

COMPANY NAME

C++课程设计报告书

**2021**

**人事管理系统**

**BY 倪路 1910400731**

目录

[一、 课设概况 2](#_Toc15378)

[二、 系统简介 3](#_Toc31554)

[1、 项目背景 3](#_Toc10298)

[2、 编写目的 3](#_Toc9829)

[3、 软件定义 3](#_Toc3738)

[4、 开发环境 3](#_Toc6157)

[三、 主题内容 3](#_Toc14062)

[Ⅰ系统设计说明 3](#_Toc25674)

[1、 需求分析 3](#_Toc19244)

[2、 系统功能描述 3](#_Toc25302)

[3、 总系统功能图 4](#_Toc11090)

[4、 模块功能说明 4](#_Toc1573)

[5、 系统流程分析 11](#_Toc28876)

[Ⅱ存储数据结构说明 31](#_Toc14244)

[1、 数据结构 31](#_Toc9760)

[2、 数据记录创建 31](#_Toc19285)

[3、 存储过程 31](#_Toc7342)

[Ⅲ 应用界面说明 32](#_Toc27882)

[1、 总系统界面 32](#_Toc22991)

[2、 简要说明 32](#_Toc15111)

[3、 主要函数 33](#_Toc7579)

[4、 核心源码 35](#_Toc10587)

[Ⅳ 遗留问题 55](#_Toc22804)

[Ⅴ 设计体会 55](#_Toc18602)

[四、 参考文献 55](#_Toc22165)

# 课设概况

|  |  |  |  |
| --- | --- | --- | --- |
| **课设题目** | **人事信息管理系统** | | |
| **姓名** | **学号** | **职务** | **任务分工** |
| 倪路 | 1910400731 | 组长 | 所有任务 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 系统简介

### 项目背景

随着人类迈入二十一世纪，人们的科学技术的飞速发展，人类正在从工业时代向信息时代迈进；利用计算机提高企业人事管理的效率，是适应现代企业制度的要求、推动企业人事管理走向科学化、规范化的调教；

### 编写目的

人事的基本信息管理是一项琐碎、复杂而又细致的工作，为了能够更加快速高效的管理大量人事的基本信息，为了达到这一要求，于是我编写了本程序。

### 软件定义

本程序提供人员信息的输入、删除、查找、修改等基本功能，并且能够按部门进行各项分类统计，对各部门或整体进行排序和筛选等操作。

### 开发环境

Visual Studio 2019

# **主题内容**

## Ⅰ系统设计说明

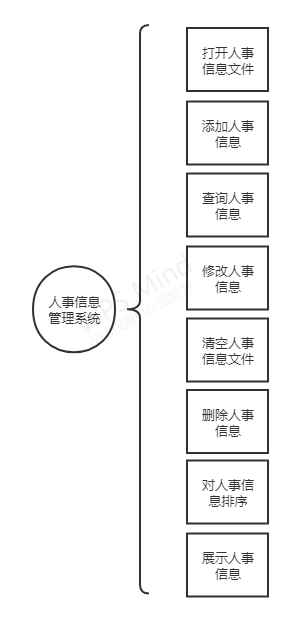
### **需求分析**

程序应采用面向对象的程序设计方法，程序应提供人员信息的输入、删除、查找、修改等基本功能，并且能够按部门进行各项分类统计，对各部门或整体进行排序和筛选等操作。

### **系统功能描述**

根据题目要求，本系统所提供的功能有打开人事信息文件，以及对人事信息的查询、添加、删除、修改、清空、展示以及排序等操作。

### **总系统功能图**



### **模块功能说明**

##### 打开人事信息文件

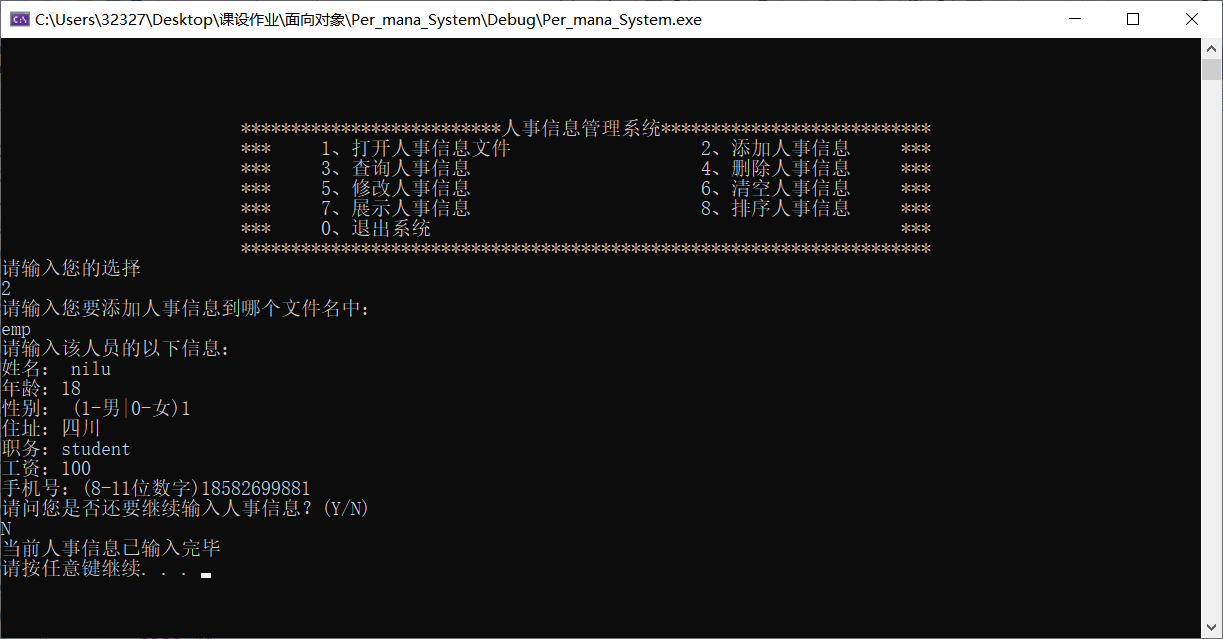
该模块用于从硬盘中读取人事信息到内存中。首先进入该模块，系统会提示用户输入要打开的人事信息名，当用户输入的文件名存在时，系统则会将人事信息读取到list容器中供用户接下来操作，若文件信息不存在，则系统会提示用户重新输入，多次输错系统会退出该模块。





##### **添加人事信息**

该模块用于添加人事信息到指定文件夹中，进入该模块，系统会先提示用户需要将信息保存至哪一个文件下，若该文件不存在则会提示用户是否要创建一个新文件；接着会让用户继续输入需要添加的人事信息，最终保存到相应文件中。



##### **查询人事信息**

该模块用于让用户根据指定人员姓名或者手机号来查询对应人员的详细信息。进入该模块，系统会提示用户根据姓名或者手机号来查找用户，用户选择后输入要查找的关键字，随后系统会在list容器中进行查找，查找成功则打印相关人事的详细信息，否则打印查无此人。





##### **修改人事信息**

该模块用于修改人事的信息。进入该模块后，系统会提示用户输入要修改具体哪一个人事的信息，如果查找到该人员，则系统再次让用户确认是否修改，确认后，用户需手动输入相应更新后的人员信息，最后完成修改操作。





##### **清空人事信息**

该模块用于清空人事信息文件的所有内容；当用户进入该模块，程序会提示用户输入要清空的人事信息文件名，若该文件存在则再次提醒用户是否确认删除，用户确认后则完成清空操作。





