通讯录管理系统

设计者：

1910400731-倪路

2020

**通讯录管理系统课设设计说明书目录**

目录

[一、概述 3](#_Toc22896)

[(1) 项目背景： 3](#_Toc20420)

[(2) 编写目的 3](#_Toc14060)

[(3) 软件定义 3](#_Toc24439)

[(4) 开发环境 3](#_Toc22141)

[二、 需求分析 3](#_Toc16230)

[(1) 问题陈述 3](#_Toc31468)

[(2) 功能概述 4](#_Toc1287)

[(3) 总系统功能图 5](#_Toc4705)

[(4) 模块功能说明 6](#_Toc580)

[三、面向对象C++程序设计实施 10](#_Toc23029)

[1、 打开通讯录 10](#_Toc20292)

[2、 添加用户信息 12](#_Toc25311)

[3、 查询用户信息 14](#_Toc21350)

[4、 删除用户信息 18](#_Toc11776)

[5、 修改用户信息 21](#_Toc20244)

[6、 清空用户信息 25](#_Toc30940)

[7、 合并用户信息 26](#_Toc13675)

[四、结束语 28](#_Toc28475)

[五、 参考文献 30](#_Toc23485)

# 一、概述

## (1)项目背景：

当今时代，手机已经成为人们生活中不可缺少的一部分，基本每个人都拥有一部智能手机，这就产生了大量的联系方式，为了能够更好的管理大量的人员的联系方式，一个优秀的通讯录管理系统就显得十分重要了。

## 编写目的

编写此系统的目的是为了将大量用户的联系方式更加方便的存储起来，以便于用户查找其他人的联系方式等信息，以用于用户之间的相互通信与交流。

## 软件定义

本软件为通讯录管理系统，用于管理用户的联系方式，姓名，住址等信息。

## (4)开发环境

VisualStudio2019

# 需求分析

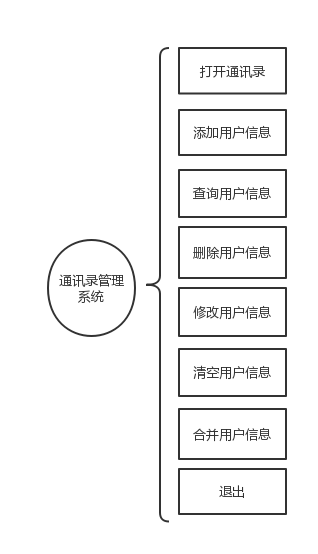
## 问题陈述

设计一个实用的小型通信录程序，具有查询和删除功能，并且能够打开或修改指定文件及将多个文件组成一个文件。它完全使用类来实现，充分体现了面向对象的程序设计特点。 本程序的参考程序使用双向链表保存数据的底层数据结构，并使用C++标准库提供的list来实现这个链表的结构。由于利用list类，所以实现起来很简单。虽然这种方法超出一般的教材，但只要学习一下标准库，还是容易理解的。C++推荐使用标准库，这也是目前编程的方向，推荐此题目为本课程设计作为选做内容，可以根据实际教学情况决定，目的是使学生熟悉和了解C++的标准库。

