

- 服务端
  - 创建一个套接字地址：实例化ServiceSocket对象，构造参数传入int类型端口号
- 客户端
  - 创建一个套接字：实例化Socket，构造方法传入InetAddress类型IP地址和int类型端口号
  - （下面的直到发送消息是服务端的笔记，写错地方了）
  - 等待客户端连接：Socket serviceSocket.accept() 这个返回的Socket其实就是客户端连接过来的Socket
  - 读取客户端发过来的信息：
    - 获取输入流：InputStream  
socket.getInputStream(), 这个方法需要对方调用socket.shutdownOutput()或者关闭了outputStream才会执行，否则会阻塞（需要验证一下这个结论）
    - 管道流读取：
      - 因为如果中文占两个字节，原本while循环体是用一个StringBuffer读取，数据过长防止缓冲区不够大导致可能最后一个汉字只截到了一个字节导致乱码，接收数据尽量使用流，接收的不是文件只是普通字符串，使用管道流ByteArrayOutputStream
      - 实例化ByteArrayOutputStream对象baos

- 创建一个byte[]缓冲区buffer大小1024，创建int数据长度length
  - while循环，循环条件为  
(length=inputStream.read(buffer))!= -1，循环体为  
baos.write(buffer,0,length)
  - 拿到数据字符串：String  
baos.toString()
- 关闭资源（先开后关，要先判断资源是否为null，放在finally代码块里）
  - baos.close()
  - input.close()
  - socket.close()
  - serviceSocket.close()
- 发送消息
  - 获取输出流：OutputStream  
socket.getOutputStream()
  - 输出：outputStream.write(byte[] bs)
  - 只发一次就不用outputStream.flush()，因为关闭流会自动刷新
  - socket.shutdownOutput() 通知服务器自己已经发送完毕了，建议不要忘了
- 关闭资源
  - outputStream.close()
  - socket.close()

- 开启聊天程序，要先开启服务端，在开启客户端，不然会报错