【实验7-5】 微信投票

### 【任务介绍】

**1.任务描述**

如今微信聊天已经普及到几乎每一个人，在聊天中，经常会有人需要帮忙在某个APP中投票。本案例要求编写一个模拟微信投票的程序，通过在控制台输入指令，实现添加候选人、查看当前投票和投票的功能。每个功能的具体要求如下：

1. 用户输入指令1代表”添加候选人”，可以在本地文件中添加被选举人。
2. 用户输入指令2代表”查看当前投票”，将本地文件中的数据打印到控制台。
3. 用户输入指令3 代表”投票”功能，在控制台输入被投票人的名字进行投票操作。
4. 用户输入指令4代表”退出”操作。

**2.运行结果**

任务运行结果如图7-1所示。

手机屏幕截图

描述已自动生成

图7-1 运行结果

运行过程中，本地D：\下会生成一个count.txt文件，打开后如图7-2所示。

社交网络的手机截图

描述已自动生成

图7-2 count.txt

### 【任务目标】

* 学会分析”投票小功能”任务实现的逻辑思路。
* 能够独立完成” 投票小功能”程序的源代码编写、编译以及运行。
* 能够利用字符流操作本地的方法。
* 掌握StringBuffer和数组的使用。

### 【实现思路】

1. 查看任务介绍和运行结果分析可知，此任务需要使用while循环实现控制台中操作指令的多次输入，使用switch语句根据控制台输入的操作指令来判断执行什么操作。
2. 输入指令1时进行添加候选人操作，先从控制台获取输入的被投票人与起始票数并暂存，再判断是否已有count.txt文件存储数据，如果有直接在文件后追加新的数据，如果没有需要新建文件夹在存入数据。这里需要使用到StringBuffer的字符串拼接来实现存入count.txt文件的特定格式，例如：“姓名：票数,”，这样方便我们读取和修改票数。
3. 输入指令2时查看当前投票，使用字符流读取count.txt文件并在控制台打印即可。
4. 输入指令3时进行投票操作，从控制台输入被投票人姓名，投票成功后，被投票人的票数加一，先取出count.txt的内容存入String类型中，根据“，”拆分为String数组（editMessage），再获取控制类中输入的被投票人姓名，使用for循环判断是否存在此人的投票，如果存在，则取出此人的票数加一，最后将修改后的数据使用StringBuffer替换到String数组（editMessage）并覆盖原有的count.txt文件
5. 输入指令4，直接退出系统，可以通过代码”System.exit（0）;”实现。

