**服务端与客户端交互对话**

线程版本：要求

1.将字节流改为字符缓存流

2.可以进行重复的说和听

3.一次可以说多句

**读线程类**

示例代码导入：

public class ReaderThread extends Thread {

private BufferedReader br;

public ReaderThread(BufferedReader br) {

this.br = br;

}

@Override

public void run() {

while (true) {

String content;

try {

content = br.readLine();

System.out.println("听到客户说：" + content);

//判断，如果客户睡了88，就表示挂电话，用break从判断语句中出来；

if (content.equals("88")) {

System.out.println("读线程结束！！！");

break;

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

解析：如何得到别的类的对象：先用私有属性的方式把BufferedReader br封装成属性，然后用构造方法的方式传参将BufferedReader的对象br传了过来；

BufferedReader br

public ReaderThread(BufferedReader br) {

this.br = br;

}

**写线程类**

public class WriterThread extends Thread{

private BufferedWriter bw ;

public WriterThread(BufferedWriter bw) {

this.bw = bw;

}同上

@Override

public void run() {

Scanner input = new Scanner(System.in);

while(true) {

try {

System.out.print("客服请输入：");//写活通过让人主动输入的方式确定客服所说的话；

String ask = input.next();

bw.write(ask+"\n");//\n是为了在ask这句话说完以后换行，以此来证明一句话结束了，这个时候才会执行另外的命令；

bw.flush();//刷新缓存才会显示出来

if(ask.equals("88")) {

System.out.println("写线程结束！！");

break;

}

} catch (Exception e) {

e.printStackTrace();

}

}

}

}

解析：如何得到别的类的对象：同上读线程类。

**服务端类**

/\*

\* 服务端：提供服务的端口（移动公司）

\*/

代码导入：

public class Server2 {

public static void main(String[] args) throws IOException {

Scanner input =new Scanner(System.in);//为了写活代码，让客户端和服务端实现对话，就将对话用扫描仪来完成输入；

// 1. 创建一个ServerScoker对象(装了一台热线电话 ：公布号码 10086)

ServerSocket server = new ServerSocket(10086);// 端口号：0~65535端口号的范围；

System.out.println("服务端已经启动........");

// 2. Socket对象，监听ServerSocket对象。（请了很多客户妹妹，监听电话）

Socket socket = server.accept();// 监听服务端口 ，如果有客户端连接那么就会返回一个socket对象

System.out.println("有客户端接入了......");

// 3. io流（交流沟通）

// 3.1说的流------OutputStream

OutputStream os = socket.getOutputStream();// 字节流

【以下的转换流是重点】

// 转换流 OutputStreamWrite

BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(os));// 字节流转字符流 字符流缓存流

// 3.2听的流----InputStream

InputStream in = socket.getInputStream();

// 转换流 InputStreamReader

BufferedReader br = new BufferedReader(new InputStreamReader(in));// 字节流转字符流 字符流转换成换成流

String content = "-1";//先给定content一个默认的值

while (content != null) {

//这里的执行顺序一定要弄清楚，作为客服，肯定是先说后听，下面灰色部分就是具体体现；

// 3.3说话 : "".getBytes(); 将字符串转为字节解析：如何得到别的类的对象：

System.out.print("客服请输入：");

String ask = input.next();

bw.write(ask+"\n");

bw.flush();//刷新缓存

// 3.4 听

content = br.readLine();

System.out.println("听到客户说：" + content);

//挂电话

if(ask.equals("88")&&content.equals("88")) {

System.out.println("挂电话了");

break;

}

}

// 4.关流（挂电话）

br.close();

in.close();

bw.close();

os.close();//关流要按照顺序，先开后关；

}

}

**客户端类**

示例代码导入：

public class Client2 {

public static void main(String[] args) throws UnknownHostException, IOException {

Scanner input = new Scanner(System.in);//为了写活代码，让客户端和服务端实现对话，就将对话用扫描仪来完成输入；

// 1.得到电话和电话的人

Socket socket = new Socket("127.0.0.1", 10086);

// 2.交流

// 2.1说的流------OutputStream

OutputStream os = socket.getOutputStream();// 字节流

// 转换流 OutputStreamWrite

BufferedWriter bw = new BufferedWriter(new OutputStreamWriter(os));// 字节流转字符流 字符流缓存流

// 2.2听的流----InputStream

InputStream in = socket.getInputStream();

// 转换流 InputStreamReader

BufferedReader br = new BufferedReader(new InputStreamReader(in));// 字节流转字符流 字符流转换成换成流

//身为客户端的客户，在接通电话以后肯定是先接听到服务端的话然后才开始自己说的；

// 2.3.先听

String content = br.readLine();

while(content != null) {

System.out.println("听到客服说：" + content);

// 2.4后说

System.out.print("客户请输入：");

String answer = input.next();

bw.write(answer+"\n");

bw.flush();

//判断是否要挂电话

if(answer.equals("88")&&content.equals("88")) {

System.out.println("挂电话了");

break;

}

//继续读

content = br.readLine();

}

// 3.关流

br.close();

in.close();

bw.close();

os.close();

}

}