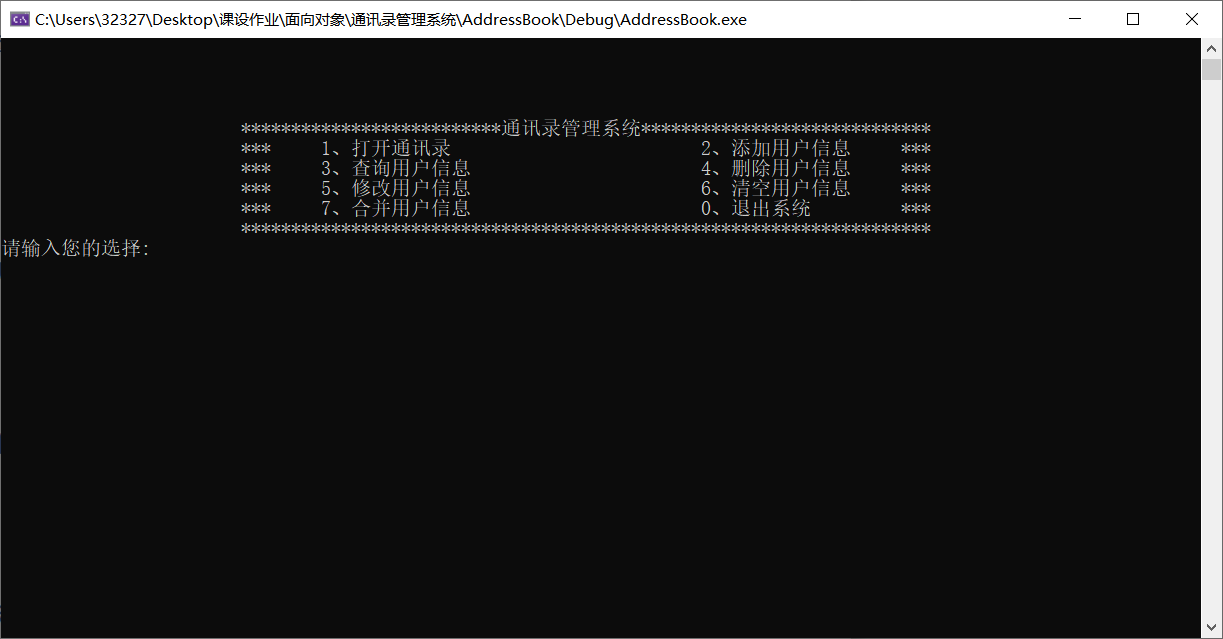
## 功能概述

根据题目要求，本程序的实现的功能有：打开通讯录、添加用户信息、查询用户信息、删除用户信息、修改用户信息、清空用户信息、合并用户信息。

## 总系统功能图

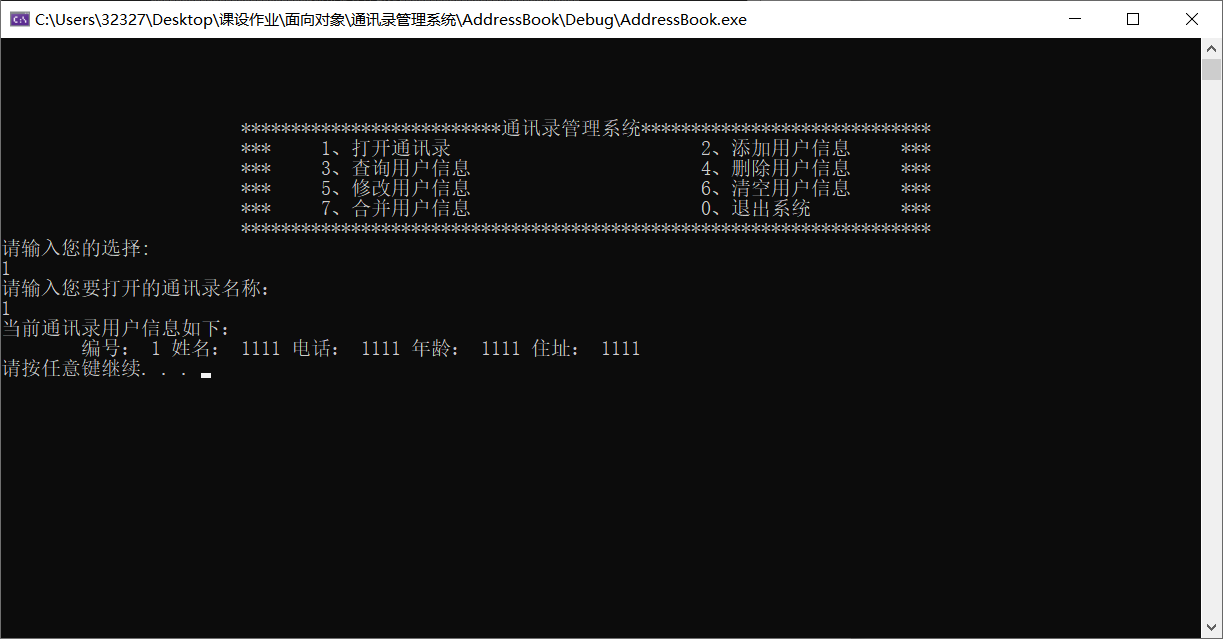


## 模块功能说明



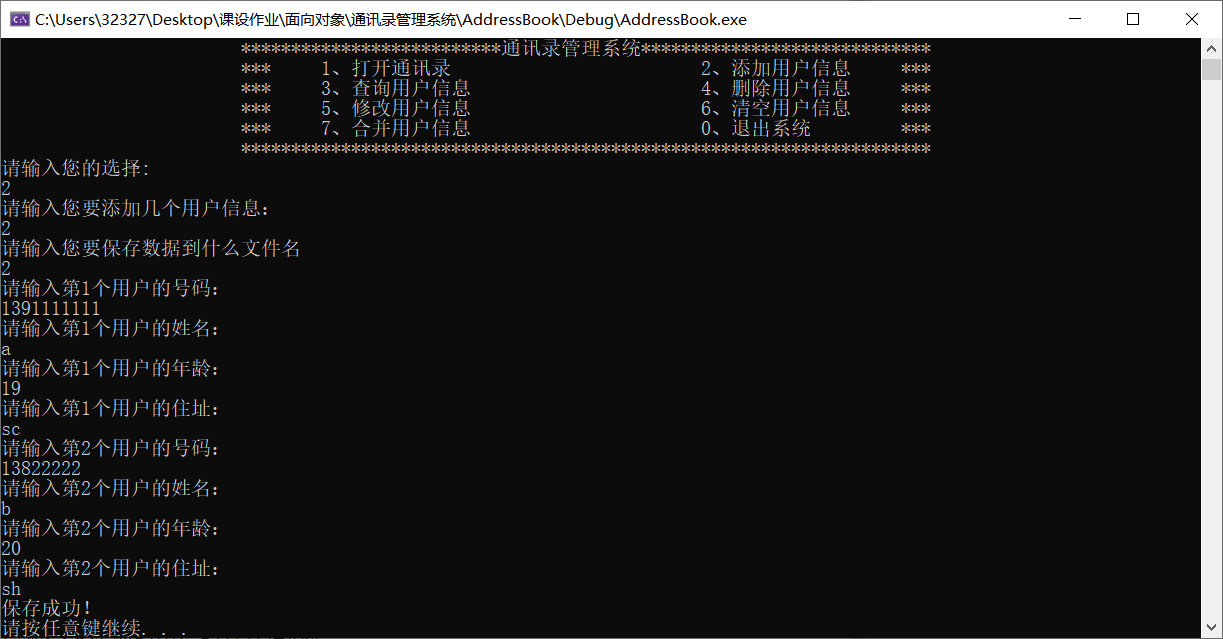
#### 1、打开通讯录

该模块用于打开某个通讯录文件以显示该通讯录中的用户信息。当使用该功能时，程序会要求用户输入一个通讯录名称，输入完毕后程序会在当前目录下进行查询，若查找到该名称的通讯录，程序则会将该通讯录中的用户信息打印至屏幕。



