【案例6-2】 学生管理系统

### 【案例介绍】

**1.任务描述**

编写一个程序，模拟学生管理系统。可以实现对学生信息的添加、删除、修改和查询功能，要求使用List集合存储自定义的对象集合，并熟练使用List集合的中常用方法实现相关的操作。

**2.运行结果**

学生管理系统首页运行结果如图6-1所示。

手机屏幕截图

描述已自动生成

图6-1 学生管理系统首页运行结果图

学生管理系统查看所有学生运行结果如图6-2所示。

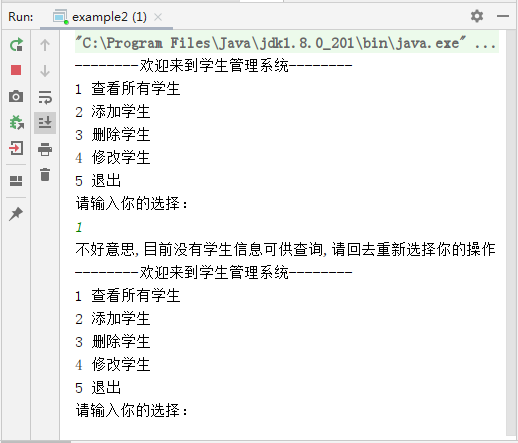


图6-2 查看所有学生信息运行结果图

学生管理系统添加学生信息运行结果如图6-3所示：



图6-3 添加学生信息运行结果图

学生管理系统修改学生信息结果如图6-4所示：

手机屏幕截图

描述已自动生成

图6-4 修改学生信息运行结果图

学生管理系统删除学生结果如图6-5所示。

手机屏幕截图

描述已自动生成

图6-5 删除学生运行结果图

### 【案例任务】

* 学会分析“学生管理系统”任务的实现思路。
* 根据思路独立完成“学生管理系统”任务的源代码编写、编译及运行。
* 掌握List集合常用方法的使用。
* 掌握循环遍历操作集合的使用。

### 【案例思路】

（1）定义学生类，自定义对象属性。

（2）学生管理系统的主界面的代码编写，创建集合对象，用于存储学生数据，打印学生管理系统主界面的相关功能，创建键盘输入功能，用switch语句实现选择的功能。最后为了实现多次操作并且手动退出系统。用while(true)循环实现.

（3） 学生管理系统查询所有学生信息的代码编写，首先判断集合中是否有数据，如果没有数据就给出提示，并让该方法不在继续往下执行。如果有数据。遍历集合输出打印数据。

（4） 学生管理系统添加学生信息的代码编写，首先输入学生学号，判断学号有没有被人占用，如果被占用重新输入学号，没有被占用继续录入学生姓名、年龄。家庭住址等信息。创建学生对象，将录入的数据存入对象。最后将学生对象添加入集合，添加学生信息成功。

（5） 学生管理系统删除学生信息的代码编写，键盘录入一个学号，到集合中去查找，看是否有学生使用该学号，如果有就删除该学生信息。如果没有学生使用该学号，返回主界面。

（6）学生管理系统修改学生信息的代码编写。键盘录入一个学号，到集合中去查找，看是否有学生使用该学号，如果过有就修改学生信息。反之返回主界面。

### 【案例实现】

学生实体类属性如文件6-1所示。

文件6-1 Student.java

1. public class Student {
2. //学号
3. private String id;
4. //姓名
5. private String name;
6. //年龄
7. private String age;
8. //居住地
9. private String address;
10. }

