代碼示例1：客戶端发送信息給服務端，服務端將數據顯示在控制台上

//客戶端  
 @Test  
 public void client() {  
 Socket socket = null;  
 OutputStream os = null;  
 try {  
 //1.創建Socket對象，指明服務器端的ip和端口號  
 InetAddress inet = InetAddress.getByName("192.168.14.100");  
 socket = new Socket(inet,8899);  
 //2.獲取一個輸出流，用於輸出數據  
 os = socket.getOutputStream();  
 //3.寫出數據的操作  
 os.write("你好，我是客戶端mm".getBytes());  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 //4.資源的關閉  
 if(os != null){  
 try {  
 os.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 if(socket != null){  
 try {  
 socket.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }  
 //服務端  
 @Test  
 public void server() {  
 ServerSocket ss = null;  
 Socket socket = null;  
 InputStream is = null;  
 ByteArrayOutputStream baos = null;  
 try {  
 //1.創建服務器端的ServerSocket，指明自己的端口號  
 ss = new ServerSocket(8899);  
 //2.調用accept()表示接收來自於客戶端的socket  
 socket = ss.accept();  
 //3.獲取輸入流  
 is = socket.getInputStream();  
  
 //不建議這樣寫，可能會有亂碼  
// byte[] buffer = new byte[1024];  
// int len;  
// while((len = is.read(buffer)) != -1){  
// String str = new String(buffer,0,len);  
// System.out.print(str);  
// }  
 //4.讀取輸入流中的數據  
 baos = new ByteArrayOutputStream();  
 byte[] buffer = new byte[5];  
 int len;  
 while((len = is.read(buffer)) != -1){  
 baos.write(buffer,0,len);  
 }  
 System.out.println(baos.toString());  
 System.out.println("收到了來自於：" + socket.getInetAddress().getHostAddress() + "的數據");  
 } catch (IOException e) {  
 e.printStackTrace();  
 } finally {  
 if(baos != null){  
 //5.關閉資源  
 try {  
 baos.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 if(is != null){  
 try {  
 is.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 if(socket != null){  
 try {  
 socket.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 if(ss != null){  
 try {  
 ss.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
 }

代碼示例2：客戶端发送文件給服務端，服務端將文件保存在本地。

/\*  
這里涉及到的異常，應該使用try-catch-finally處理  
 \*/  
@Test  
public void client() throws IOException {  
 //1.  
 Socket socket = new Socket(InetAddress.getByName("127.0.0.1"),9090);  
 //2.  
 OutputStream os = socket.getOutputStream();  
 //3.  
 FileInputStream fis = new FileInputStream(new File("beauty.jpg"));  
 //4.  
 byte[] buffer = new byte[1024];  
 int len;  
 while((len = fis.read(buffer)) != -1){  
 os.write(buffer,0,len);  
 }  
 //5.  
 fis.close();  
 os.close();  
 socket.close();  
}  
/\*  
這里涉及到的異常，應該使用try-catch-finally處理  
 \*/  
@Test  
public void server() throws IOException {  
 //1.  
 ServerSocket ss = new ServerSocket(9090);  
 //2.  
 Socket socket = ss.accept();  
 //3.  
 InputStream is = socket.getInputStream();  
 //4.  
 FileOutputStream fos = new FileOutputStream(new File("beauty1.jpg"));  
 //5.  
 byte[] buffer = new byte[1024];  
 int len;  
 while((len = is.read(buffer)) != -1){  
 fos.write(buffer,0,len);  
 }  
 //6.  
 fos.close();  
 is.close();  
 socket.close();  
 ss.close();  
}

代碼示例3：從客戶端發送文件給服務端，服務端保存到本地。並返回“發送成功”給客戶端。並關閉相應的連接。

/\*  
 這里涉及到的異常，應該使用try-catch-finally處理  
 \*/  
@Test  
public void client() throws IOException {  
 //1.  
 Socket socket = new Socket(InetAddress.getByName("127.0.0.1"),9090);  
 //2.  
 OutputStream os = socket.getOutputStream();  
 //3.  
 FileInputStream fis = new FileInputStream(new File("beauty.jpg"));  
 //4.  
 byte[] buffer = new byte[1024];  
 int len;  
 while((len = fis.read(buffer)) != -1){  
 os.write(buffer,0,len);  
 }  
 //關閉數據的輸出(這樣主機端才知道檔案傳輸結束才能夠接收訊息)  
 socket.shutdownOutput();  
  
 //5.接收來自於服務器端的數據，並顯示到控制台上  
 InputStream is = socket.getInputStream();  
 ByteArrayOutputStream baos = new ByteArrayOutputStream();  
 byte[] bufferr = new byte[20];  
 int len1;  
 while((len1 = is.read(buffer)) != -1){  
 baos.write(buffer,0,len1);  
 }  
 System.out.println(baos.toString());  
 //6.  
 fis.close();  
 os.close();  
 socket.close();  
 baos.close();  
}  
/\*  
這里涉及到的異常，應該使用try-catch-finally處理  
 \*/  
@Test  
public void server() throws IOException {  
 //1.  
 ServerSocket ss = new ServerSocket(9090);  
 //2.  
 Socket socket = ss.accept();  
 //3.  
 InputStream is = socket.getInputStream();  
 //4.  
 FileOutputStream fos = new FileOutputStream(new File("beauty2.jpg"));  
 //5.  
 byte[] buffer = new byte[1024];  
 int len;  
 while((len = is.read(buffer)) != -1){  
 fos.write(buffer,0,len);  
 }  
  
 System.out.println("圖片傳輸完成");  
  
 //6.服務器端給予客戶端反饋  
 OutputStream os = socket.getOutputStream();  
 os.write("你好，美女，照片我已收到，非常漂亮！".getBytes());  
  
 //7.  
 fos.close();  
 is.close();  
 socket.close();  
 ss.close();  
 os.close();  
}