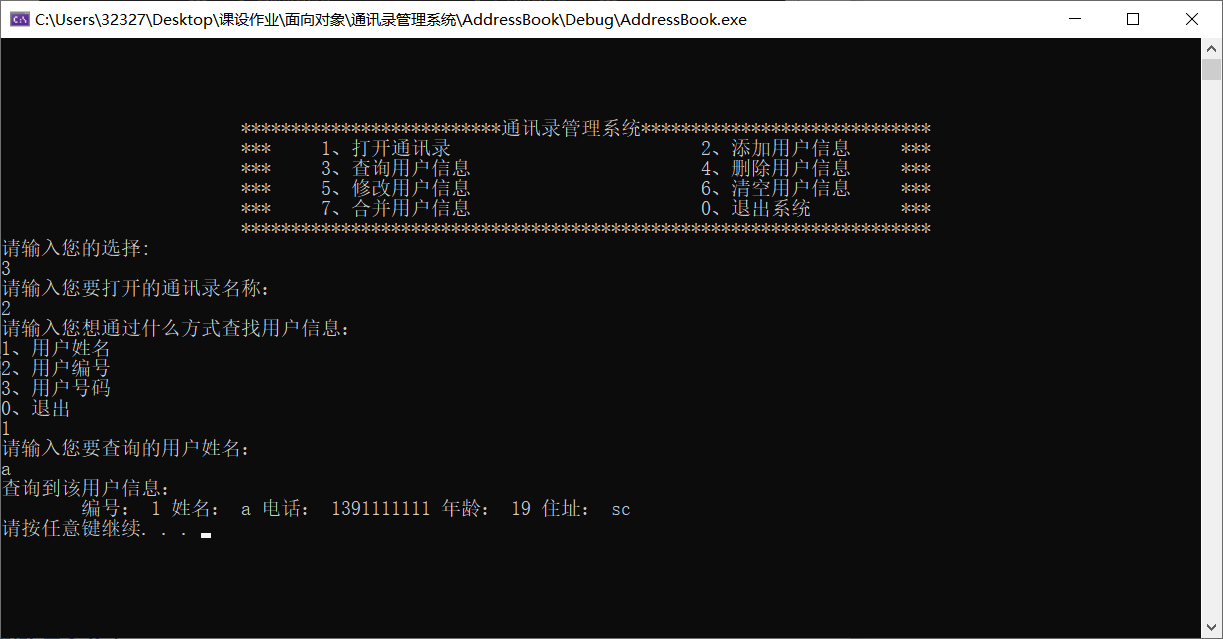
#### 2、添加用户信息

该模块用于将用户的信息添加至通讯录中。当用户使用该功能时，程序会提示用户输入添加几个用户以及添加至什么文件名，随后会让用户输入要添加的用户的各种信息。



#### 3、查询用户信息

该模块用于查询通讯录中的用户信息。当用户使用该功能时，程序会首先提示用户要查询哪个通讯录的用户信息，当用户输入要查询的通讯录后，程序会提示用户选择查询信息的方式，紧接着进行查找。