### 【实现代码】

投票小功能的代码实现如文件7-1所示。

文件7-1 vote.java

1. package chapter0704;
2. import java.io.BufferedOutputStream;
3. import java.io.File;
4. import java.io.FileInputStream;
5. import java.io.FileNotFoundException;
6. import java.io.FileOutputStream;
7. import java.io.FileWriter;
8. import java.util.Scanner;
9. public class vote {
10. private static String message = "";
11. public static void main(String[] args) throws Exception {
12. Scanner sc = new Scanner(System.in);
13. System.out.println("1:添加候选人 2:查看当前投票 3:投票 4:退出");
14. while (true) {
15. System.out.print("请输入操作指令：");
16. int command = sc.nextInt();
17. switch (command) {
18. case 1:
19. System.out.println("请输入需要被投票的人：");
20. String name=sc.next();
21. System.out.println("请输入"+name+"的起始票数：");
22. int num=sc.nextInt();
23. addvote(name,num);// 1:添加投票人
24. break;
25. case 2:
26. String me = readvote();// 2:查看投票
27. System.out.println(me);
28. break;
29. case 3:
30. editvote();// 3:投票
31. break;
32. case 4:
33. System.out.println("您已退出系统，谢谢使用！");
34. System.exit(0);
35. break;
36. default:
37. System.out.println("您输入的指令错误！");
38. break;
39. }
40. }
41. }
42. /\*
43. \* 查看投票
44. \*/
45. public static String readvote() throws Exception{
46. FileInputStream in = new FileInputStream("D:\\count.txt");
47. byte[] b = new byte[in.available()];
48. in.read(b);
49. message =new String(b);
50. in.close();
51. return message;
52. }
53. /\*
54. \* 添加投票
55. \*/
56. public static void addvote(String name,int num) throws
57. FileNotFoundException {
58. String SEPARATE\_FIELD = "\n";// 换行
59. BufferedOutputStream out = null;
60. StringBuffer sbf = new StringBuffer();// 拼接内容
61. File file=new File("D:\\count.txt"); //判断文件是否存在
62. try {
63. // 当已存在count.txt文件，则在文件内容后追加
64. if (file.canExecute()) {
65. // 创建输出流，用于追加文件
66. out = new BufferedOutputStream(new
67. FileOutputStream("D:\\count.txt", true));
68. } else {// 不存在当天文件，则新建文件
69. // 创建输出流，用于保存文件
70. out = new BufferedOutputStream(new
71. FileOutputStream("D:\\count.txt"));
72. }
73. sbf.append(name).append(":");
74. sbf.append(num).append(","+SEPARATE\_FIELD);
75. String str = sbf.toString();
76. byte[] b = str.getBytes();
77. for (int i = 0; i < b.length; i++) {
78. out.write(b[i]);// 将内容写入本地文件
79. }
80. } catch (Exception e) {
81. e.printStackTrace();
82. } finally {
83. try {
84. if (out != null)
85. out.close();// 关闭输出流
86. } catch (Exception e2) {
87. e2.printStackTrace();
88. }
89. }
90. }
91. /\*
92. \* 投票 将控制台输入的姓名的票数加一后保存
93. \*/
94. private static void editvote() throws Exception {
95. FileWriter out = null;
96. // 将count.txt文字根据“，”拆分成数组
97. String[] editMessage = message.split(",");
98. Scanner sc = new Scanner(System.in);
99. System.out.println("请输入要投票人的姓名");
100. String inputMessage = sc.next();
101. //for循环拆分后的数组
102. for (int i = 0; i < editMessage.length; i++) {
103. //当数组中有有包含输入的名字时
104. if(editMessage[i].contains(inputMessage)) {
105. //取出输入名字现在的票数
106. String a =
107. editMessage[i].substring(editMessage[i].indexOf(":")+1,
108. editMessage[i].length());
109. //将取出的票数强转为int类型
110. int b = Integer.parseInt(a);
111. //将票数+1
112. b++;
113. //new出一个StringBuffer用于后面的字符拼接
114. StringBuffer sb = new StringBuffer();
115. //取出原文件中的姓名和：
116. String c=editMessage[i].substring(0,editMessage[i].indexOf(":")+1);
117. //使用StringBuffer拼接姓名和选票
118. sb.append(c);
119. sb.append(b);
120. sb.append(",");
121. //将拼接后的字符强转为String类型
122. String s = sb.toString();
123. //修改好的字符替换原有的字符
124. System.out.println("投票后的票数：");
125. System.out.println(s);
126. editMessage[i] = s;
127. }
128. }
129. //editMessage是数组，我们需要将editMessage数组拼接为StringBuffer类型。
130. StringBuffer sb1 = new StringBuffer();
131. for (int i = 0; i < editMessage.length; i++) {
132. sb1.append(editMessage[i]);
133. }
134. out = new FileWriter("D:\\count.txt");//覆盖原有文件
135. //sb1是StringBuffer类型，需要使用toString（）方法
136. out.write(sb1.toString());// 写入暂存的内容
137. out.close();
138. }
139. }

在文件7-1中，第十行代码创建了文件内容的全局变量。第14~40行代码是循环主体，用于循环接收用户的指令，当16行代码获取指令后，通过switch语句判断应该执行的操作，第45-52行代码是查看投票功能，第56-90行代码是添加投票功能，使用StringBuffer拼接字符串再存入本地文件。第94-139行代码是投票功能，先将count.txt的内容使用split方法拆分为数组，在循环查询出需要增加投票的人，将票数加一后暂存，再将修改好的字符替换原有的count.txt文件。