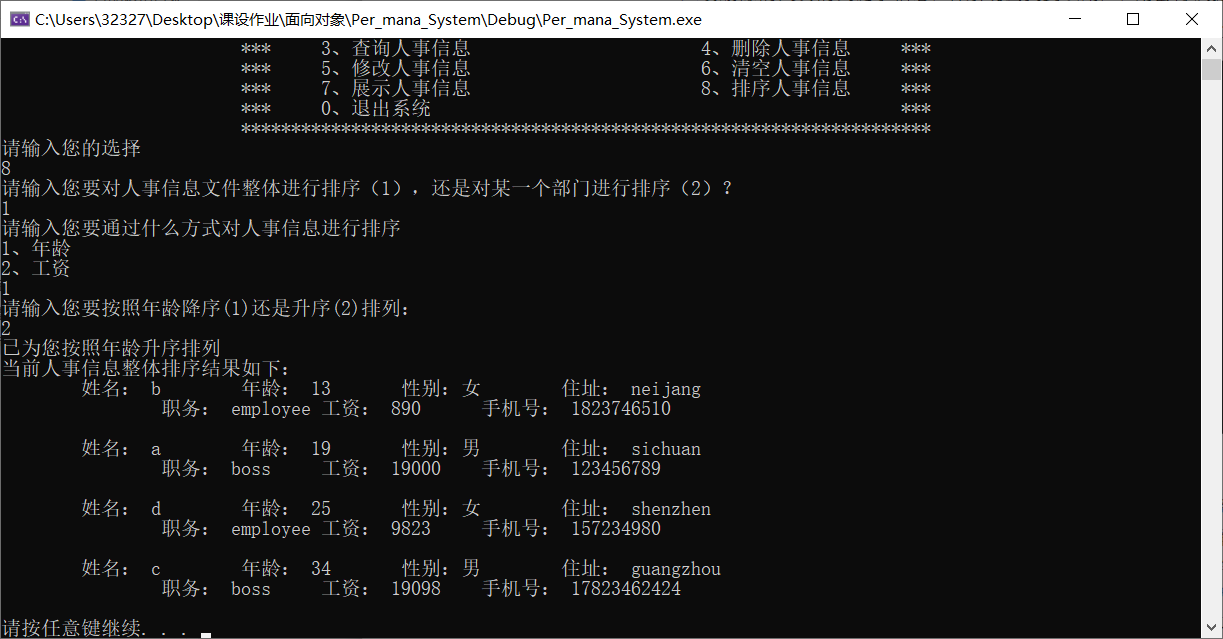
##### **删除人事信息**

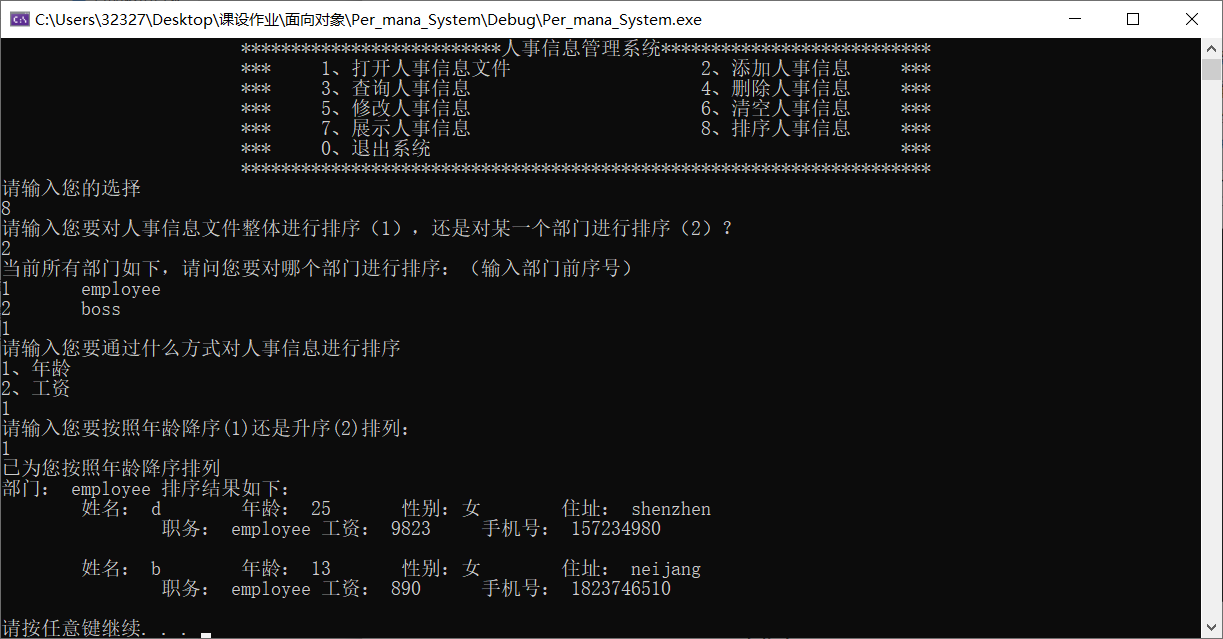
该模块用于删除人事信息；当用户进入该模块后，系统会提示用户输入要删除的人员的关键字信息，若系统查找到该用户，则会再次提醒用户是否确认删除，用户确认后则删除成功并更新相应人事信息文件。



##### **对人事信息排序**

该模块用于对人事信息进行排序；当用户进入该模块后，系统会提示用户是对整体进行排序或者是对某一个部门进行排序；当用户选择对整体排序后，系统会直接对list容器调用其内指定sort算法并按照年龄或者工资来对人事信息排序；若用户选择对部门进行排序，则系统会将所有部门展示出来并提示用户要对哪一个部门进行排序，最后完成排序要求。





##### **展示人事信息**

该模块用于展示人事信息；当用户进入该模块后，系统会将用户打开的当前人事信息文件中的人事信息按照部门分类统计好打印输入至屏幕上。

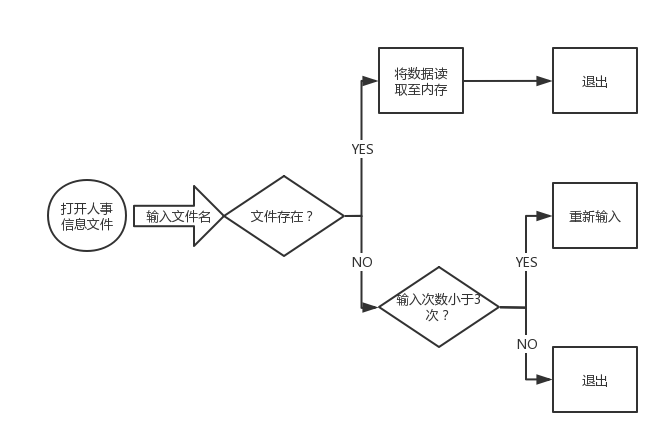


### **系统流程分析**

##### 打开人事信息文件

进入该功能，系统会提示用户输入要打开的人事信息名，当用户输入的文件名存在时，系统则会将人事信息读取到内存中，若文件信息不存在，则系统会提示用户重新输入，多次输错系统会退出该模块。

**①流程图**

****

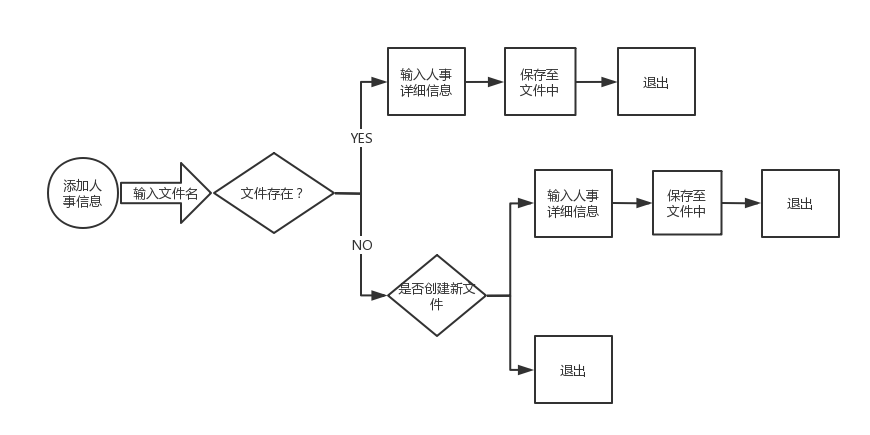
**②源码**

1. **void** read\_info(list<Person>& mylist,PortList\*& Port ,string& Filename)
3. mylist.clear();
4. string name;
5. **int** age;
6. string port;
7. **float** wages;
8. string addr;
9. string phone\_num;
10. **int** file\_error\_count = 0;
11. **int** sex;
12. Person person;
13. cout << "请输入您要打开的人事信息文件：" << endl;
14. cin >> Filename;
15. Filename = Filename + ".txt";
16. ifstream ifs(Filename, ios::in);
17. **while** (file\_error\_count < 3)
18. {
19. **if** (!ifs)
20. {
21. cout << "您输入的人事管理信息文件不存在，请检查后重新输入..." << endl;
22. file\_error\_count++;
23. cout << "您现在还有 " << (3 - file\_error\_count) << "次输入机会" << endl;
24. **if** (file\_error\_count < 3)
25. {
26. cout << "请输入您要打开的人事信息文件：" << endl;
27. cin >> Filename;
28. Filename = Filename + ".txt";
29. ifstream ifs(Filename, ios::in);
30. }
31. }
32. **else**
33. {
34. **break**;
35. }
36. }
37. **if** (file\_error\_count == 3)
38. {
39. cout << "输入错误次数过多，已自动退出！" << endl;
40. **return**;
41. }
42. **int** count = 0;
43. **while** (!ifs.eof())
44. {
45. ifs >> name >> age >>sex>> addr >> port >> wages >> phone\_num;
46. **if** (age < 0)
47. {
48. **continue**;
49. }
50. person.set\_name(name);
51. person.set\_age(age);
52. person.set\_sex(sex);
53. person.set\_phone(phone\_num);
54. person.set\_port(port);
55. person.set\_wages(wages);
56. person.set\_addr(addr);
57. mylist.push\_front(person);
58. count++;
59. }
60. ifs.close();
61. read\_count\_info(mylist, Port);
62. }

##### **添加人事信息**

进入该模块，系统会先提示用户需要将信息保存至哪一个文件下，若该文件不存在则会提示用户是否要创建一个新文件；接着会让用户继续输入需要添加的人事信息，最终保存到相应文件中。

