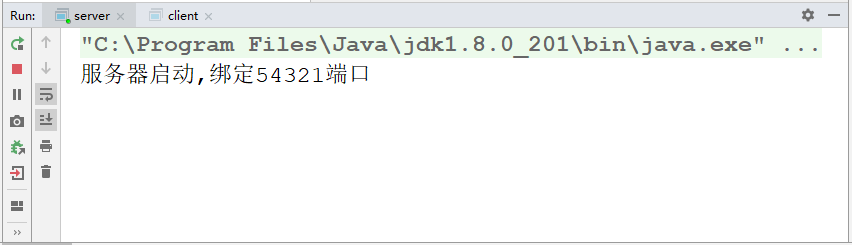
【案例9-3】 反转字符串

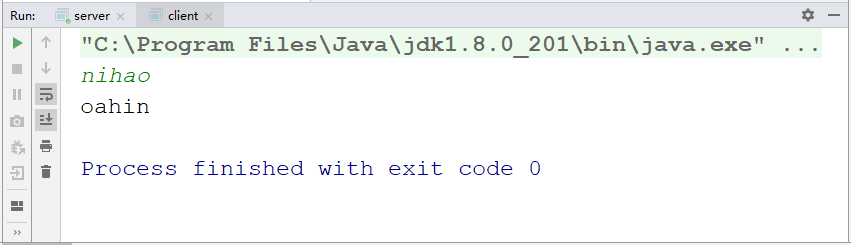
**【案例介绍】**

**1.案例描述**

编写一个小程序。实现客户端向服务器传递一个字符串(键盘录入) ，服务器(多线程)将字符串反转后写回，客户端再次接读取到的是反转后的字符串。要求使用多线程与TCP通信相关知识实现。

**2.运行结果**





**【案例目标】**

* 学会分析“翻跟斗小程序”思路。
* 根据思路独立完成“翻跟斗小程序”任务的源代码编写、编译及运行。
* 掌握网络通信中TCP协议的编程原理。
* 掌握TCP网络通信ServerSocket和Socket的使用。

**【案例分析】**

（1）根据任务描述可以知道该程序用TCP通信技术实现，所以第一条就是定义客户端，键盘录入数据定义Scanner来实现，然后创建客户端指定IP地址和端口号，之后获取输出流，与输入流，最后将字符串写到服务器并将反转后的结果读出来打印在控制台。

（2）实现服务端的代码编写，首先创建服务端绑定客户端的端口号，并用Server的accept()方法接受客户端的请求。

（3）服务端定义run()方法实现之后获取输入输出流，将客户端发送过来的数据读取出来并采用链式编程的思想将字符串反转后返回到客户端。

**【案例实现】**

（1）创建客户端，用于录入输入的数据。其代码具体如下所示。

Client.java

1. import java.io.BufferedReader;
2. import java.io.IOException;
3. import java.io.InputStreamReader;
4. import java.io.PrintStream;
5. import java.net.Socket;
6. import java.net.UnknownHostException;
7. import java.util.Scanner;
8. public class client {
9. public static void main(String[] args) throws UnknownHostException,
10. IOException {
11. //创建键盘录如对象
12. Scanner sc = new Scanner(System.in);
13. //创建客户端,指定ip地址和端口号
14. Socket socket = new Socket("127.0.0.1", 54321);
15. BufferedReader br = new BufferedReader(new
16. InputStreamReader(socket.getInputStream())); //获取输入流
17. //获取输出流
18. PrintStream ps = new PrintStream(socket.getOutputStream());
19. //将字符串写到服务器去
20. ps.println(sc.nextLine());
21. System.out.println(br.readLine()); //将反转后的结果读出来
22. socket.close();
23. }
24. }

上述代码中，第14行代码创建客户端指定IP地址与端口号，第15~18行代码获取输入与输出流，第20行代码用于将字符串写到服务器中去。第21行代码用于将反转后的结果读取出来。

（2）创建服务端实现将客户端数据反转并返回到客户端，其代码如下所示。

Server.java

1. import java.io.BufferedReader;
2. import java.io.IOException;
3. import java.io.InputStreamReader;
4. import java.io.PrintStream;
5. import java.net.ServerSocket;
6. import java.net.Socket;
7. public class server {
8. public static void main(String[] args) throws IOException {
9. ServerSocket server = new ServerSocket(54321);
10. System.out.println("服务器启动,绑定54321端口");
11. while(true) {
12. final Socket socket = server.accept(); //接受客户端的请求
13. new Thread() { //开启一条线程
14. public void run() {
15. try {
16. BufferedReader br = new BufferedReader(new
17. InputStreamReader
18. socket.getInputStream())); //获取输入流
19. PrintStream ps = new PrintStream
20. (socket.getOutputStream());//获取输出流
21. //将客户端写过来的数据读取出来
22. String line = br.readLine()
23. line = new StringBuilder(line).
24. reverse().toString(); //链式编程
25. ps.println(line); //反转后写回去
26. socket.close();
27. } catch (IOException e) {
28. e.printStackTrace();
29. }
30. }
31. }.start();
32. }
33. }
34. }

上述代码中，第9行代码用于创建Server服务器绑定端口号。第12行代码用Server的accept()方法来接收客户端的请求。第22行代码用readLine()方法将客户端写过来的数据读取出来，第23~24行代码用链式编程的方式将字符串反转，第25行代码将反转后的字符串返回给客户端打印。