以上代码是我们自定义对象实体类所包括的属性，我们在用时需要重写get（）、set（）以及构造方法。

模拟学生管理系统的代码实现，如文件6-2所示。

文件6-2 example2.java

1. package com.itheima;
2. import java.util.ArrayList;
3. import java.util.Scanner;
4. public class example2 {
5. public static void main(String[] args) {
6. //创建集合对象，用于存储学生数据
7. ArrayList<Student> array = new ArrayList<Student>();
8. //为了让程序能够回到这里来，我们使用循环
9. while(true) {
10. //这是学生管理系统的主界面
11. System.out.println("--------欢迎来到学生管理系统--------");
12. System.out.println("1 查看所有学生");
13. System.out.println("2 添加学生");
14. System.out.println("3 删除学生");
15. System.out.println("4 修改学生");
16. System.out.println("5 退出");
17. System.out.println("请输入你的选择：");
18. //创建键盘录入对象
19. Scanner sc = new Scanner(System.in);
20. String choiceString = sc.nextLine();
21. //用switch语句实现选择
22. switch(choiceString) {
23. case "1":
24. //查看所有学生
25. findAllStudent(array);
26. break;
27. case "2":
28. //添加学生
29. addStudent(array);
30. break;
31. case "3":
32. //删除学生
33. deleteStudent(array);
34. break;
35. case "4":
36. //修改学生
37. updateStudent(array);
38. break;
39. case "5":
40. //退出
41. //System.out.println("谢谢你的使用");
42. //break;
43. default:
44. System.out.println("谢谢你的使用");
45. System.exit(0); //JVM退出
46. break;
47. }
48. }
49. }
50. //查看所有学生
51. public static void findAllStudent(ArrayList<Student> array) {
52. //首先来判断集合中是否有数据，如果没有数据，就给出提示，并让该方法不继续往
53. 下执行
54. if(array.size() == 0) {
55. System.out.println("不好意思,目前没有学生信息可供查询,请回去重新选择你
56. 的操作");
57. return;
58. }
59. //\t 其实就是一个tab键的位置
60. System.out.println("学号\t姓名\t年龄\t居住地");
61. for(int x=0; x<array.size(); x++) {
62. Student s = array.get(x);
63. System.out.println(s.getId()+"\t"+s.getName()+"\t"
64. +s.getAge()+"\t"+s.getAddress());
65. }
66. }
67. }
68. //添加学生
69. public static void addStudent(ArrayList<Student> array) {
70. //创建键盘录入对象
71. Scanner sc = new Scanner(System.in);
72. //为了让id能够被访问到，我们就把id定义在了循环的外面
73. String id;
74. //为了让代码能够回到这里，用循环
75. while(true) {
76. System.out.println("请输入学生学号：");
77. //String id = sc.nextLine();
78. id = sc.nextLine();
79. //判断学号有没有被人占用
80. //定义标记
81. boolean flag = false;
82. //遍历集合，得到每一个学生
83. for(int x=0; x<array.size(); x++) {
84. Student s = array.get(x);
85. //获取该学生的学号，和键盘录入的学号进行比较
86. if(s.getId().equals(id)) {
87. flag = true; //说明学号被占用了
88. break;
89. }
90. }
91. if(flag) {
92. System.out.println("你输入的学号已经被占用,请重新输入");
93. }else {
94. break; //结束循环
95. }
96. }
97. System.out.println("请输入学生姓名：");
98. String name = sc.nextLine();
99. System.out.println("请输入学生年龄：");
100. String age = sc.nextLine();
101. System.out.println("请输入学生居住地：");
102. String address = sc.nextLine();
103. //创建学生对象
104. Student s = new Student();
105. s.setId(id);
106. s.setName(name);
107. s.setAge(age);
108. s.setAddress(address);
109. //把学生对象作为元素添加到集合
110. array.add(s);
111. //给出提示
112. System.out.println("添加学生成功");
113. }
114. //删除学生
115. public static void deleteStudent(ArrayList<Student> array) {
116. //删除学生的思路：键盘录入一个学号，到集合中去查找，看是否有学生使用的是该
117. 学号，如果有就删除该学生
118. //创建键盘录入对象
119. Scanner sc = new Scanner(System.in);
120. System.out.println("请输入你要删除的学生的学号：");
121. String id = sc.nextLine();
122. //我们必须给出学号不存在的时候的提示
123. //定义一个索引
124. int index = -1;
125. //遍历集合
126. for(int x=0; x<array.size(); x++) {
127. //获取到每一个学生对象
128. Student s = array.get(x);
129. //拿这个学生对象的学号和键盘录入的学号进行比较
130. if(s.getId().equals(id)) {
131. index = x;
132. break;
133. }
134. }
135. if(index == -1) {
136. System.out.println("不好意思,你要删除的学号对应的学生信息不存在,请回去
137. 重新你的选择");
138. }else {
139. array.remove(index);
140. System.out.println("删除学生成功");
141. }
142. }
143. //修改学生
144. public static void updateStudent(ArrayList<Student> array) {
145. //修改学生的思路：键盘录入一个学号，到集合中去查找，看是否有学生使用的是该
146. 学号，如果有就修改该学生
147. //创建键盘录入对象
148. Scanner sc = new Scanner(System.in);
149. System.out.println("请输入你要修改的学生的学号：");
150. String id = sc.nextLine();
151. //定义一个索引
152. int index = -1;
153. //遍历集合
154. for(int x=0; x<array.size(); x++) {
155. //获取每一个学生对象
156. Student s = array.get(x);
157. //拿学生对象的学号和键盘录入的学号进行比较
158. if(s.getId().equals(id)) {
159. index = x;
160. break;
161. }
162. }
163. if(index == -1) {
164. System.out.println("不好意思,你要修改的学号对应的学生信息不存在,请回去
165. 重新你的选择");
166. }else {
167. System.out.println("请输入学生新姓名：");
168. String name = sc.nextLine();
169. System.out.println("请输入学生新年龄：");
170. String age = sc.nextLine();
171. System.out.println("请输入学生新居住地：");
172. String address = sc.nextLine();
173. //创建学生对象
174. Student s = new Student();
175. s.setId(id);
176. s.setName(name);
177. s.setAge(age);
178. s.setAddress(address);
179. //修改集合中的学生对象
180. array.set(index, s);
181. //给出提示
182. System.out.println("修改学生成功");
183. }
184. }

在文件6-2中，第8行定义一个集合用于存储学生数据。其中该集合的泛型为Student。第10行代码定义一个while(true)循环使程序能够回到学生管理系统的主界面，第12~18行用于显示主界面的菜单信息。第20~21行代码定义键盘录入对象，用于选择我们要实现的功能。第23~48行代码定义switch语句实现选择。case对应相应的功能。调用相应的程序代码 ，其中System.exit(0)表示JVM退出。第52~68行，定义一个实现学生信息查询方法。其中第55~59行，首先用if判断集合中是否有数据，如果没有数据，就给出提示，并让该方法不继续往下执行。如果有数据，执行第62~66行遍历集合。并输出打印结果。第70~114行，定义一个实现添加学生信息的方法，其中第73行，定义一个id表示输入的学生的学号, 为了让id能够被访问到，我们就把id定义在了循环的外面。第76行，用while循环保证多次输入学号，直到输入的学号不重复。第84~85行代码表示循环遍历集合，获取每一个学生。第87行代码用if判断学号是否被占用。第116~142,行，定义一个删除学生信息的方法。第145~186行，定义一个实现修改学生信息的方法。