**实 验 三 Socket编程**

（四学时）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **课 程** | 计算机网络基础 | **实验项目** | Socket程序设计 | **成 绩** |  |
| **专业班级** | 软件工程2022级 | **学 号** | 202231060920 | **批阅日期** |  |
| **姓 名** | 李浩楠 | **实验日期** | 2024/1/3 | **指导教师** | 何中越 |

**一【实验目的】**

1. 掌握基于多线程的复杂网络程序的开发方法和开发流程；
2. 按照要求设计实现软件，以培养应用复杂计算系统开发的能力，并在设计和开发过程中体现出创新意识。

**二【实验内容】**

1-1 假设Tom和Jerry利用Java UDP进行聊天，请为他们编写程序。具体如下：

（1）、Tom和Jerry聊天的双方都应该具有发送端和接收端；

（2）、利用DatagramSocket与DatagramPacket；

（3）、实现 java.lang.Runnable类，重写 run()方法。

1-2 利用Java Socket编写一个简单的Web服务器，具体如下：

（1）、使用 ServerSocket 监听某一端口，然后等待连接获取 Socket对象；

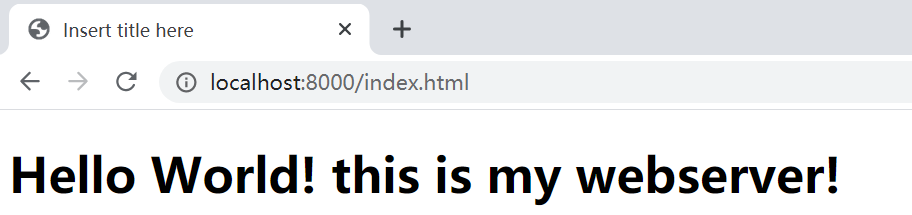
（2）、创建一个类 HttpServer 继承 java.lang.Thread 类，重写 run()方法，执行浏览器请求；

（3）、获得浏览器请求，解析资源文件路径；

（4）、读取资源文件，响应给浏览器；

（5）、浏览器地址栏输入： <http://localhost:8000/index.html>；

（6）、效果如下：



**三【实验步骤】**

1. 设计思路

通过流程图或者伪代码描述程序设计思路

实验1-1

创建Tom和Jerry的发送端和接收端。

创建DatagramSocket对象，用于发送和接收数据。

创建DatagramPacket对象，用于存储要发送的数据。

继承Thread类，重写run()方法。

在run()方法中，使用DatagramSocket对象发送数据。

使用DatagramSocket对象接收数据。

打印接收到的数据。

实验1-2

创建一个ServerSocket对象，监听8000端口。

当有客户端连接时，获取Socket对象。

创建一个HttpServer线程类，继承Thread类，重写run()方法。

在run()方法中，读取客户端请求。

解析请求中的资源文件路径。

根据资源文件路径读取文件内容。

将文件内容作为响应发送给客户端。

关闭Socket连接。

1. 实现过程

给出主要步骤

实验1-1

创建一个名为TomReceive、JerryRecevie、TomSend、JerrySend的类，继承Thread类。

重写类的run()方法。在run()方法中，使用TomSend、JerrySend对象发送数据。使用TomReceive、JerryRecevie对象接收数据。

在主函数中，创建UserTom、UserJerry对象，并启动线程。

实验1-2

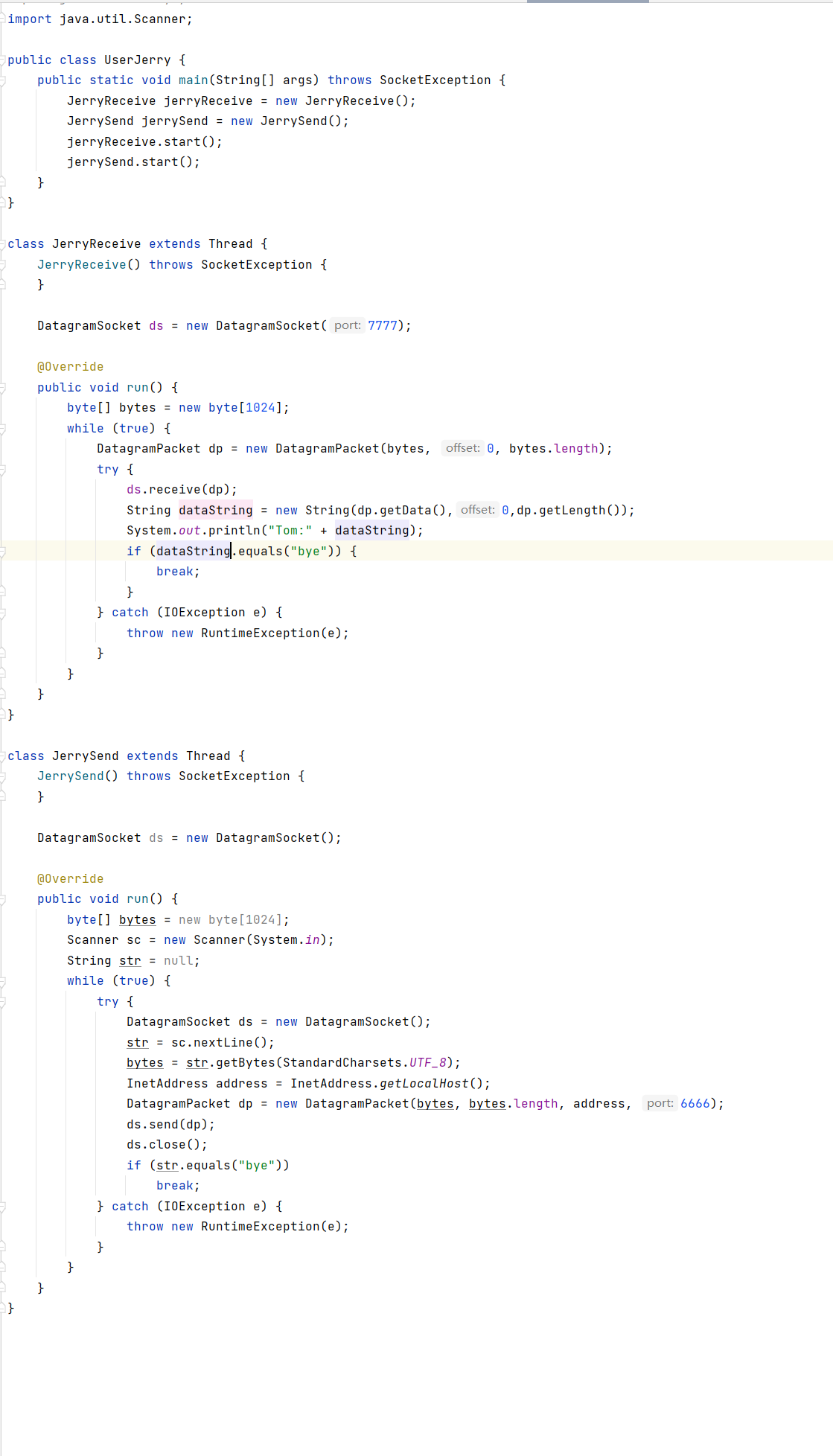
1. 导入所需的库。
2. 创建一个ServerSocket对象，监听8000端口。
3. 使用while循环等待客户端连接。
4. 当有客户端连接时，获取Socket对象。
5. 创建一个HttpServer线程类，继承Thread类，重写run()方法。
6. 在run()方法中，读取客户端请求。
7. 解析请求中的资源文件路径。
8. 根据资源文件路径读取文件内容。
9. 将文件内容作为响应发送给客户端。
10. 关闭Socket连接。
11. 重复步骤3-10，直到服务器关闭。