①流程图



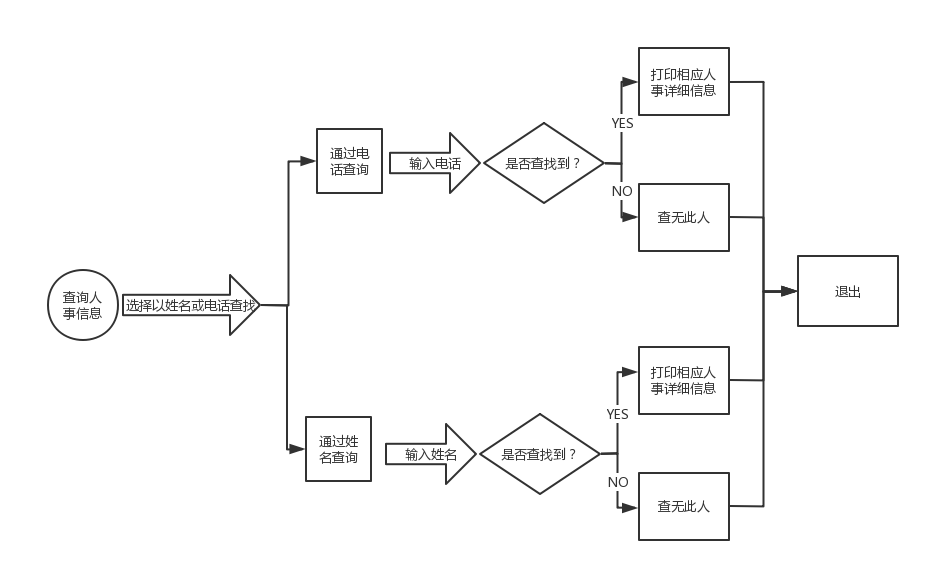
②源码

1. **void** input\_info()
2. {
3. string Filenamne;
4. list<Person> mylist;
5. PortList \*Port;
6. cout << "请输入您要添加人事信息到哪个文件名中： " << endl;
7. cin >> Filenamne;
8. Filenamne = Filenamne + ".txt";
9. ifstream ifs(Filenamne);
10. **if** (!ifs)
11. {
12. cout << "当前文件不存在，请问您是否要创建新文件？(Y/N)" << endl;
13. **char** c;
14. K:  cin >> c;
15. **if**(c == 'N' || c == 'n')
16. {
17. cout << "添加人事信息失败！" << endl;
18. **return**;
19. }
20. **else**  **if** (c == 'Y' || c == 'y')
21. {}
22. **else**
23. {
24. cout << "您的输入有误，请重新输入：" << endl;
25. **goto** K;
26. }
27. }
28. **else**
29. {
30. read\_info\_temp(mylist, Filenamne);
31. }
32. string name;
33. **int** age;
34. string port;
35. **float** wages;
36. string addr;
37. string phone\_num;
38. Person person;
39. **int** sex;
40. **while** (1)
41. {
42. cout << "请输入该人员的以下信息：" << endl;
43. cout << "姓名： ";
44. cin >> name;
45. cout << "年龄：";
46. H:  cin >> age;
47. **if** (age <= 0 || age > 150)
48. {
49. cout << "您的输入不合法，请重新输入" << endl;
50. **goto** H;
51. }
52. cout << "性别： (1-男|0-女)";
53. cin >> sex;
54. cout << "住址：";
55. cin >> addr;
56. cout << "职务：";
57. cin >> port;
58. cout << "工资：";
59. cin >> wages;
60. cout << "手机号：(8-11位数字)";
61. regex pattern("\\d{8,11}");
62. **int** count = 0;
63. Z:  cin >> phone\_num;
64. **if** (!regex\_match(phone\_num, pattern))
65. {
66. cout << "您的输入不合法，请重新输入！" << endl;
67. count++;
68. **if** (count < 3)
69. **goto** Z;
70. **else**
71. {
72. cout << "您已多次错误输入，已退出！" << endl;
73. **return**;
74. }
75. }
76. person.set\_name(name);
77. person.set\_addr(addr);
78. person.set\_age(age);
79. person.set\_phone(phone\_num);
80. person.set\_port(port);
81. person.set\_sex(sex);
82. person.set\_wages(wages);
83. mylist.push\_back(person);
84. cout << "请问您是否还要继续输入人事信息？(Y/N)" << endl;
85. **char** choice;
86. A:  cin >> choice;
87. **if** (choice == 'Y' || choice == 'y')
88. {
89. **continue**;
90. }
91. **else** **if** (choice == 'N' || choice == 'n')
92. {
93. cout << "当前人事信息已输入完毕" << endl;
94. **break**;
95. }
96. **else**
97. {
98. cout << "您的输入有误，请重新输入" << endl;
99. **goto** A;
100. }
101. }
102. save\_info(mylist, Filenamne);
103. }

##### **查询人事信息**

进入该模块，系统会提示用户根据姓名或者手机号来查找用户，用户选择后输入要查找的关键字，随后系统会进行查找，查找成功则打印相关人事的详细信息，否则打印查无此人。

①**流程图**



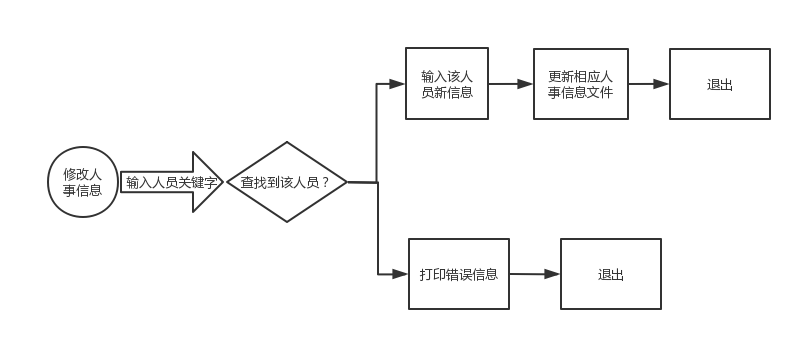
**②源码**

1. **void** search\_info(list<Person>& mylist)
2. {
3. **if** (mylist.empty())
4. {
5. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
6. **return**;
7. }
8. cout << "请输入您要通过什么方式查找人事信息" << endl;
9. cout << "1、姓名" << endl;
10. cout << "2、电话" << endl;
11. string name, phone;
12. **int** choice;
13. **bool** flag = **false**;
14. A:  cin >> choice;
15. **switch** (choice)
16. {
17. **case** 1:
18. {
19. cout << "请输入您要查找的人员姓名： " << endl;
20. cin >> name;
21. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
22. {
23. **if** (it->get\_name() == name)
24. {
25. cout << "已找到该用户，该用户详细信息如下：" << endl;
26. it->dis\_info();
27. flag = **true**;
28. **break**;
29. }
30. }
31. **if** (!flag)
32. {
33. cout << "当前人事信息文件中未查找到姓名为： "<<name<<"的用户！" << endl;
34. }
35. **break**;
36. }
37. **case** 2:
38. {
39. cout << "请输入您要查找的人员电话：(8-11位数字) " << endl;
40. regex pattern("\\d{8,11}");
41. **int** count = 0;
42. Z:  cin >> phone;
43. **if** (!regex\_match(phone, pattern))
44. {
45. cout << "您的输入不合法，请重新输入！" << endl;
46. count++;
47. **if**(count<3)
48. **goto** Z;
49. **else**
50. {
51. cout << "您已多次错误输入，已退出！" << endl;
52. **return**;
53. }
54. }
55. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
56. {
57. **if** (it->get\_phone() == phone)
58. {
59. cout << "已找到该用户，该用户详细信息如下：" << endl;
60. it->dis\_info();
61. flag = **true**;
62. **break**;
63. }
64. }
65. **if** (!flag)
66. {
67. cout << "当前人事信息文件中未查找到手机号为： " << phone << "的用户！" << endl;
68. }
69. **break**;
70. }
71. **default**:
72. cout << "您的输入有误，请重新输入！" << endl;
73. **goto** A;
74. }
75. }

##### **修改人事信息**

进入该模块后，系统会提示用户输入要修改具体哪一个人事的信息，如果查找到该人员，则系统再次让用户确认是否修改，确认后，用户需手动输入相应更新后的人员信息，最后完成修改操作。