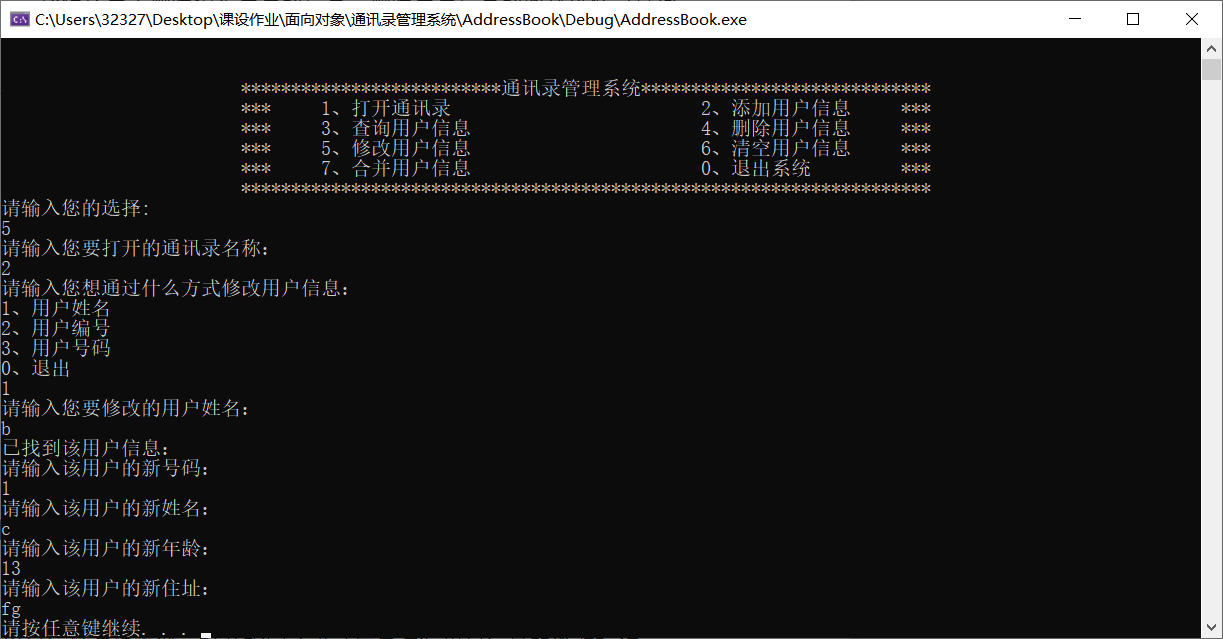
#### 4、删除用户信息

该模块用于删除指定用户的信息。删除用户信息的函数过程与查询用户信息过程类似，打开指定通讯录后再进行用户信息的删除操作。



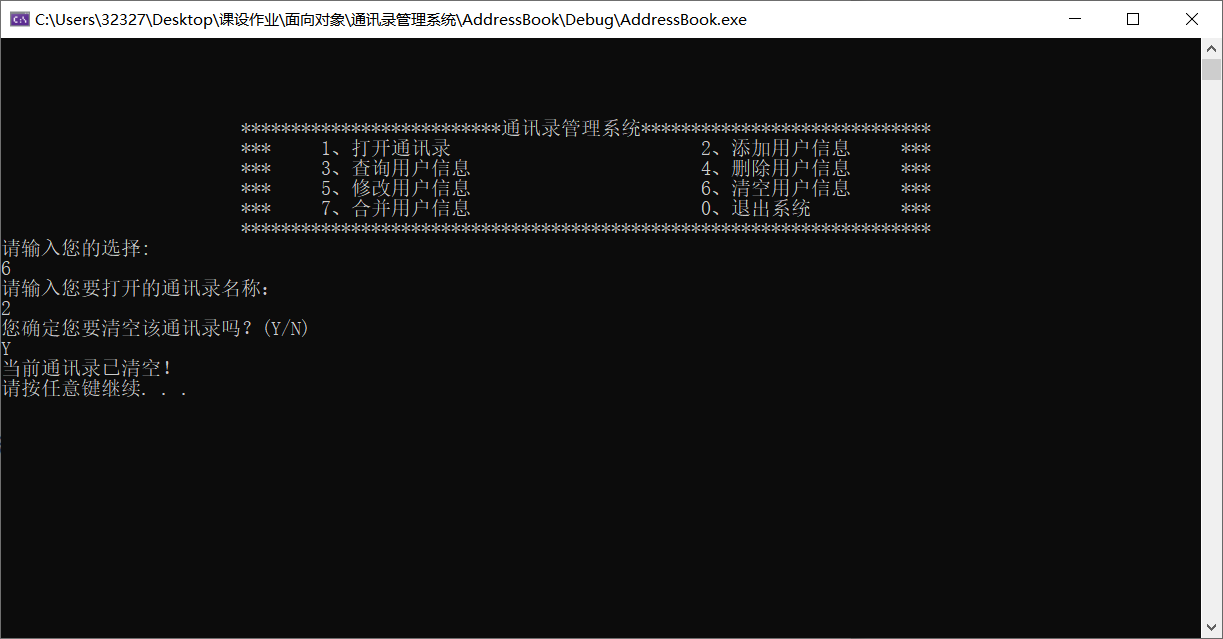
#### 5、修改用户信息

该模块用于修改指定用户的信息。修改用户信息的函数过程与查询用户信息过程类似，打开指定通讯录后再进行用户信息的修改操作。



#### 6、清空用户信息

该模块用于清空指定通讯录的信息。首先进入该模块后，程序会提示用户要清空哪一个通讯录的信息，当用户输入指定通讯录后，程序会让用户再次确认是否清空，若用户确认，则该通讯录信息被清空。



#### 7、合并通讯录信息

该模块用于合并多个通讯录信息。当用户进入该模块后，程序会提示用户要合并几个通讯录文件，用户接着输入要合并的通讯录名称，最后指定要合并到哪个通讯录名，合并通讯录成功。

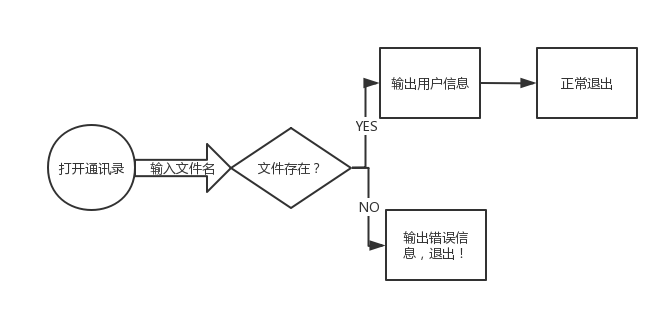


# 三、面向对象C++程序设计实施

## 1、打开通讯录

当使用该功能时，程序会要求用户输入一个通讯录名称，输入完毕后程序会在当前目录下进行查询，若查找到该名称的通讯录，程序则会将该通讯录中的用户信息打印至屏幕。

#### (1)流程图

****

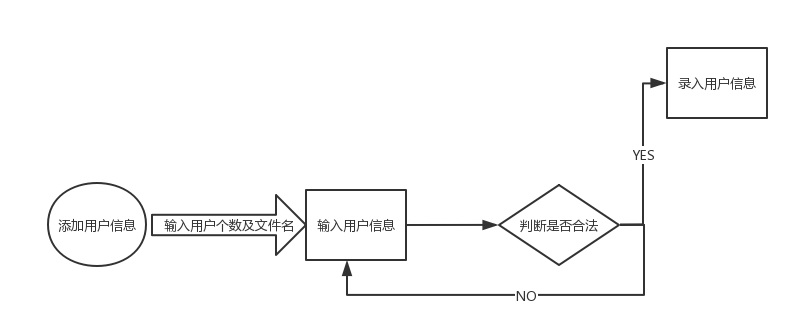
#### (2)源码

1. **void** open\_File(list<User> &User\_List)
2. string File\_name;
3. string user\_addr;
4. string phone\_num;
5. string temp="";
6. string user\_name;
7. **int** user\_No;
8. **int** user\_age;
9. cout << "请输入您要打开的通讯录名称：" << endl;
10. cin >> File\_name;
11. ifstream ifs;
12. User user;
13. ifs.open(File\_name, ios::in);
14. **if** (!ifs)
15. {
16. cout << "你输入的通讯录不存在！" << endl;
17. **return**;
18. }
19. **else**
20. {
22. **while** (!ifs.eof())
23. {
24. ifs >> phone\_num >> user\_addr >> user\_name >> user\_age >> user\_No;
25. user.set\_addr(user\_addr);
26. user.set\_num(phone\_num);
27. user.set\_name(user\_name);
28. user.set\_age(user\_age);
29. user.set\_No(user\_No);
30. **if** (temp != phone\_num)
31. User\_List.push\_back(user);
32. temp = phone\_num;
33. }
34. ifs.close();
35. **if** (!User\_List.empty())
36. {
37. cout << "当前通讯录用户信息如下：" << endl;
38. **for** (list<User>::iterator it = User\_List.begin(); it != User\_List.end(); it++)
39. {
40. cout << "\t编号： " << it->get\_No() << " 姓名： " << it->get\_name() << " 电话： " << it->get\_num() << " 年龄： " << it->get\_age() << " 住址： " << it->get\_addr() << endl;
41. }
42. User\_List.clear();
43. }
44. **else**
45. {
46. cout << "当前通讯录下用户信息为空！" << endl;
47. **return**;
48. }
49. }

## 添加用户信息

当用户使用该功能时，程序会提示用户输入添加几个用户以及添加至什么文件名，随后会让用户输入要添加的用户的各种信息。在用户输入信息时，程序会判断用户输入是否合法，比如输入电话号码时会对号码位数进行判断，不合法则提示用户重新输入。

