【案例5-2】 模拟默认密码自动生成

### 【案例介绍】

**1.任务描述**

本例要求编写一个程序，模拟默认密码的自动生成策略，手动输入用户名，根据用户名自动生成默认密码。在生成密码时，将用户名反转即为默认的密码。

**2.运行结果**

运行结果如图5-2所示。

手机屏幕截图

描述已自动生成

图5-2 模拟密码自动生成运行结果图

### 【案例任务】

* 学会分析“模拟默认密码的生成”任务的实现思路。
* 根据思路独立完成“模拟默认密码的生成”任务的源代码编写、编译及运行。
* 掌握String类及Scanner类中常用方法的使用。
* 掌握之前学习的for循环、方法调用等知识。

### 【案例思路】

（1） 分析任务描述可知，做此任务首先需要用Scanner类相关方法实现键盘手动输入一个字符串代表用户名。

（2） 要实现默认密码自动生成，根据任务描述可知默认密码就是手动输入用户名的反转及字符串的反转。因此需要将字符串用循环倒着遍历，用charAt()方法接收遍历的字符并赋值给空串。

（3） 在主函数入口调用刚才的方法。并定义一个字符串变量接收结果

（4） 输出结果，观察控制台的效果。

### 【案例实现】

模拟默认密码生成策略的实现代码，如文件5-2所示。

文件5-2 example2.java

1. package com.itheima;
2. import java.util.Scanner;
3. public class example2 {
4. public static void main(String[] args) {
5. //键盘输入一个字符串，用Scanner实现
6. Scanner sc = new Scanner(System.in);
7. System.out.println("请您输入用户名：");
8. String line = sc.nextLine();
9. //调用方法，用一个变量接收
10. String reverse = reverse(line);
11. //输出结果
12. System.out.println("恭喜"+line+"用户注册成功,您的初始密码
13. 为:"+reverse);
14. }
15. //定义一个方法实现字符串反转
16. public static String reverse(String s) {
17. String ss = "";
18. //在方法中将字符串倒着遍历，然后把每一个得到的字符拼接成一个字符串并且返
19. 回
20. for (int i = s.length()-1; i >=0; i--) {
21. ss+=s.charAt(i);
22. }
23. return ss;
24. }
25. }

在文件5-2中， 第6~8行用Scanner实现了一个从键盘输入字符串，第10行代码调用了reverse()方法，该方法实现了循环遍历字符串中的字符。并且是倒叙遍历的。然后用charAt()方法将单个字符依次返回。最后将倒过来的字符串返回。第12行代码用System类实现将结果输出到打印控制台。