①**流程图**



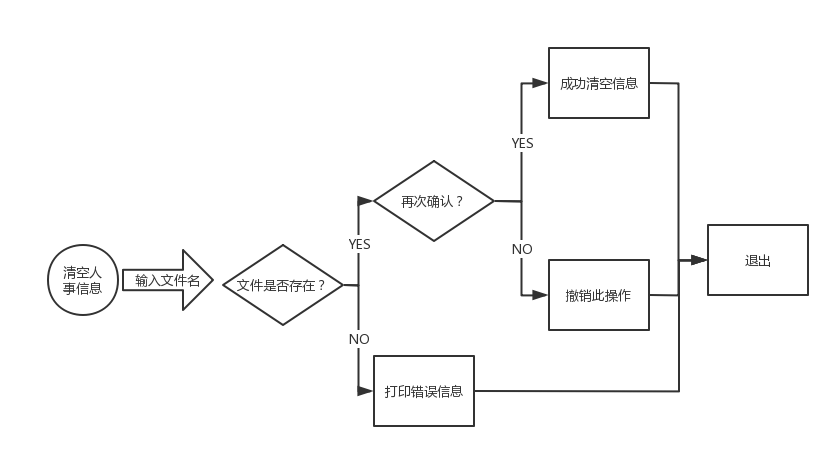
**②源码**

1. **void** mod\_info(list<Person>& mylist,string &Filename)  //修改人事信息
2. {
3. **if** (mylist.empty())
4. {
5. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
6. **return**;
7. }
8. string name;
9. **int** age;
10. string port;
11. **float** wages;
12. string addr;
13. string phone\_num;
14. **int** sex;
15. **if** (mylist.empty())
16. {
17. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
18. **return**;
19. }
20. cout << "请输入您要通过什么方式修改人事信息" << endl;
21. cout << "1、姓名" << endl;
22. cout << "2、电话" << endl;
23. string n\_name, phone;
24. **int** choice;
25. **bool** flag = **false**;
26. C:  cin >> choice;
27. **switch** (choice)
28. {
29. **case** 1:
30. {
31. cout << "请输入您要修改的人员姓名： " << endl;
32. cin >> n\_name;
33. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
34. {
35. **if** (it->get\_name() == n\_name)
36. {
37. cout << "已找到该用户，请问您是否确认将其修改：(Y/N)" << endl;
38. **char** a;
39. A:  cin >> a;
40. **if** (a == 'Y' || a == 'y')
41. {
42. flag = **true**;
43. cout << "请输入该人员的以下信息：" << endl;
44. cout << "姓名： ";
45. cin >> name;
46. cout << "年龄：" ;
47. H:  cin >> age;
48. **if** (age <= 0 || age > 150)
49. {
50. cout << "您的输入不合法，请重新输入" << endl;
51. **goto** H;
52. }
53. cout << "性别： (1-男|0-女)" ;
54. cin >> sex;
55. cout << "住址：";
56. cin >> addr;
57. cout << "职务：";
58. cin >> port;
59. cout << "工资：";
60. cin >> wages;
61. cout << "手机号：(8-11位数字)";
62. regex pattern("\\d{8,11}");
63. **int** count = 0;
64. Z:  cin >> phone\_num;
65. **if** (!regex\_match(phone\_num, pattern))
66. {
67. cout << "您的输入不合法，请重新输入！" << endl;
68. count++;
69. **if** (count < 3)
70. **goto** Z;
71. **else**
72. {
73. cout << "您已多次错误输入，已退出！" << endl;
74. **return**;
75. }
76. }
77. it->set\_name(name);
78. it->set\_addr(addr);
79. it->set\_age(age);
80. it->set\_phone(phone\_num);
81. it->set\_port(port);
82. it->set\_sex(sex);
83. it->set\_wages(wages);
84. cout << "当前用户信息已修改完毕！" << endl;
85. }
86. **else** **if** (a == 'N' || a == 'n')
87. {
88. cout << "已撤销修改操作！" << endl;
89. }
90. **else**
91. {
92. cout << "您的输入有误，请重新输入：" << endl;
93. **goto** A;
94. }
95. **break**;
96. }
97. }
98. **if** (!flag)
99. {
100. cout << "当前人事信息下未找到该人员！" << endl;
101. }
102. **break**;
103. }
104. **case** 2:
105. {
106. cout << "请输入您要修改的人员电话： " << endl;
107. cin >>phone;
108. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
109. {
110. **if** (it->get\_phone() == phone)
111. {
112. cout << "已找到该用户，请问您是否确认将其修改：(Y/N)" << endl;
113. **char** a;
114. B:  cin >> a;
115. **if** (a == 'Y' || a == 'y')
116. {
117. flag = **true**;
118. cout << "请输入该人员的以下信息：" << endl;
119. cout << "姓名： ";
120. cin >> name;
121. cout << "年龄：";
122. R:  cin >> age;
123. **if** (age <= 0 || age > 150)
124. {
125. cout << "您的输入不合法，请重新输入" << endl;
126. **goto** R;
127. }
128. cout << "性别： (1-男|0-女)";
129. cin >> sex;
130. cout << "住址：";
131. cin >> addr;
132. cout << "职务：";
133. cin >> port;
134. cout << "工资：";
135. cin >> wages;
136. cout << "手机号：(8-11位数字)";
137. regex pattern("\\d{8,11}");
138. **int** count = 0;
139. Y:  cin >> phone\_num;
140. **if** (!regex\_match(phone\_num, pattern))
141. {
142. cout << "您的输入不合法，请重新输入！" << endl;
143. count++;
144. **if** (count < 3)
145. **goto** Y;
146. **else**
147. {
148. cout << "您已多次错误输入，已退出！" << endl;
149. **return**;
150. }
151. }
152. it->set\_name(name);
153. it->set\_addr(addr);
154. it->set\_age(age);
155. it->set\_phone(phone);
156. it->set\_port(port);
157. it->set\_sex(sex);
158. it->set\_wages(wages);
159. cout << "当前用户信息已修改完毕！" << endl;
160. }
161. **else** **if** (a == 'N' || a == 'n')
162. {
163. cout << "已撤销修改操作！" << endl;
164. }
165. **else**
166. {
167. cout << "您的输入有误，请重新输入：" << endl;
168. **goto** B;
169. }
170. **break**;
171. }
172. }
173. **if** (!flag)
174. {
175. cout << "当前人事信息下未找到该人员！" << endl;
176. }
177. **break**;
178. }
179. **default**:
180. cout << "您的输入有误，请重新输入！" << endl;
181. **goto** C;
182. }
183. **if** (flag)
184. {
185. save\_info(mylist, Filename);
186. }
187. }

##### **清空人事信息**

当用户进入该模块，程序会提示用户输入要清空的人事信息文件名，若该文件存在则再次提醒用户是否确认删除，用户确认后则完成清空操作。

**①流程图**



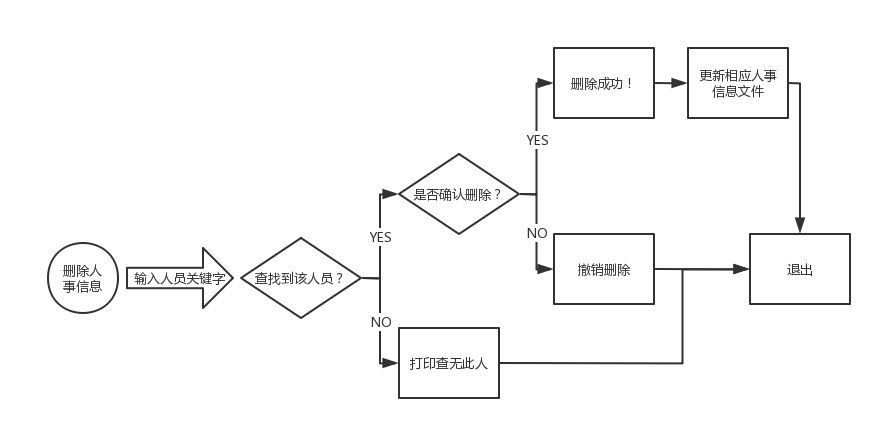
**②源码**

1. **void** clear\_info()
2. {
3. string filename;
4. cout << "请输入您要清空的人事文件名：" << endl;
5. cin >> filename;
6. filename = filename + ".txt";
7. ifstream ifs(filename, ios::in);
8. **if** (!ifs)
9. {
10. cout << "当前人事信息文件不存在，请检查后再执行本操作！" << endl;
11. ifs.close();
12. **return**;
13. }
14. cout << "请再次确认您是否要清空该人事信息文件？(Y/N)" << endl;
15. **char** choice;
16. A:  cin >> choice;
17. **if** (choice == 'Y' || choice == 'y')
18. {
19. ofstream ofs(filename, ios::trunc);
20. ofs.close();
21. cout << "当前人事信息文件已清空！" << endl;
22. }
23. **else** **if**(choice=='n'||choice=='N')
24. {
25. cout << "清空文件操作已撤销！" << endl;
26. **return**;
27. }
28. **else**
29. {
30. cout << "您的输入有误，请重新输入" << endl;
31. **goto** A;
32. }
33. }