**四【实验结果】**

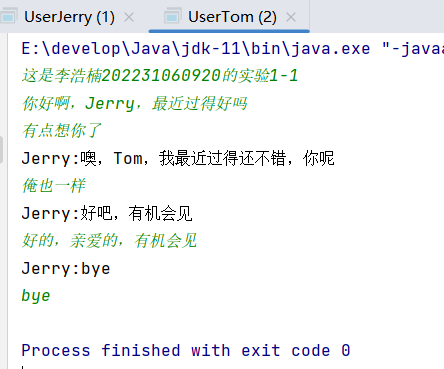
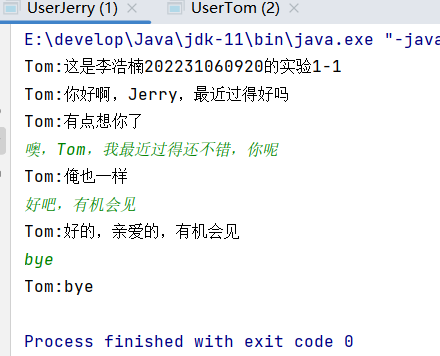
主要运行结果及说明

1-1：

实现UDP通信的代码：Tom和Jerry通过UDP实现多线程聊天，当任意一方发送的信息是bye时结束聊天



实验结果：双发可以正常聊天



1-2

实验代码：





运行结果：通过键入<http://localhost:8000/index.html>获得此页面



**五【总结】**

可以包括：

1. 问题及解决方法

实验1-1

双方都是接收端和发送端，因此在发送与接收数据后需要区别端口，如果不加以区分会造成接收方接收了数据，发送方也会接收到自己的数据，解决方法就是使用两个不同的端口。

实验1-2

若出现请求成功，响应页面为空白，可以去掉 index.html 页面的 <html>标签。

浏览器可能无法解析。原因是："HTTP /1.1 200 ok" 中，"HTTP"和"/1.1" 不能有空格，否则有些浏览器无法解析 。

关于换行符问题补充：

a. windows 换行符： "\r\n"

b. linux 换行符： "\n"

c. mac 换行符："\r"

1. UDP、TCP协议对比

TCP和UDP都是传输层的重要协议，属于TCP/IP协议簇。它们在数据传输方式和特性上存在显著的差异。TCP是一种面向连接的协议，这确保了数据的可靠传输，适用于要求数据准确无误的场景。UDP是无连接的，即在发送数据之前无需建立连接，UDP传输速度更快，但不保证数据的完整性。TCP是面向字节流的，UDP是面向数据报文的。TCP支持一对一、一对多、多对多的通信模式，有拥塞控制机制来避免网络拥堵。UDP只支持一对一和一对多的通信模式。

TCP和UDP在首部长度上也有所不同，TCP报文首部为20个字节，而UDP首部只有8个字节。

TCP和UDP各有其优势和适用场景。TCP适合需要可靠、有序和高吞吐量的应用场景，如文件传输、电子邮件等；而UDP则适用于实时应用，如视频聊天、在线游戏等，这些场景更注重速度和实时性，可以容忍一定的数据丢失或乱序。

1. 服务器编程要点

Web前端基础：包括HTML/CSS/Javascript，这是你在使用Java服务器后台生成内容时需要掌握的基础知识。网络编程基础：包括网络入门、协议、网络通信协议等网络编程三要素，如IP地址、端口号等。

协议：了解HTTP协议和TCP/IP协议等，这是进行网络编程的基础。

多线程编程：在Java服务器编程中，多线程编程是非常重要的一部分，可以大大提高程序的性能。