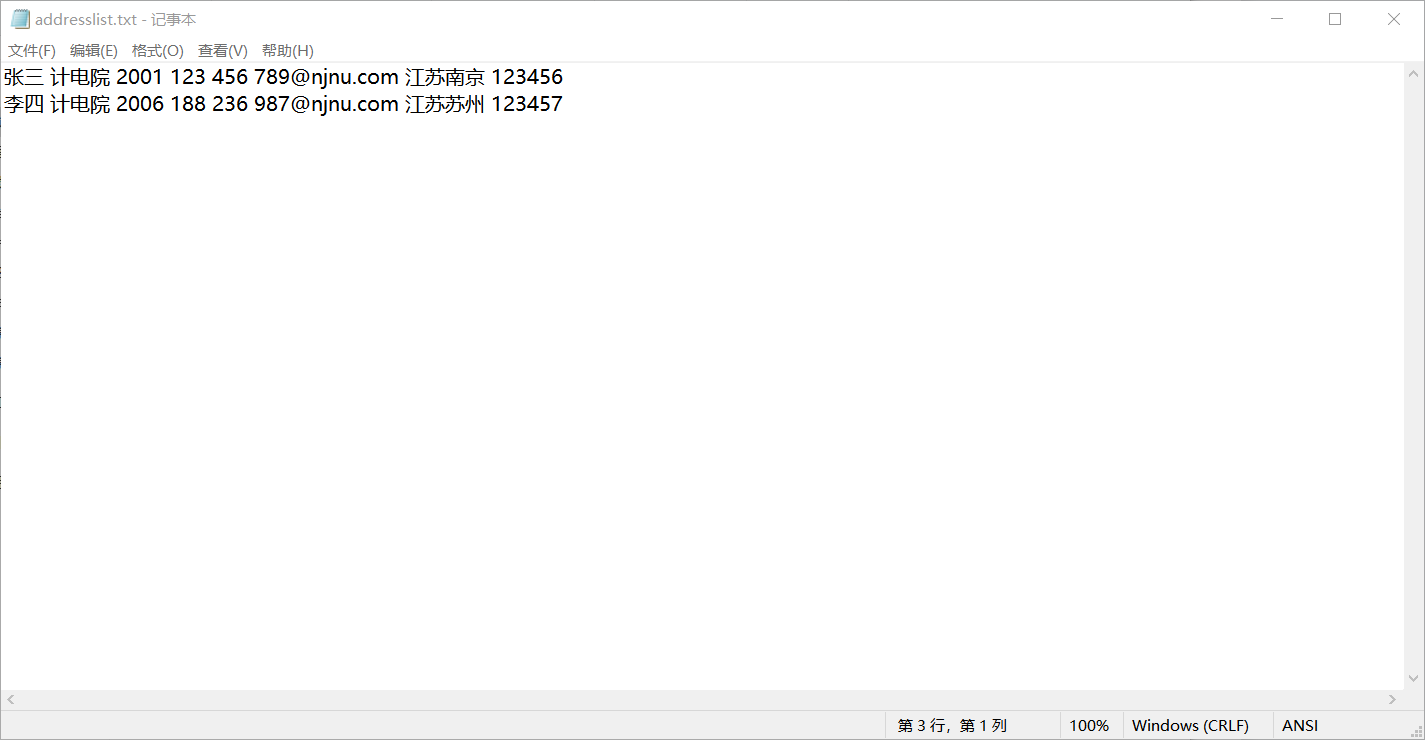
我们选择3，删除信息功能，按提示输入，得到删除成功反馈后，继续打开addresslist.txt，发现相关信息已删除。