##### **删除人事信息**

当用户进入该模块后，系统会提示用户输入要删除的人员的关键字信息，若系统查找到该用户，则会再次提醒用户是否确认删除，用户确认后则删除成功并更新相应人事信息文件。

**①流程图**



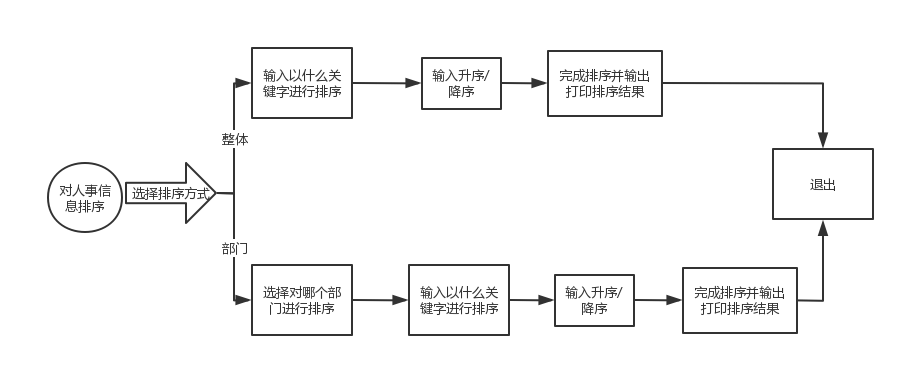
**②源码**

1. **void** del\_info(list<Person>& mylist, string &Filename)
3. **if** (mylist.empty())
4. {
5. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
6. **return**;
7. }
8. cout << "请输入您要通过什么方式删除人事信息" << endl;
9. cout << "1、姓名" << endl;
10. cout << "2、电话" << endl;
11. string name, phone;
12. **int** choice;
13. **bool** flag = **false**;
14. :   cin >> choice;
15. **switch** (choice)
16. {
17. **case** 1:
18. {
19. cout << "请输入您要删除的人员姓名： " << endl;
20. cin >> name;
21. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
22. {
23. **if** (it->get\_name() == name)
24. {
25. cout << "已找到该用户，请问您是否确认将其删除：(Y/N)" << endl;
26. **char** a;
27. A:  cin >> a;
28. **if** (a == 'Y' || a == 'y')
29. {
30. flag = **true**;
31. mylist.erase(it);
32. cout << "当前人员已删除！" << endl;
33. }
34. **else** **if** (a == 'N' || a == 'n')
35. {
36. cout << "已撤销删除操作！" << endl;
37. }
38. **else**
39. {
40. cout << "您的输入有误，请重新输入：" << endl;
41. **goto** A;
42. }
43. **break**;
44. }
45. }
46. **break**;
47. }
48. **case** 2:
49. {
50. cout << "请输入您要删除的人员电话： (8-11位数字)" << endl;
51. cin >> phone;
52. regex pattern("\\d{8,11}");
53. **int** count = 0;
54. Z:  cin >> phone;
55. **if** (!regex\_match(phone, pattern))
56. {
57. cout << "您的输入不合法，请重新输入！" << endl;
58. count++;
59. **if** (count < 3)
60. **goto** Z;
61. **else**
62. {
63. cout << "您已多次错误输入，已退出！" << endl;
64. **return**;
65. }
66. }
67. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
68. {
69. **if** (it->get\_phone() == phone)
70. {
71. cout << "已找到该用户，请问您是否确认将其删除：(Y/N)" << endl;
72. **char** a;
73. B:  cin >> a;
74. **if** (a == 'Y' || a == 'y')
75. {
76. flag = **true**;
77. mylist.erase(it);
78. cout << "当前人员已删除！" << endl;
79. }
80. **else** **if** (a == 'N' || a == 'n')
81. {
82. cout << "已撤销删除操作！" << endl;
83. }
84. **else**
85. {
86. cout << "您的输入有误，请重新输入：" << endl;
87. **goto** B;
88. }
89. **break**;
90. }
91. }
92. **break**;
93. }
94. **default**:
95. cout << "您的输入有误，请重新输入！" << endl;
96. **goto** C;
97. }
98. **if** (flag)
99. {
100. save\_info(mylist,Filename);
101. }
102. **else**
103. {
104. cout << "未在当前人事信息下查找到该人员！" << endl;
105. }

##### **对人事信息排序**

当用户进入该模块后，系统会提示用户是对整体进行排序或者是对某一个部门进行排序；当用户选择好排序方式后，程序会提示用户是对年龄或者工资进行排序，紧接着会让用户选择升序或者降序，最后完成排序操作并打印输入排序后结果。

**①流程图**



**②源码**

1. **void** all\_sort(list<Person>& mylist, PortList\*& Port)   //包括对整体排序还有部门单独排序
2. {
3. read\_count\_info(mylist, Port);
4. **if** (mylist.empty())
5. {
6. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
7. **return**;
8. }
9. cout << "请输入您要对人事信息文件整体进行排序（1），还是对某一个部门进行排序（2）？" << endl;
10. **int** count = 0;
11. **bool** flag = **true**;
12. **int** choice;
13. **while** (count < 3&&flag)
14. {
15. cin >> choice;
16. **switch** (choice)
17. {
18. **case** 1:
19. {
20. sort\_info(mylist);
21. cout << "当前人事信息整体排序结果如下： " << endl;
22. dis\_info(mylist);
23. flag = **false**;
24. system("pause");
25. system("cls");
26. **break**;
27. }
28. **case** 2:
29. {
30. PortList\* p, \* pre;
31. p = Port->next;
32. **int** sum = 1;
33. cout << "当前所有部门如下，请问您要对哪个部门进行排序：（输入部门前序号） " << endl;
34. **while** (p != NULL)
35. {
36. cout << sum++ <<"\t"<< p->port << endl;
37. p = p->next;
38. }
39. p = Port->next;
40. **int** a;
41. cin >> a;
42. **if** (a<1 || a>=sum)
43. {
44. cout << "您输入的部门不存在,请重新操作！" << endl;
45. count++;
46. **break**;
47. }
48. **for** (**int** i = 0; i < a-1; i++)
49. {
50. p = p->next;
51. }
52. **if** (p->list.empty())
53. {
54. cout << "当前人事信息为空，已退出！" << endl;
55. **return**;
56. }
57. sort\_info(p->list);
58. cout << "部门： " << p->port << " 排序结果如下： " << endl;
59. dis\_info(p->list);
60. flag = **false**;
61. system("pause");
62. system("cls");
63. **break**;
64. }
65. **default**:
66. {
67. cout << "您的输入有误，请重新输入" << endl;
68. count++;
69. system("pause");
70. system("cls");
71. }
72. }
73. }
74. **if** (count >= 3)
75. {
76. cout << "您的操作错误次数过多，已退出！" << endl;
77. **return**;
78. }
79. }

##### **展示人事信息**

当用户进入该模块后，系统会将用户打开的当前人事信息文件中的人事信息按照部门分类统计好打印输入至屏幕上。

**①流程图**



**②源码**

1. **void** dis\_count\_info(list<Person>& mylist,PortList\*& Port)  //展示所有部门分类信息
2. {
3. **if** (mylist.empty())
4. {
5. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
6. **return**;
7. }
8. read\_count\_info(mylist, Port);
9. PortList\* p, \*pre;
10. p = Port->next;
11. pre = Port;
12. cout << "当前人事信息按照职位分类如下: " << endl;
13. **while** (p != NULL)
14. {
15. cout << "职位： " << p->port << "职员总数： " << p->count << endl;
16. **for** (list<Person>::iterator it = p->list.begin(); it != p->list.end(); it++)
17. {
18. it->dis\_info();
19. cout << endl;
20. }
21. p = p->next;
22. }
23. }

## Ⅱ存储数据结构说明

### **数据结构**

在本系统中，由于人事信息管理需要进行大量的增删查改等操作，且数据之间关系为一对一关系，所以在设计逻辑结构时我使用的是线性结构；在存储结构方面本系统使用的是链式存储结构，并且为了在删除及添加等操作时更加的方便快捷，我使用的是系统STL库中的list容器来实现的，这大大方便了系统对人事信息进行管理操作。

### **数据记录创建**

本系统中，用户要创建数据记录需要手动输入人员的各种基本信息，当用户输入完毕后，系统将人员存入自定义的人员类对象中，再将每一个对象存入list容器，由此来达到输入数据记录的功能。

### **存储过程**

当用户输入完数据记录信息后，系统会遍历list容器，将list容器中的用户对象按照顺序存入到用户指定的人事信息文件中。

## Ⅲ 应用界面说明

### **总系统界面**



### **简要说明**

##### (1)打开人事信息文件

该模块用于从硬盘中读取人事信息到内存中。首先进入该模块，系统会提示用户输入要打开的人事信息名，当用户输入的文件名存在时，系统则会将人事信息读取到list容器中供用户接下来操作，若文件信息不存在，则系统会提示用户重新输入，多次输错系统会退出该模块。

##### **(2)添加人事信息**

该模块用于添加人事信息到指定文件夹中，进入该模块，系统会先提示用户需要将信息保存至哪一个文件下，若该文件不存在则会提示用户是否要创建一个新文件；接着会让用户继续输入需要添加的人事信息，最终保存到相应文件中。

##### **(3)查询人事信息**

该模块用于让用户根据指定人员姓名或者手机号来查询对应人员的详细信息。进入该模块，系统会提示用户根据姓名或者手机号来查找用户，用户选择后输入要查找的关键字，随后系统会在list容器中进行查找，查找成功则打印相关人事的详细信息，否则打印查无此人。

##### **(4)修改人事信息**

该模块用于修改人事的信息。进入该模块后，系统会提示用户输入要修改具体哪一个人事的信息，如果查找到该人员，则系统再次让用户确认是否修改，确认后，用户需手动输入相应更新后的人员信息，最后完成修改操作.

##### **(5)清空人事信息**

该模块用于清空人事信息文件的所有内容；当用户进入该模块，程序会提示用户输入要清空的人事信息文件名，若该文件存在则再次提醒用户是否确认删除，用户确认后则完成清空操作。

##### **(6)删除人事信息**

该模块用于删除人事信息；当用户进入该模块后，系统会提示用户输入要删除的人员的关键字信息，若系统查找到该用户，则会再次提醒用户是否确认删除，用户确认后则删除成功并更新相应人事信息文件。

##### **(7)对人事信息排序**

该模块用于对人事信息进行排序；当用户进入该模块后，系统会提示用户是对整体进行排序或者是对某一个部门进行排序；当用户选择对整体排序后，系统会直接对list容器调用其内指定sort算法并按照年龄或者工资来对人事信息排序；若用户选择对部门进行排序，则系统会将所有部门展示出来并提示用户要对哪一个部门进行排序，最后完成排序要求。

##### **(8)展示人事信息**

该模块用于展示人事信息；当用户进入该模块后，系统会将用户打开的当前人事信息文件中的人事信息按照部门分类统计好打印输入至屏幕上。

### **主要函数**

1. **void read\_info(list<Person>& mylist, PortList\*& Port,string &Filename);**

//该函数用于打开一个人事信息文件并将文件内的所有人事信息保存至mylist容器中，并且将按照部门分类好后的人事信息保存至Port结构体中供用户操作。

1. **void save\_info(list<Person>& mylist,string& Filename);**

//该函数用于将mylist容器中的所有人员信息保存至Filename文件中，用户需提前输入要保存至哪一个Filename文件中

1. **bool cmp\_age\_de(Person& p1, Person& p2);**

//此函数为对list容器排序时的判断依据函数，此函数实现了对list容器的年龄进行降序排列操作

1. **void del\_info(list<Person>& mylist, string& Filename);**

//此函数实现了对mylist容器中的人员信息进行删除操作，当删除操作进行成功后，会将更新后的mylist保存至Filename文件中

1. **void dis\_info(list<Person>& mylist);**

//此函数用于展示未进行分类时的所有人事信息

1. **void input\_info();**

//此函数用于输入人员信息，在函数中需手动输入要保存至什么文件名下，当用户输入完毕后，系统会调用save\_info函数进行数据保存

1. **void search\_info(list<Person>& mylist);**

//此函数用于查询人员信息，进入该模块，系统会提示用户根据姓名或者手机号来查找用户，用户选择后输入要查找的关键字，随后系统会在mylist容器中进行查找，查找成功则打印相关人事的详细信息，否则打印查无此人。

