【案例3-4】学生投票系统

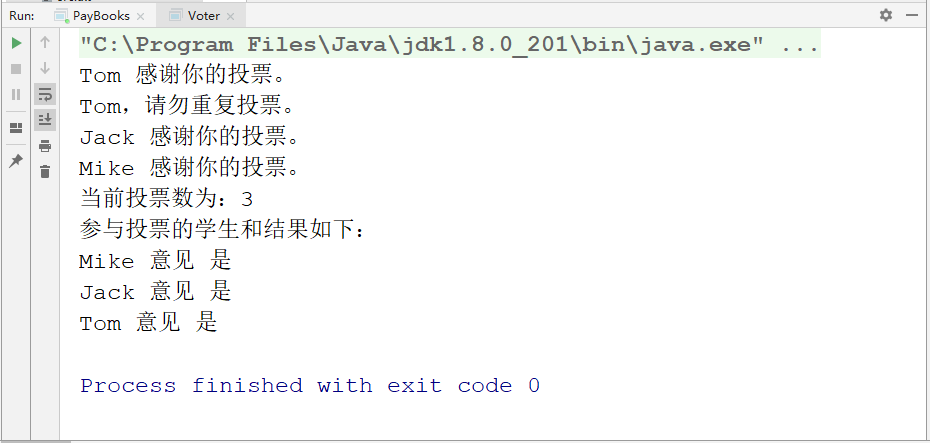
**【案例介绍】**

1. **案例描述**

某班级投票竞选班干部，班级学生人数为100人，每个学生只能投一票。

本任务要求，编程实现一个投票程序，每个学生只能投一次票，投票成功提示“感谢你的投票”，若重复投票，提示“请勿重复投票”。当投票总数达到100时或者主观结束投票时，同时统计投票学生人数和投票结果。

1. **运行结果**



**【任务目标】**

* 学会分析“学生投票系统”程序任务实现的逻辑思路。
* 能够独立完成“学生投票系统”程序的源代码编写、编译、运行。
* 掌握static关键字的使用。
* 区分实例变量和类变量、实例方法和类方法的区别。

**【实现思路】**

（1）通过任务描述可知，需要定义一个参与投票的学生类Voter类，它具有姓名，最大投票数，当前投票总数，和投票意见。

（2）因为所有参与投票的学生都会改变同一个数据，即投票次数，一次把它定义成静态变量：private static int count；

（3）另外，为了防止学生重复投票，必须保存参与投票的学生信息，可采用一个集合来存放已经投票的学生对象。

private static Set<Voter> voters = new HashSet<Voter>();

（4）最后编写测试Voter类的投票和打印投票结果功能。

**【实现代码】**

Voter.java

1. import java.util.HashSet;
2. import java.util.Set;
3. public class Voter {
4. // 属性的定义
5. private static final int MAX\_COUNT = 100; // 最大投票数
6. private static int count; // 投票数
7. // 静态变量，存放已经投票的学生
8. private static Set<Voter> voters = new HashSet<Voter>();
9. private String name;
10. private String answer;
11. // 构造方法
12. public Voter(String name) {
13. this.name = name;
14. }
15. // 投票
16. public void voterFor(String answer) {
17. if (count == MAX\_COUNT){
18. System.out.println("投票结束。");
19. return ;
20. }
21. if (voters.contains(this)){
22. System.out.println(name+"，请勿重复投票。");
23. } else {
24. this.answer = answer;
25. count ++;
26. voters.add(this);
27. System.out.println(name+" 感谢你的投票。");
28. }
29. }
30. // 打印投票结果
31. public static void printVoterResult() {
32. System.out.println("当前投票数为："+count);
33. System.out.println("参与投票的学生和结果如下：");
35. for (Voter voter: voters) {
36. System.out.println(voter.name+" 意见 "+voter.answer);
37. }
38. }
39. public static void main(String[] args) {
40. // 创建参与投票的学生对象
41. Voter tom = new Voter("Tom");
42. Voter jack = new Voter("Jack");
43. Voter mike = new Voter("Mike");
44. // 学生开始投票
45. tom.voterFor("是");
46. tom.voterFor("否");
47. jack.voterFor("是");
48. mike.voterFor("是");
49. // 打印投票结果
50. Voter.printVoterResult();
51. }
52. }