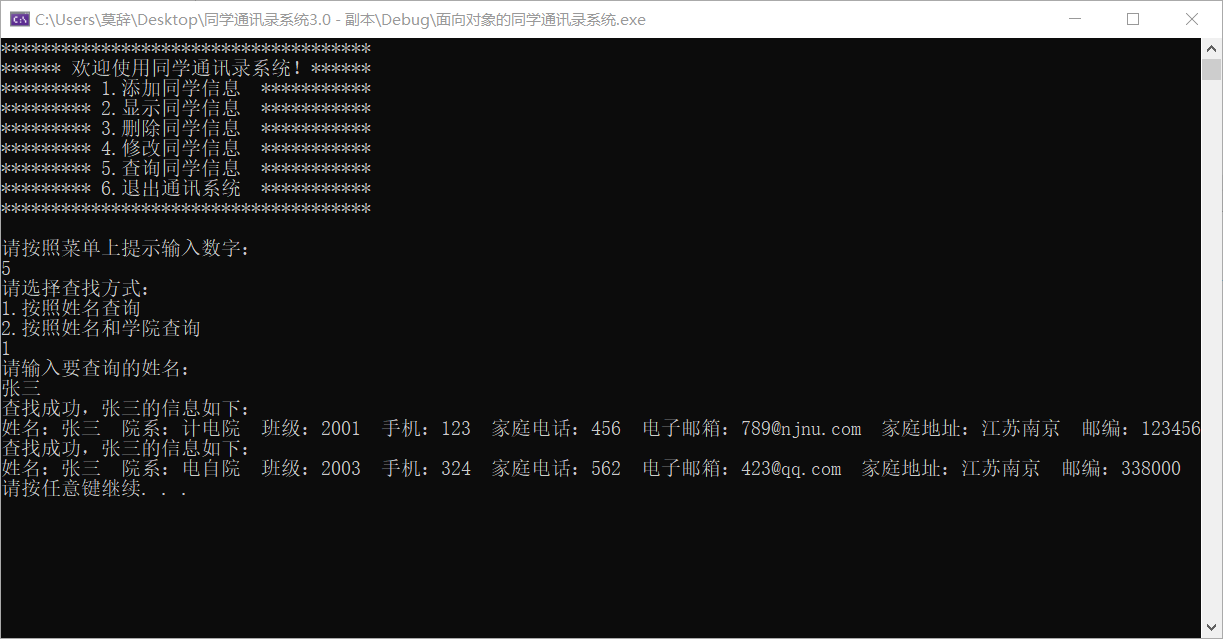


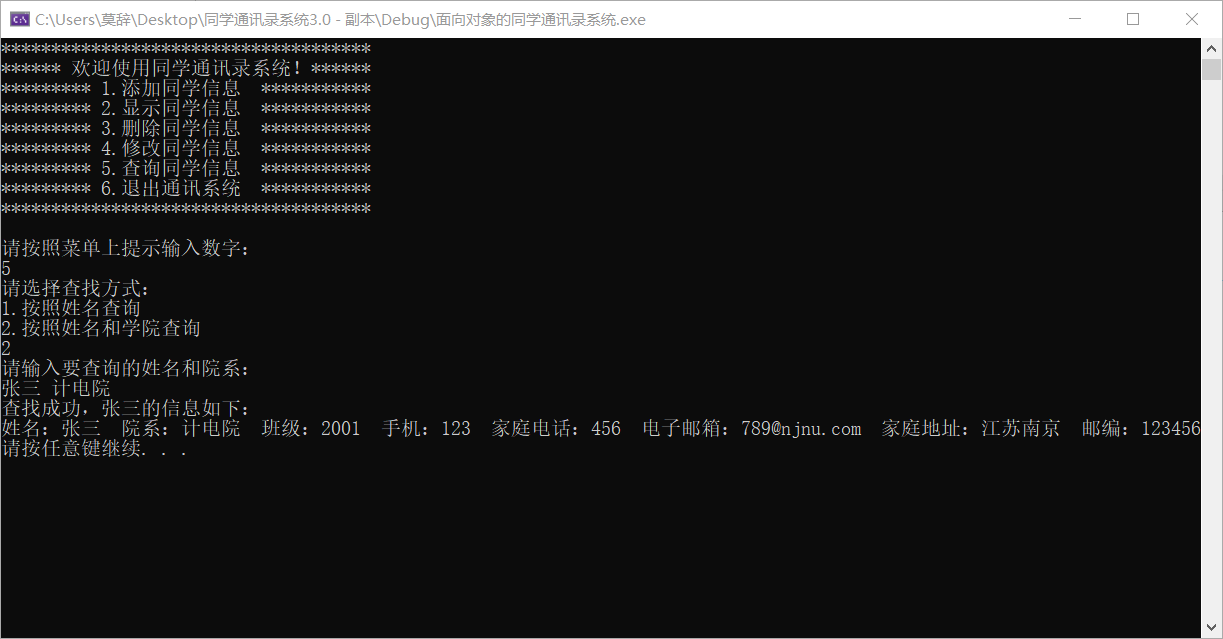
之后，我们选择4，修改信息，将李四的院系改为计电院，文件中自动同步数据。





我们再添加一个叫张三的人的信息，但院系是电自院，我们选择5，查询功能，进行两次查询，第一次只考虑姓名，得到两条信息，第二次多字段联合查询，得到唯一信息。





最后，我们选择6，退出系统。

1. 收获与体会

开始做这个项目的时候真的是晕头转向，经常是拆东墙补西墙，忘记变量名和函数名，写到后面才发现前面有些东西没考虑进去，这时候再往回看又有点看不懂自己的代码了。并且随着我一边又一边调试，最后程序崩溃，后来发现内存没清理干净。做之前看过一些其他人类似功能的代码，发现他们大多数是在一个cpp文件实现所有功能，但这样少了些封装的味道，我还是选择多文件项目，把每一个部分单独拎出来，先画思维导图，之后一一去实现，尽量取有意义的变量名和函数名，并且多写注释，同时也庆幸自己用的是Visual Studio，它能在你出错的时候第一时间告诉你哪里有问题。对于重名的问题，最好的解决办法是引入学号，在此程序中是进一步考虑院系情况。

整个源代码经过三个版本的迭代，第一个版本只是简单的实现各个功能，使用页面和提示信息并不友好，整体感觉非常粗糙。第二个版本对于文件的交互进行了更新，对于各个功能增加了其他情况的考虑和反馈，比如输入数字有误提示，文件为空或不存在提示，功能实现是否成功提示，增加各个功能完成后返回上级目录和清理屏幕功能，增加文件名的宏定义。第三个版本加入多态和抽象类的语法，手动去释放一些内存，考虑删除、修改、查询发现重名情况，优化算法。虽然程序能正常运行，但这些代码基本是只运用我上学期所学的东西，少了一些新东西，其实有尝试过加入类的模块，但出现了一堆我乱码问题，还是不够了解，本人学识尚浅，程序依然有很多问题和不完善的地方，还希望老师不吝赐教。

通过这次的程序设计，我明白只有在实践中才能锻炼自己的动手能力和独立思考能力，在设计的过程中发现自己的不足，对知识的把握还有欠缺，我所掌握的只是这个体系内的冰山一角。我也体会到C++的封装性，各种功能相对独立，还有巧妙的函数可以快速完成复杂的功能。

1. 参考文献

[1]陈浩杰.Visual C++程序设计课程阶梯法教学实践[J].电子技术,2021,50(08):44-45.

[2]郑莉，董渊，张瑞丰．C++语言程序设计．3 版．北京：清华大学出版社，2014．30-34

[3]曲维光，姚望舒. C++面向对象程序设计. 北京：科学出版社. 2016

[4]杨进才, 王奇睿. C++数组与指针深入剖析[J]. 微型机与应用, 2000, 19(10):7-9.