1. **void mod\_info(list<Person>& mylist, string& Filename);**

**//**此函数用于修改人员信息，系统会提示用户输入要修改具体哪一个人事的信息，如果查找到该人员，则系统再次让用户确认是否修改，确认后，用户需手动输入相应更新后的人员信息，最后将修改完成后的mylist保存至Filename文件下.

**(9)void read\_count\_info(list<Person>& mylist, PortList\* &Port);**

//此函数用于实现将mylist中的所有人员信息按照部门进行分类，并将分类后的结果保存至Port结构体中以供用户进行查看等操作。

**(10)void dis\_count\_info(list<Person>& mylist ,PortList\*& Port);**

//此函数用于将按照部门进行分类后的人员信息打印输出至屏幕上

**(11)void all\_sort(list<Person>& mylist, PortList\* &Port);**

//此函数用于对人员信息进行排序操作，进入此函数后，系统会提示用户是对整体进行排序或者是对某一个部门进行排序；当用户选择对整体排序后，系统会直接对list容器调用其内指定sort算法并按照年龄或者工资来对人事信息排序；若用户选择对部门进行排序，则系统会将所有部门展示出来并提示用户要对哪一个部门进行排序，最后完成排序要求。

1. **void clear\_info();**

//此函数用于清空人事信息文件

1. **void dis\_main\_menu();**

//此函数用于展示系统主菜单

1. **void func\_main(list<Person>& mylist, PortList\*& Port);**

//此函数用于实现系统主功能选择

### **核心源码**

1. **void read\_info(list<Person>& mylist, PortList\*& Port,string &Filename);**
2. **void** read\_info(list<Person>& mylist,PortList\*& Port ,string& Filename)
4. mylist.clear();
5. string name;
6. **int** age;
7. string port;
8. **float** wages;
9. string addr;
10. string phone\_num;
11. **int** file\_error\_count = 0;
12. **int** sex;
13. Person person;
14. cout << "请输入您要打开的人事信息文件：" << endl;
15. cin >> Filename;
16. Filename = Filename + ".txt";
17. ifstream ifs(Filename, ios::in);
18. **while** (file\_error\_count < 3)
19. {
20. **if** (!ifs)
21. {
22. cout << "您输入的人事管理信息文件不存在，请检查后重新输入..." << endl;
23. file\_error\_count++;
24. cout << "您现在还有 " << (3 - file\_error\_count) << "次输入机会" << endl;
25. **if** (file\_error\_count < 3)
26. {
27. cout << "请输入您要打开的人事信息文件：" << endl;
28. cin >> Filename;
29. Filename = Filename + ".txt";
30. ifstream ifs(Filename, ios::in);
31. }
32. }
33. **else**
34. {
35. **break**;
36. }
37. }
38. **if** (file\_error\_count == 3)
39. {
40. cout << "输入错误次数过多，已自动退出！" << endl;
41. **return**;
42. }
43. **int** count = 0;
44. **while** (!ifs.eof())
45. {
46. ifs >> name >> age >>sex>> addr >> port >> wages >> phone\_num;
47. **if** (age < 0)
48. {
49. **continue**;
50. }
51. person.set\_name(name);
52. person.set\_age(age);
53. person.set\_sex(sex);
54. person.set\_phone(phone\_num);
55. person.set\_port(port);
56. person.set\_wages(wages);
57. person.set\_addr(addr);
58. mylist.push\_front(person);
59. count++;
60. }
61. ifs.close();
62. read\_count\_info(mylist, Port);
63. }
64. **void save\_info(list<Person>& mylist,string& Filename);**
65. **void** save\_info(list<Person>& mylist,string &Filename)  //清空文件方式保存信息
66. {
67. ofstream ofs(Filename, ios::trunc);
68. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end();)
69. {
70. ofs << it->get\_name() << " " << it->get\_age() <<" "<<it->get\_sex()<< " " << it->get\_addr() << " " << it->get\_port() << " " << it->get\_wages() << " " << it->get\_phone();
71. **if** (++it != mylist.end())
72. {
73. ofs << endl;
74. }
75. }
76. ofs.close();
77. }

**(3)bool cmp\_age\_de(Person& p1, Person& p2);**

1. **bool** cmp\_age\_de(Person &p1,Person &p2)
2. {
3. **if** (p1.get\_age() == p2.get\_age())
4. {
5. **return** p1.get\_wages ()> p1.get\_wages();  //按照工资降序
6. }
7. **else**
8. {
9. **return** p1.get\_age() > p2.get\_age();    //按照年龄降序
10. }
11. }

**(4)void del\_info(list<Person>& mylist, string& Filename);**

1. **void** del\_info(list<Person>& mylist, string &Filename)
3. **if** (mylist.empty())
4. {
5. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
6. **return**;
7. }
8. cout << "请输入您要通过什么方式删除人事信息" << endl;
9. cout << "1、姓名" << endl;
10. cout << "2、电话" << endl;
11. string name, phone;
12. **int** choice;
13. **bool** flag = **false**;
14. :   cin >> choice;
15. **switch** (choice)
16. {
17. **case** 1:
18. {
19. cout << "请输入您要删除的人员姓名： " << endl;
20. cin >> name;
21. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
22. {
23. **if** (it->get\_name() == name)
24. {
25. cout << "已找到该用户，请问您是否确认将其删除：(Y/N)" << endl;
26. **char** a;
27. A:  cin >> a;
28. **if** (a == 'Y' || a == 'y')
29. {
30. flag = **true**;
31. mylist.erase(it);
32. cout << "当前人员已删除！" << endl;
33. }
34. **else** **if** (a == 'N' || a == 'n')
35. {
36. cout << "已撤销删除操作！" << endl;
37. }
38. **else**
39. {
40. cout << "您的输入有误，请重新输入：" << endl;
41. **goto** A;
42. }
43. **break**;
44. }
45. }
46. **break**;
47. }
48. **case** 2:
49. {
50. cout << "请输入您要删除的人员电话： (8-11位数字)" << endl;
51. cin >> phone;
52. regex pattern("\\d{8,11}");
53. **int** count = 0;
54. Z:  cin >> phone;
55. **if** (!regex\_match(phone, pattern))
56. {
57. cout << "您的输入不合法，请重新输入！" << endl;
58. count++;
59. **if** (count < 3)
60. **goto** Z;
61. **else**
62. {
63. cout << "您已多次错误输入，已退出！" << endl;
64. **return**;
65. }
66. }
67. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
68. {
69. **if** (it->get\_phone() == phone)
70. {
71. cout << "已找到该用户，请问您是否确认将其删除：(Y/N)" << endl;
72. **char** a;
73. B:  cin >> a;
74. **if** (a == 'Y' || a == 'y')
75. {
76. flag = **true**;
77. mylist.erase(it);
78. cout << "当前人员已删除！" << endl;
79. }
80. **else** **if** (a == 'N' || a == 'n')
81. {
82. cout << "已撤销删除操作！" << endl;
83. }
84. **else**
85. {
86. cout << "您的输入有误，请重新输入：" << endl;
87. **goto** B;
88. }
89. **break**;
90. }
91. }
92. **break**;
93. }
94. **default**:
95. cout << "您的输入有误，请重新输入！" << endl;
96. **goto** C;
97. }
98. **if** (flag)
99. {
100. save\_info(mylist,Filename);
101. }
102. **else**
103. {
104. cout << "未在当前人事信息下查找到该人员！" << endl;
105. }

**(5)void dis\_info(list<Person>& mylist);**

1. **void** dis\_info(list<Person>& mylist)   //展示未分类的人事信息
3. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
4. {
5. it->dis\_info();
6. cout << endl;
7. }

**(6)void input\_info();**

1. **void** input\_info()
2. {
3. string Filenamne;
4. list<Person> mylist;
5. PortList \*Port;
6. cout << "请输入您要添加人事信息到哪个文件名中： " << endl;
7. cin >> Filenamne;
8. Filenamne = Filenamne + ".txt";
9. ifstream ifs(Filenamne);
10. **if** (!ifs)
11. {
12. cout << "当前文件不存在，请问您是否要创建新文件？(Y/N)" << endl;
13. **char** c;
14. K:  cin >> c;
15. **if**(c == 'N' || c == 'n')
16. {
17. cout << "添加人事信息失败！" << endl;
18. **return**;
19. }
20. **else**  **if** (c == 'Y' || c == 'y')
21. {}
22. **else**
23. {
24. cout << "您的输入有误，请重新输入：" << endl;
25. **goto** K;
26. }
27. }
28. **else**
29. {
30. read\_info\_temp(mylist, Filenamne);
31. }
32. string name;
33. **int** age;
34. string port;
35. **float** wages;
36. string addr;
37. string phone\_num;
38. Person person;
39. **int** sex;
40. **while** (1)
41. {
42. cout << "请输入该人员的以下信息：" << endl;
43. cout << "姓名： ";
44. cin >> name;
45. cout << "年龄：";
46. H:  cin >> age;
47. **if** (age <= 0 || age > 150)
48. {
49. cout << "您的输入不合法，请重新输入" << endl;
50. **goto** H;
51. }
52. cout << "性别： (1-男|0-女)";
53. cin >> sex;
54. cout << "住址：";
55. cin >> addr;
56. cout << "职务：";
57. cin >> port;
58. cout << "工资：";
59. cin >> wages;
60. cout << "手机号：(8-11位数字)";
61. regex pattern("\\d{8,11}");
62. **int** count = 0;
63. Z:  cin >> phone\_num;
64. **if** (!regex\_match(phone\_num, pattern))
65. {
66. cout << "您的输入不合法，请重新输入！" << endl;
67. count++;
68. **if** (count < 3)
69. **goto** Z;
70. **else**
71. {
72. cout << "您已多次错误输入，已退出！" << endl;
73. **return**;
74. }
75. }
76. person.set\_name(name);
77. person.set\_addr(addr);
78. person.set\_age(age);
79. person.set\_phone(phone\_num);
80. person.set\_port(port);
81. person.set\_sex(sex);
82. person.set\_wages(wages);
83. mylist.push\_back(person);
84. cout << "请问您是否还要继续输入人事信息？(Y/N)" << endl;
85. **char** choice;
86. A:  cin >> choice;
87. **if** (choice == 'Y' || choice == 'y')
88. {
89. **continue**;
90. }
91. **else** **if** (choice == 'N' || choice == 'n')
92. {
93. cout << "当前人事信息已输入完毕" << endl;
94. **break**;
95. }
96. **else**
97. {
98. cout << "您的输入有误，请重新输入" << endl;
99. **goto** A;
100. }
101. }
102. save\_info(mylist, Filenamne);
103. }