#### (1)流程图

****

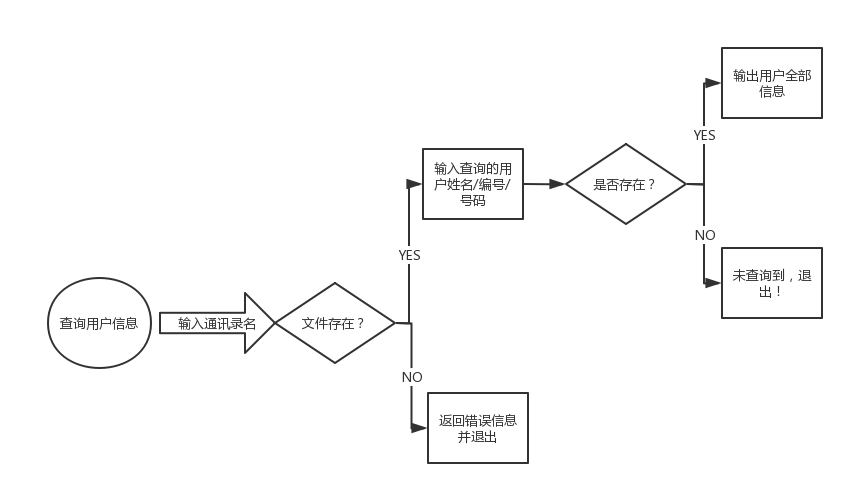
#### (2)源码

1. **void** add\_File()
2. {
3. **int** user\_No;
4. string temp = "";
5. string File\_name;
6. string phone\_num;
7. string user\_name;
8. **int** user\_age;
9. string user\_addr;
10. User user;
11. cout << "请输入您要添加几个用户信息：" << endl;
12. **int** n;
13. cin >> n;
14. **if** (n <= 0)
15. {
16. cout << "您输入的用户个数有误！" << endl;
17. **return**;
18. }
19. cout << "请输入您要保存数据到什么文件名" << endl;
20. cin >> File\_name;
21. ofstream outfile;
22. ifstream ifs;
23. ifs.open(File\_name, ios::in);
24. **if**(ifs){
25. **while** (!ifs.eof())
26. {
27. ifs >> phone\_num >> user\_addr >> user\_name >> user\_age >> user\_No;
28. **if** (temp != phone\_num)
29. User::add\_User();
30. temp = phone\_num;
31. }
32. }
33. ifs.close();
34. outfile.open(File\_name, ios::out | ios::app);
35. **for** (**int** i = 0; i < n; i++)
36. {
37. User::add\_User();
38. cout << "请输入第"<<(i+1)<<"个用户的号码：" << endl;
39. cout << "(示例：13822222222/021-1234567)" << endl;
40. B:      cin >> phone\_num;
41. regex pattern("(\\d{11})|(\\d{3}-\\d{7,8})");
42. **if** (regex\_match(phone\_num, pattern))
43. {
44. user.set\_num(phone\_num);
45. }
46. **else**
47. {
48. cout << "您输入的电话号码不符合格式要求，请重新输入！" << endl;
49. **goto** B;
50. }
51. cout << "请输入第" << (i + 1) << "个用户的姓名：" << endl;
52. cin >> user\_name;
53. user.set\_name(user\_name);
54. cout << "请输入第" << (i + 1) << "个用户的年龄：" << endl;
55. A:  cin >> user\_age;
56. **if** (user\_age <= 0 || user\_age >= 100)
57. {
58. cout << "您输入年龄不合法，请重新输入！" << endl;
59. **goto** A;
60. }
61. user.set\_age(user\_age);
62. cout << "请输入第" << (i + 1) << "个用户的住址：" << endl;
63. cin >> user\_addr;
64. user.set\_addr(user\_addr);
65. user.set\_No(User::get\_sum());
66. outfile << user.get\_num() << " " << user.get\_addr() << " " << user.get\_name() << " " << user.get\_age() <<" "<<user.get\_No()<< endl;
67. }
68. User::set\_sum\_0();
69. outfile.close();
71. cout << "保存成功！" << endl;
73. }

## 查询用户信息

当用户使用该功能时，程序会首先提示用户要查询哪个通讯录的用户信息，当用户输入要查询的通讯录后，程序会提示用户选择查询信息的方式，紧接着进行查找，与添加用户一样，这里输入查询条件时也会对用户输入数据进行合法性判断。

