【案例2-7】 抽取幸运观众

### 【案例介绍】

1. **任务描述**

在一些节目活动中，经常会有抽取幸运观众环节。本案例要求编写程序实现幸运观众的抽取，在指定人群中随机抽取一名幸运观众。

案例功能要求如下：

1. 从键盘输入3名观众。
2. 存储观众姓名。
3. 总览观众姓名。
4. 随机选取一名观众，并打印出该观众姓名。
5. **运行结果**

任务运行结果如图2-7所示。

手机屏幕截图

描述已自动生成

图2-7 运行结果

### 【案例目标】

* 学会分析"抽取幸运观众"程序的实现思路。
* 根据思路独立完成” 抽取幸运观众”的源代码编写、编译及运行。
* 能够根据"抽取幸运观众"程序功能的不同，将功能封装到不同的方法中。
* 能够使用数组解决多个观众姓名的存储问题。
* 掌握方法和数组的使用。

### 【案例思路】

1. 在存储观众姓名时，如果对每个观众都定义一个变量进行姓名存储，则会出现过多鼓励的变量，很难一次性将全部数据持有。此时，可以使用数组解决多个数据的存储问题。创建一个可以存储多个观众姓名的数组，这里我们设置数组的长度为3。
2. 键盘输入观众姓名。将输入的姓名依次赋值给数组各元素，此时便存储了所有观众姓名。键盘输入需要使用Scanner类，以下代码能够从键盘输入中读取一个字符串。

Scanner sc=new Scanner(System.in);

String str=sc.next()；

1. 对数组进行遍历，打印出数组每个元素的值，即实现了对每个观众姓名的总览。
2. 根据数组长度，获取随机索引，这里我们的数组长度为3，所以获取的随机索引只能在0-2之间，通过随机索引获取数组中的姓名，该姓名也就是随机的姓名。获取随机索引可以使用Random类中的nextInt(int n）方法。
3. "抽取幸运观众"明确第分为了3个功能，如果将多个独立功能的代码写在一起，代码相对冗长，可以针对不同的功能将其封装到不同的方法中，将完整独立的功能分离出来，然后只需要在程序的main()方法中调用这3个方法即可。

### 【案例实现】

新员工分配部门的实现代码，如文件2-7所示。

文件2-7 audience.java

1. import java.util.Random;
2. import java.util.Scanner;
3. public class audience {
4. /\*
5. \* 存储姓名
6. \* 创建一个存储多个姓名的容器（数组）
7. \* 键盘输入每个同学的姓名，存储到容器中（数组）
8. \*/
9. public static void addAudience(String[] audience) {
10. //键盘输入多个姓名到容器中
11. Scanner sc=new Scanner(System.in);
12. for (int i = 0; i < audience.length; i++) {
13. System.out.println("存储第"+(i+1)+"个姓名：");
14. //接收控制台录入的姓名字符串
15. audience[i]=sc.next();
16. }
17. }
18. /\*
19. \* 总览全部姓名
20. \*/
21. public static void printAudience(String[] audience) {
22. //遍历数组，得到每个观众姓名
23. for (int i = 0; i < audience.length; i++) {
24. String name=audience[i];
25. System.out.println("第"+(i+1)+"个观众姓名:"+name);
26. }
27. }
28. /\*
29. \* 随机点名
30. \*/
31. public static String randomAudience(String[] audience) {
32. //根据数组的长度，获取睡觉索引
33. int index = new Random().nextInt(audience.length);
34. //通过随机索引从数组中获取姓名
35. String name=audience[index];
36. //返回随机到的姓名
37. return name;
38. }
39. public static void main(String[] args) {
40. System.out.println("--------抽取幸运观众---------");
41. //创建一个长度为3的数组存储观众姓名
42. String[] audience=new String[3];
43. //存储观众姓名
44. addAudience(audience);
45. //总览观众姓名
46. printAudience(audience);
47. //随机抽取一名幸运观众
48. String randomName=randomAudience(audience);
49. System.out.println("随机抽取的幸运观众是："+randomName);
50. }
51. }

在文件2-7中，第10~18行代码定义了一个addAudience()方法，用于实现存储观众姓名的功能。其中，通过创建一个String类型的数组audience来存储多个观众姓名，借助Scanner类通过键盘输入的方式输入观众的姓名，并将这些姓名依次存储到数组元素中;第22~28行代码定义一个printAudience ()方法，用于实现总览观众姓名的功能，通过遍历数组audience的方式打印每一位观众的姓名;第32~39行代码定义一个randomAudience ()方法，用于实现随机点名其中一人的功能，通过Random类的nextInt(intn)方法获取随机索引，然后根据随机索引从数组中获取姓名，这个姓名就是幸运观众的姓名。