**(7)void search\_info(list<Person>& mylist);**

1. **void** search\_info(list<Person>& mylist)
2. {
3. **if** (mylist.empty())
4. {
5. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
6. **return**;
7. }
8. cout << "请输入您要通过什么方式查找人事信息" << endl;
9. cout << "1、姓名" << endl;
10. cout << "2、电话" << endl;
11. string name, phone;
12. **int** choice;
13. **bool** flag = **false**;
14. A:  cin >> choice;
15. **switch** (choice)
16. {
17. **case** 1:
18. {
19. cout << "请输入您要查找的人员姓名： " << endl;
20. cin >> name;
21. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
22. {
23. **if** (it->get\_name() == name)
24. {
25. cout << "已找到该用户，该用户详细信息如下：" << endl;
26. it->dis\_info();
27. flag = **true**;
28. **break**;
29. }
30. }
31. **if** (!flag)
32. {
33. cout << "当前人事信息文件中未查找到姓名为： "<<name<<"的用户！" << endl;
34. }
35. **break**;
36. }
37. **case** 2:
38. {
39. cout << "请输入您要查找的人员电话：(8-11位数字) " << endl;
40. regex pattern("\\d{8,11}");
41. **int** count = 0;
42. Z:  cin >> phone;
43. **if** (!regex\_match(phone, pattern))
44. {
45. cout << "您的输入不合法，请重新输入！" << endl;
46. count++;
47. **if**(count<3)
48. **goto** Z;
49. **else**
50. {
51. cout << "您已多次错误输入，已退出！" << endl;
52. **return**;
53. }
54. }
55. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
56. {
57. **if** (it->get\_phone() == phone)
58. {
59. cout << "已找到该用户，该用户详细信息如下：" << endl;
60. it->dis\_info();
61. flag = **true**;
62. **break**;
63. }
64. }
65. **if** (!flag)
66. {
67. cout << "当前人事信息文件中未查找到手机号为： " << phone << "的用户！" << endl;
68. }
69. **break**;
70. }
71. **default**:
72. cout << "您的输入有误，请重新输入！" << endl;
73. **goto** A;
74. }
75. }

**(8)void mod\_info(list<Person>& mylist, string& Filename);**

1. **void** mod\_info(list<Person>& mylist,string &Filename)  //修改人事信息
2. {
3. **if** (mylist.empty())
4. {
5. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
6. **return**;
7. }
8. string name;
9. **int** age;
10. string port;
11. **float** wages;
12. string addr;
13. string phone\_num;
14. **int** sex;
15. **if** (mylist.empty())
16. {
17. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
18. **return**;
19. }
20. cout << "请输入您要通过什么方式修改人事信息" << endl;
21. cout << "1、姓名" << endl;
22. cout << "2、电话" << endl;
23. string n\_name, phone;
24. **int** choice;
25. **bool** flag = **false**;
26. C:  cin >> choice;
27. **switch** (choice)
28. {
29. **case** 1:
30. {
31. cout << "请输入您要修改的人员姓名： " << endl;
32. cin >> n\_name;
33. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
34. {
35. **if** (it->get\_name() == n\_name)
36. {
37. cout << "已找到该用户，请问您是否确认将其修改：(Y/N)" << endl;
38. **char** a;
39. A:  cin >> a;
40. **if** (a == 'Y' || a == 'y')
41. {
42. flag = **true**;
43. cout << "请输入该人员的以下信息：" << endl;
44. cout << "姓名： ";
45. cin >> name;
46. cout << "年龄：" ;
47. H:  cin >> age;
48. **if** (age <= 0 || age > 150)
49. {
50. cout << "您的输入不合法，请重新输入" << endl;
51. **goto** H;
52. }
53. cout << "性别： (1-男|0-女)" ;
54. cin >> sex;
55. cout << "住址：";
56. cin >> addr;
57. cout << "职务：";
58. cin >> port;
59. cout << "工资：";
60. cin >> wages;
61. cout << "手机号：(8-11位数字)";
62. regex pattern("\\d{8,11}");
63. **int** count = 0;
64. Z:  cin >> phone\_num;
65. **if** (!regex\_match(phone\_num, pattern))
66. {
67. cout << "您的输入不合法，请重新输入！" << endl;
68. count++;
69. **if** (count < 3)
70. **goto** Z;
71. **else**
72. {
73. cout << "您已多次错误输入，已退出！" << endl;
74. **return**;
75. }
76. }
77. it->set\_name(name);
78. it->set\_addr(addr);
79. it->set\_age(age);
80. it->set\_phone(phone\_num);
81. it->set\_port(port);
82. it->set\_sex(sex);
83. it->set\_wages(wages);
84. cout << "当前用户信息已修改完毕！" << endl;
85. }
86. **else** **if** (a == 'N' || a == 'n')
87. {
88. cout << "已撤销修改操作！" << endl;
89. }
90. **else**
91. {
92. cout << "您的输入有误，请重新输入：" << endl;
93. **goto** A;
94. }
95. **break**;
96. }
97. }
98. **if** (!flag)
99. {
100. cout << "当前人事信息下未找到该人员！" << endl;
101. }
102. **break**;
103. }
104. **case** 2:
105. {
106. cout << "请输入您要修改的人员电话： " << endl;
107. cin >>phone;
108. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
109. {
110. **if** (it->get\_phone() == phone)
111. {
112. cout << "已找到该用户，请问您是否确认将其修改：(Y/N)" << endl;
113. **char** a;
114. B:  cin >> a;
115. **if** (a == 'Y' || a == 'y')
116. {
117. flag = **true**;
118. cout << "请输入该人员的以下信息：" << endl;
119. cout << "姓名： ";
120. cin >> name;
121. cout << "年龄：";
122. R:  cin >> age;
123. **if** (age <= 0 || age > 150)
124. {
125. cout << "您的输入不合法，请重新输入" << endl;
126. **goto** R;
127. }
128. cout << "性别： (1-男|0-女)";
129. cin >> sex;
130. cout << "住址：";
131. cin >> addr;
132. cout << "职务：";
133. cin >> port;
134. cout << "工资：";
135. cin >> wages;
136. cout << "手机号：(8-11位数字)";
137. regex pattern("\\d{8,11}");
138. **int** count = 0;
139. Y:  cin >> phone\_num;
140. **if** (!regex\_match(phone\_num, pattern))
141. {
142. cout << "您的输入不合法，请重新输入！" << endl;
143. count++;
144. **if** (count < 3)
145. **goto** Y;
146. **else**
147. {
148. cout << "您已多次错误输入，已退出！" << endl;
149. **return**;
150. }
151. }
152. it->set\_name(name);
153. it->set\_addr(addr);
154. it->set\_age(age);
155. it->set\_phone(phone);
156. it->set\_port(port);
157. it->set\_sex(sex);
158. it->set\_wages(wages);
159. cout << "当前用户信息已修改完毕！" << endl;
160. }
161. **else** **if** (a == 'N' || a == 'n')
162. {
163. cout << "已撤销修改操作！" << endl;
164. }
165. **else**
166. {
167. cout << "您的输入有误，请重新输入：" << endl;
168. **goto** B;
169. }
170. **break**;
171. }
172. }
173. **if** (!flag)
174. {
175. cout << "当前人事信息下未找到该人员！" << endl;
176. }
177. **break**;
178. }
179. **default**:
180. cout << "您的输入有误，请重新输入！" << endl;
181. **goto** C;
182. }
183. **if** (flag)
184. {
185. save\_info(mylist, Filename);
186. }
187. }