#### (1)流程图

****

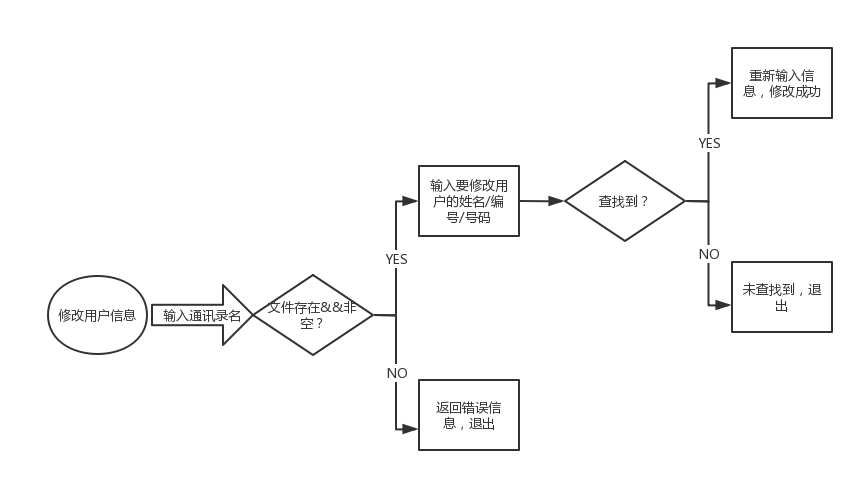
#### 源码

1. **void** search\_info(list<User> &User\_List)
2. {
3. **int** begin = 0;
4. string File\_name;
5. **if** (!read\_File(User\_List, File\_name, begin))
6. {
7. **return**;
8. }
9. **if** (User\_List.empty())
10. {
11. cout << "当前通讯录内容为空！" << endl;
12. **return**;
13. }
14. cout << "请输入您想通过什么方式查找用户信息：" << endl;
15. cout << "1、用户姓名" << endl;
16. cout << "2、用户编号" << endl;
17. cout << "3、用户号码" << endl;
18. cout << "0、退出" << endl;
19. **int** choice;
20. cin >> choice;
21. **switch** (choice)
22. {
23. **case** 1:
24. {   string name;
25. cout << "请输入您要查询的用户姓名： " << endl;
26. cin >> name;
27. **bool** flag = **false**;
28. **for** (list<User>::iterator it = User\_List.begin(); it != User\_List.end(); it++)
29. {
30. **if** (it->get\_name() == name)
31. {
32. cout << "查询到该用户信息：" << endl;
33. cout << "\t编号： " << it->get\_No() << " 姓名： " << it->get\_name() << " 电话： " << it->get\_num() << " 年龄： " << it->get\_age() << " 住址： " << it->get\_addr() << endl;
34. flag = **true**;
35. **break**;
36. }
37. }
38. **if** (!flag)
39. {
40. cout << "当前文件中未查询到相关用户信息!" << endl;
41. }
42. system("pause");
43. system("cls");
44. **break**;
45. }
46. **case** 2:
47. {
48. **int** no;
49. cout << "请输入您要查询的用户编号： " << endl;
50. cin >> no;
51. **bool** flag = **false**;
52. **for** (list<User>::iterator it = User\_List.begin(); it != User\_List.end(); it++)
53. {
54. **if** (it->get\_No() == no)
55. {
56. cout << "查询到该用户信息：" << endl;
57. cout << "\t编号： " << it->get\_No() << " 姓名： " << it->get\_name() << " 电话： " << it->get\_num() << " 年龄： " << it->get\_age() << " 住址： " << it->get\_addr() << endl;
58. **break**;
59. }
60. }
61. **if** (!flag)
62. {
63. cout << "当前文件中未查询到相关用户信息!" << endl;
64. }
65. system("pause");
66. system("cls");
67. **break**;
68. }
69. **case** 3:
70. {
71. string num;
72. cout << "请输入您要查询的用户号码： " << endl;
73. cin >> num;
74. **bool** flag = **false**;
75. **for** (list<User>::iterator it = User\_List.begin(); it != User\_List.end(); it++)
76. {
77. **if** (it->get\_num() == num)
78. {
79. cout << "查询到该用户信息：" << endl;
80. cout << "\t编号： " << it->get\_No() << " 姓名： " << it->get\_name() << " 电话： " << it->get\_num() << " 年龄： " << it->get\_age() << " 住址： " << it->get\_addr() << endl;
81. flag = **true**;
82. **break**;
83. }
84. }
85. **if** (!flag)
86. {
87. cout << "当前文件中未查询到相关用户信息!" << endl;
88. }
89. system("pause");
90. system("cls");
91. **break**;
92. }
93. **case** 0:
94. **return**;
95. **default**:
96. cout << "您的输入有误，请重新输入！" << endl;
97. system("pause");
98. system("cls");
99. **break**;
100. }
101. }

## 删除用户信息

当用户使用该功能时，程序会首先提示用户要删除哪个通讯录的用户信息，当用户输入要删除的通讯录后，程序会提示用户选择删除信息的方式，紧接着进行删除，与添加用户一样，这里输入查询条件时也会对用户输入数据进行合法性判断。

#### (1)流程图

****

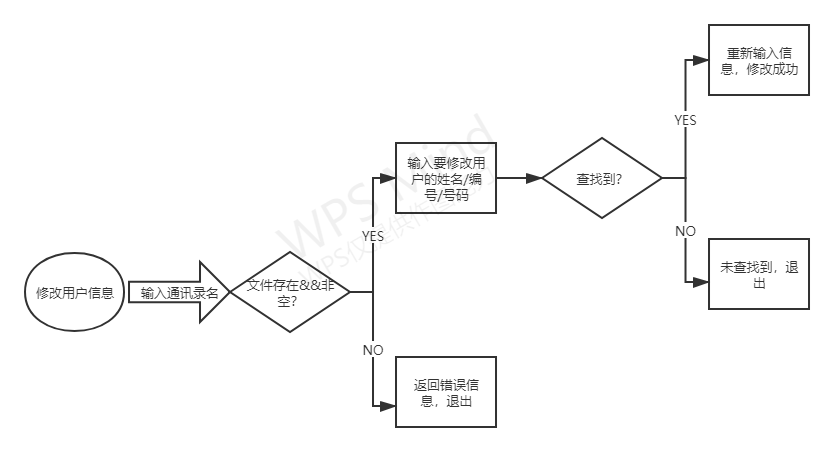
#### (2)源码

1. **void** del\_User(list<User>& User\_List)
3. **int** begin = 0;
4. string File\_name;
5. **if** (!read\_File(User\_List, File\_name, begin))
6. {
7. **return**;
8. }
9. **if** (User\_List.empty())
10. {
11. cout << "当前通讯录内容为空！" << endl;
12. **return**;
13. }
14. cout << "请输入您想通过什么方式删除用户信息：" << endl;
15. cout << "1、用户姓名" << endl;
16. cout << "2、用户编号" << endl;
17. cout << "3、用户号码" << endl;
18. cout << "0、退出" << endl;
19. **int** choice;
20. cin >> choice;
21. **switch** (choice)
22. {
23. **case** 1:
24. {   string name;
25. cout << "请输入您要删除的用户姓名： " << endl;
26. cin >> name;
27. **bool** flag = **false**;
28. **for** (list<User>::iterator it = User\_List.begin(); it != User\_List.end(); it++)
29. {
30. **if** (it->get\_name() == name)
31. {
32. User\_List.erase(it);
33. cout << "已删除该用户" << endl;
34. flag = **true**;
35. **break**;
36. }
37. }
38. **if** (!flag)
39. {
40. cout << "当前文件中未查询到相关用户信息!" << endl;
41. }
42. system("pause");
43. system("cls");
44. **break**;
45. }
46. **case** 2:
47. {
48. **int** no;
49. cout << "请输入您要删除的用户编号： " << endl;
50. cin >> no;
51. **bool** flag = **false**;
52. **for** (list<User>::iterator it = User\_List.begin(); it != User\_List.end(); it++)
53. {
54. **if** (it->get\_No() == no)
55. {
56. User\_List.erase(it);
57. cout << "已删除该用户" << endl;
58. flag = **true**;
59. **break**;
60. }
61. }
62. **if** (!flag)
63. {
64. cout << "当前文件中未查询到相关用户信息!" << endl;
65. }
66. system("pause");
67. system("cls");
68. **break**;
69. }
70. **case** 3:
71. {
72. string num;
73. cout << "请输入您要删除的用户号码： " << endl;
74. cin >> num;
75. **bool** flag = **false**;
76. **for** (list<User>::iterator it = User\_List.begin(); it != User\_List.end(); it++)
77. {
78. **if** (it->get\_num() == num)
79. {
80. User\_List.erase(it);
81. cout << "已删除该用户" << endl;
82. flag = **true**;
83. **break**;
84. }
85. }
86. **if** (!flag)
87. {
88. cout << "当前文件中未查询到相关用户信息!" << endl;
89. }
90. system("pause");
91. system("cls");
92. **break**;
93. }
94. **case** 0:
95. **return**;
96. **default**:
97. cout << "您的输入有误，请重新输入！" << endl;
98. system("pause");
99. system("cls");
100. **break**;
101. }
102. save\_File(User\_List,File\_name);
103. User\_List.clear();
104. }