**(9)void read\_count\_info(list<Person>& mylist, PortList\* &Port);**

1. **void** read\_count\_info(list<Person>& mylist, PortList\*&Port)
3. PortList \*p,\*p1,\*pre;
4. Port = **new** PortList;
5. Port->port = "";
6. Port->next = NULL;
7. Port->count = 0;
8. p = Port->next;
9. p1 =pre= p;
10. **bool** flag = **false**;
11. **for** (list<Person>::iterator it = mylist.begin(); it != mylist.end(); it++)
12. {
13. flag = **false**;
14. **if** (p == NULL)
15. {
16. p = **new** PortList;
17. p->port = it->get\_port();
18. p->count=1;
19. p->list.push\_back(\*it);
20. p->next = NULL;
21. Port->next = p;
22. **continue**;
23. }
24. p1 = p;
25. **while** (p1!= NULL)
26. {
27. **if** (p1->port == it->get\_port())
28. {
29. p1->count++;
30. flag = **true**;
31. p1->list.push\_back(\*it);
32. }
33. pre = p1;
34. p1 = p1->next;
35. }
36. **if** (!flag)
37. {
38. p1 = **new** PortList;
39. p1->port = it->get\_port();
40. p1->count = 1;
41. p1->next = NULL;
42. p1->list.push\_back(\*it);
43. pre->next = p1;
44. }
45. }

**(10)void dis\_count\_info(list<Person>& mylist ,PortList\*& Port);**

1. **void** dis\_count\_info(list<Person>& mylist,PortList\*& Port)  //展示所有部门分类信息
2. {
3. **if** (mylist.empty())
4. {
5. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
6. **return**;
7. }
8. read\_count\_info(mylist, Port);
9. PortList\* p, \*pre;
10. p = Port->next;
11. pre = Port;
12. cout << "当前人事信息按照职位分类如下: " << endl;
13. **while** (p != NULL)
14. {
15. cout << "职位： " << p->port << "职员总数： " << p->count << endl;
16. **for** (list<Person>::iterator it = p->list.begin(); it != p->list.end(); it++)
17. {
18. it->dis\_info();
19. cout << endl;
20. }
21. p = p->next;
22. }
23. }

**(11)void all\_sort(list<Person>& mylist, PortList\* &Port);**

1. **void** all\_sort(list<Person>& mylist, PortList\*& Port)   //包括对整体排序还有部门单独排序
2. {
3. read\_count\_info(mylist, Port);
4. **if** (mylist.empty())
5. {
6. cout << "当前人事信息文件为空或未打开文件，已退出！" << endl;
7. **return**;
8. }
9. cout << "请输入您要对人事信息文件整体进行排序（1），还是对某一个部门进行排序（2）？" << endl;
10. **int** count = 0;
11. **bool** flag = **true**;
12. **int** choice;
13. **while** (count < 3&&flag)
14. {
15. cin >> choice;
16. **switch** (choice)
17. {
18. **case** 1:
19. {
20. sort\_info(mylist);
21. cout << "当前人事信息整体排序结果如下： " << endl;
22. dis\_info(mylist);
23. flag = **false**;
24. system("pause");
25. system("cls");
26. **break**;
27. }
28. **case** 2:
29. {
30. PortList\* p, \* pre;
31. p = Port->next;
32. **int** sum = 1;
33. cout << "当前所有部门如下，请问您要对哪个部门进行排序：（输入部门前序号） " << endl;
34. **while** (p != NULL)
35. {
36. cout << sum++ <<"\t"<< p->port << endl;
37. p = p->next;
38. }
39. p = Port->next;
40. **int** a;
41. cin >> a;
42. **if** (a<1 || a>=sum)
43. {
44. cout << "您输入的部门不存在,请重新操作！" << endl;
45. count++;
46. **break**;
47. }
48. **for** (**int** i = 0; i < a-1; i++)
49. {
50. p = p->next;
51. }
52. **if** (p->list.empty())
53. {
54. cout << "当前人事信息为空，已退出！" << endl;
55. **return**;
56. }
57. sort\_info(p->list);
58. cout << "部门： " << p->port << " 排序结果如下： " << endl;
59. dis\_info(p->list);
60. flag = **false**;
61. system("pause");
62. system("cls");
63. **break**;
64. }
65. **default**:
66. {
67. cout << "您的输入有误，请重新输入" << endl;
68. count++;
69. system("pause");
70. system("cls");
71. }
72. }
73. }
74. **if** (count >= 3)
75. {
76. cout << "您的操作错误次数过多，已退出！" << endl;
77. **return**;
78. }
79. }

**(12)void clear\_info();**

1. **void** clear\_info()
2. {
3. string filename;
4. cout << "请输入您要清空的人事文件名：" << endl;
5. cin >> filename;
6. filename = filename + ".txt";
7. ifstream ifs(filename, ios::in);
8. **if** (!ifs)
9. {
10. cout << "当前人事信息文件不存在，请检查后再执行本操作！" << endl;
11. ifs.close();
12. **return**;
13. }
14. cout << "请再次确认您是否要清空该人事信息文件？(Y/N)" << endl;
15. **char** choice;
16. A:  cin >> choice;
17. **if** (choice == 'Y' || choice == 'y')
18. {
19. ofstream ofs(filename, ios::trunc);
20. ofs.close();
21. cout << "当前人事信息文件已清空！" << endl;
22. }
23. **else** **if**(choice=='n'||choice=='N')
24. {
25. cout << "清空文件操作已撤销！" << endl;
26. **return**;
27. }
28. **else**
29. {
30. cout << "您的输入有误，请重新输入" << endl;
31. **goto** A;
32. }
33. }

**(13)void dis\_main\_menu();**

1. **void** dis\_main\_menu()
2. {
3. cout << endl << endl << endl << endl;
4. cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*人事信息管理系统\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
5. cout << "\t\t\t\*\*\*\t1、打开人事信息文件\t\t      2、添加人事信息\t  \*\*\*" << endl;
6. cout << "\t\t\t\*\*\*\t3、查询人事信息\t\t\t      4、删除人事信息  \t  \*\*\*" << endl;
7. cout << "\t\t\t\*\*\*\t5、修改人事信息\t\t\t      6、清空人事信息     \*\*\*" << endl;
8. cout << "\t\t\t\*\*\*\t7、展示人事信息\t\t\t      8、排序人事信息     \*\*\*" << endl;
9. cout << "\t\t\t\*\*\*\t0、退出系统\t\t\t      \t\t\t  \*\*\*" << endl;
10. cout << "\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;
11. }

**(14)void func\_main(list<Person>& mylist, PortList\*& Port);**

1. **void** func\_main(list<Person>& mylist, PortList\*& Port)
3. string Filename;
4. **while** (1)
5. {
6. dis\_main\_menu();
7. cout << "请输入您的选择" << endl;
8. **int** choice;
9. cin >> choice;
10. **switch** (choice)
11. {
12. **case** 1:
13. read\_info(mylist,Port, Filename);
14. cout << "当前人事信息文件已打开，请继续操作！" << endl;
15. system("pause");
16. system("cls");
17. **break**;
18. **case** 2:
19. input\_info();
20. system("pause");
21. system("cls");
22. **break**;
23. **case** 3:
24. search\_info(mylist);
25. system("pause");
26. system("cls");
27. **break**;
28. **case** 4:
29. del\_info(mylist, Filename);
30. system("pause");
31. system("cls");
32. **break**;
33. **case** 5:
34. mod\_info(mylist, Filename);
35. system("pause");
36. system("cls");
37. **break**;
38. **case** 6:
39. clear\_info();
40. system("pause");
41. system("cls");
42. **break**;
43. **case** 7:
44. dis\_count\_info(mylist,Port);
45. system("pause");
46. system("cls");
47. **break**;
48. **case** 8:
49. {
50. all\_sort(mylist, Port);
51. system("pause");
52. system("cls");
53. **break**;
54. }
55. **case** 0:
56. system("cls");
57. cout << "欢迎下次使用本系统！" << endl;
58. system("pause");
59. system("cls");
60. **return**;
61. **default**:
62. cout << "您的输入有误，请重新输入！" << endl;
63. system("pause");
64. system("cls");
65. **break**;
66. }
67. }

## Ⅳ 遗留问题

在本次我设计的人事信息管理系统中，对于人员的信息存储不够详细，应再多添加些人员信息，比如身份证号，毕业大学等，这些能更加充实我的管理系统；并且在进行一些敏感操作时，验证力度并不大，应在多增加一些验证力度，比如添加一些管理员账户来专门进行一些敏感行为操作。

## Ⅴ 设计体会

经过此次课设设计，我认识到了学习C++要切忌浮躁，要明白欲速则不达。如果想要学好C++就要多看多写，而最关键的是要上课听老师分析。

通过一个学期的学习，我更了解计算机编程了，同时也更喜欢计算机了。并且我也知道了如何学好C++，学C++靠的不是上课的那点儿时间，而是课前的预习工作及课后的复习工作，还有对写程序的及时练习，练习写程序不仅是对知识的复习也是对知识的熟练从而也能更深刻的理解程序，理解C++。我们学习C++不仅要看课本，更重要的是看学习资料，从而使掌握的知识更广阔。只有拥有丰富的知识以及掌握对编程的熟练度，才能对程序灵活书写。

C++是比C语言更高级的一门语言，我们也可以通过学习C语言来学习C++，这样就可以减少学习中遇到的困难。想要学好C++，让我们一起努力吧。

# **参考文献**

**［1］郑莉、傅仕星编，《C++面向对象程序设计》，清华大学出版社，2003年9月出版**

**［2］谭浩强编，《C++面向对象程序设计》，清华大学出版社，2005年7月出版**

**［3］李师贤等译，《C++精髓》，机械工业出版社，2002年8月出版**

**［4］Davis Chapman，《学用Visual c++ 6.0》，清华大学出版社，2001年7月出版 ［5］郑莉、傅仕星编，C++语言面向对象程序设计习题与实验指导，清华大学出版社，2003年9月出版**

**［5］杜茂康等, C++.NET程序设计,清华大学出版社，2009年5月第一版**