## 修改用户信息

当用户使用该功能时，程序会首先提示用户要修改哪个通讯录的用户信息，当用户输入要查修改的通讯录后，程序会提示用户选择修改信息的方式，紧接着进行修改，与添加用户一样，这里输入查询条件时也会对用户输入数据进行合法性判断。

#### (1)流程图

****

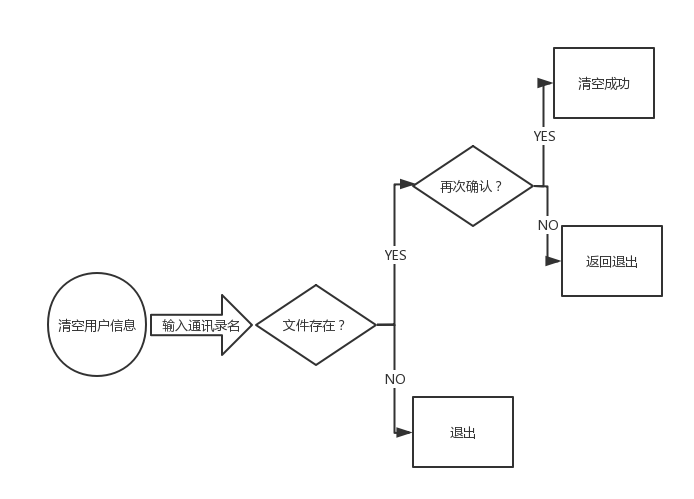
#### (2)源码

1. **void** mod\_User(list<User>& User\_List)
2. {
3. string phone\_num;
4. string user\_name;
5. **int** user\_age;
6. string user\_addr;
7. **int** begin = 0;
8. string File\_name;
9. **if** (!read\_File(User\_List, File\_name, begin))
10. {
11. **return**;
12. }
13. **if** (User\_List.empty())
14. {
15. cout << "当前通讯录内容为空！" << endl;
16. **return**;
17. }
18. cout << "请输入您想通过什么方式修改用户信息：" << endl;
19. cout << "1、用户姓名" << endl;
20. cout << "2、用户编号" << endl;
21. cout << "3、用户号码" << endl;
22. cout << "0、退出" << endl;
23. **int** choice;
24. cin >> choice;
25. **switch** (choice)
26. {
27. **case** 1:
28. {   string name;
29. cout << "请输入您要修改的用户姓名： " << endl;
30. cin >> name;
31. **bool** flag = **false**;
32. **for** (list<User>::iterator it = User\_List.begin(); it != User\_List.end(); it++)
33. {
34. **if** (it->get\_name() == name)
35. {
36. cout << "已找到该用户信息：" << endl;
37. cout << "请输入该用户的新号码：" << endl;
38. cin >> phone\_num;
39. it->set\_num(phone\_num);
40. cout << "请输入该用户的新姓名：" << endl;
41. cin >> user\_name;
42. it->set\_name(user\_name);
43. cout << "请输入该用户的新年龄：" << endl;
44. cin >> user\_age;
45. it->set\_age(user\_age);
46. cout << "请输入该用户的新住址：" << endl;
47. cin >> user\_addr;
48. it->set\_addr(user\_addr);
50. flag = **true**;
51. **break**;
52. }
53. }
54. **if** (!flag)
55. {
56. cout << "当前文件中未查询到相关用户信息!" << endl;
57. }
58. system("pause");
59. system("cls");
60. **break**;
61. }
62. **case** 2:
63. {
64. **int** no;
65. cout << "请输入您要修改的用户编号： " << endl;
66. cin >> no;
67. **bool** flag = **false**;
68. **for** (list<User>::iterator it = User\_List.begin(); it != User\_List.end(); it++)
69. {
70. **if** (it->get\_No() == no)
71. {
72. cout << "已找到该用户信息：" << endl;
73. cout << "请输入该用户的新号码：" << endl;
74. cin >> phone\_num;
75. it->set\_num(phone\_num);
76. cout << "请输入该用户的新姓名：" << endl;
77. cin >> user\_name;
78. it->set\_name(user\_name);
79. cout << "请输入该用户的新年龄：" << endl;
80. cin >> user\_age;
81. it->set\_age(user\_age);
82. cout << "请输入该用户的新住址：" << endl;
83. cin >> user\_addr;
84. it->set\_addr(user\_addr);
86. flag = **true**;
87. **break**;
88. }
89. }
90. **if** (!flag)
91. {
92. cout << "当前文件中未查询到相关用户信息!" << endl;
93. }
94. system("pause");
95. system("cls");
96. **break**;
97. }
98. **case** 3:
99. {
100. string num;
101. cout << "请输入您要修改的用户号码： " << endl;
102. cin >> num;
103. **bool** flag = **false**;
104. **for** (list<User>::iterator it = User\_List.begin(); it != User\_List.end(); it++)
105. {
106. **if** (it->get\_num() == num)
107. {
108. cout << "已找到该用户信息：" << endl;
109. cout << "请输入该用户的新号码：" << endl;
110. cin >> phone\_num;
111. it->set\_num(phone\_num);
112. cout << "请输入该用户的新姓名：" << endl;
113. cin >> user\_name;
114. it->set\_name(user\_name);
115. cout << "请输入该用户的新年龄：" << endl;
116. cin >> user\_age;
117. it->set\_age(user\_age);
118. cout << "请输入该用户的新住址：" << endl;
119. cin >> user\_addr;
120. it->set\_addr(user\_addr);
121. flag = **true**;
122. **break**;
123. }
124. }
125. **if** (!flag)
126. {
127. cout << "当前文件中未查询到相关用户信息!" << endl;
128. }
129. system("pause");
130. system("cls");
131. **break**;
132. }
133. **case** 0:
134. **return**;
135. **default**:
136. cout << "您的输入有误，请重新输入！" << endl;
137. system("pause");
138. system("cls");
139. **break**;
140. }
141. save\_File(User\_List, File\_name);
142. User\_List.clear();
143. }

## 清空用户信息

首先进入该模块后，程序会提示用户要清空哪一个通讯录的信息，当用户输入指定通讯录后，程序会让用户再次确认是否清空，若用户确认，则该通讯录信息被清空。

#### (1)流程图

****

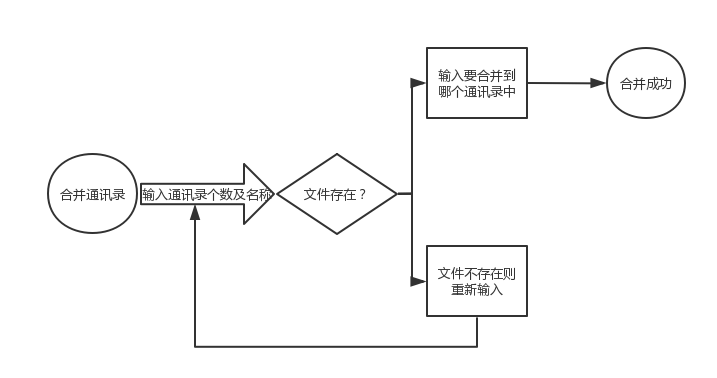
#### (2)源码

1. **void** clear\_User(list<User>& User\_List)
3. **int** begin = 0;
4. string File\_name;
5. **if** (!read\_File(User\_List, File\_name, begin))
6. {
7. **return**;
8. }
9. **if** (User\_List.empty())
10. {
11. cout << "当前通讯录内容为空,无需执行此操作！" << endl;
12. **return**;
13. }
14. cout << "您确定您要清空该通讯录吗？(Y/N)" << endl;
15. **char** choice;
16. :   cin >> choice;
17. **if** (choice == 'Y' || choice == 'y')
18. {
19. User\_List.clear();
20. save\_File(User\_List, File\_name);
21. cout << "当前通讯录已清空！" << endl;
22. }
23. **else** **if** (choice == 'N' || choice == 'n')
24. {
25. User\_List.clear();
26. cout << "已撤销清空操作！" << endl;
27. **return**;
28. }
29. **else**
30. {
31. cout << "您的输入有误,请重新输入！" << endl;
32. **goto** A;
33. }

## 合并用户信息

当用户进入该模块后，程序会提示用户要合并几个通讯录文件，用户接着输入要合并的通讯录名称，最后指定要合并到哪个通讯录名，合并通讯录成功。

#### (1)流程图

****

#### (2)源码

1. **void** merge\_File(list<User>& User\_List)
2. {
3. cout << "请输入您要合并几个通讯录文件：" << endl;
4. **int** n;
5. **int** count = 1;
6. **int** begin = 0;
7. cin >> n;
8. **if** (n <= 0)
9. {
10. cout << "您输入的通讯录个数有误！" << endl;
11. **return**;
12. }
13. string Filename;
14. **for** (**int** i = 0; i < n; i++)
15. {
16. cout << "请输入第 " << (i + 1) << " 个文件名：" << endl;
17. **bool** p=read\_File(User\_List, Filename, begin);
18. **if** (!p&&count<=3)
19. {
20. i--;
21. count++;
22. cout << "请重新输入通讯录名称：" << endl;
23. }
24. **else**
25. {
26. **if** (count <= 3)
27. remove(Filename.c\_str());
28. **else**
29. {
30. cout << "您输入通讯录名称次数过多，默认您已输入完毕！" << endl;
31. **break**;
32. }
33. }
34. }
35. **if** (User\_List.empty())
36. {
37. cout << "您输入的通讯录文件内用户信息均为空，无法合并！" << endl;
38. **return**;
39. }
40. cout << "请输入您要将以上通讯录文件内信息合并到哪个通讯录文件中：" << endl;
41. cin >> Filename;
42. ofstream ofs;
43. ofs.open(Filename, ios::out);
44. **for** (list<User>::iterator it = User\_List.begin(); it != User\_List.end(); it++)
45. {
46. ofs << it->get\_num() << " " << it->get\_addr() << " " << it->get\_name() << " " << it->get\_age() << " " << it->get\_No() << endl;
47. }
48. User\_List.clear();
49. cout << "合并完毕！" << endl;
50. ofs.close();
51. }

# 四、结束语

**通过此次课程设计，使我更加扎实的掌握了有关链表和结构体以及类方面的知识，在设计过程中，中虽然遇到了一些问题，但经过一次又一次的思考，一遍又一遍的检查终于找出了原因。所在，也暴露出了前期我在这方面的知识欠缺和经验不足。实践出真知，通过亲自动手制作，使我们掌握的知识不再是纸上谈兵。这次课程设计终于顺利完成了，在今后社会的发展和学习实践过程中，一定要不懈努力，不能遇到问题就想到要退缩，一定要不厌其烦的发现问题所在，然后一一进行解决，只有这样，才能成功的做成想做的事，才能在今后的道路上劈荆斩棘，而不是知难而退，那样永远不可能收课程设计诚然是一门专业课，给我很多专业知识以及专业技能上的提升，同时又是一门讲道课，一门辩思课，给了我许多道，给了我很多思，给了我莫大的空间。同时，设计让我感触很深。使我对抽象的理论有了具体的认识。通过这次课程设计，我对建立一个工程有了一个新的认识，首先是想清楚这个系统的流程，将框架搭建起来后，再起写一些细节方面的东西，千万不能稀里糊涂的写，到最后没有用处，白白浪费时间。**

**我认为此次课设不仅培养了独立思考、动手操作的能力，在各种其它能力上也都有了提高。同时也希望今后能够多花时间去思考，去动手写一些代码。**

# 参考文献

[1]谭浩强 《c++程序设计》 北京；清华大学出版社 2004

[2]钱能 《c++程序设计》 北京；清华大学出版社 1999  
　　 [3]邰非。 基于实践的C++互动教学模式的建立[J]. 高教学刊

[4]徐晓凯 《c++语言基础教程》 北京；清华大学出版社 2002