南昌大学实验报告

姓名：**陈华豪**

学号：**6130116238**

邮箱地址：**6130116238@email.ncu.edu.cn**

专业班级：**网络工程161班**

实验日期：**2018.10.28**

课程名称：**网络协议分析与实现**

实验项目名称

家庭作业一

实验目的

The purpose of this assignment is to give you experience writing programs that implement sockets. Sockets are the key interface between
applications and the Internet. In this assignment you must program both TCP and UDP sockets

实验数据或结果

code：

- tcpjava:

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.PrintStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.Scanner;

//默认服务器ip为本机
```

```
public class serverjavatcp {
    public static void main(String[] args) throws Exception{

        Socket client = null;

        System.out.print("good> sever ");
        Scanner sc = new Scanner(System.in);            //获取服务端输入
        int port = sc.nextInt();                           //存储要监听的端
口号
        System.out.println("[Eventually, after the server responds, it sh
ould just exit]");

        //监听输入的端口请求的TCP连接
        ServerSocket server = new ServerSocket(port);
        System.out.print("good> ");
        client = server.accept();

        //为每个客户端连接开启一个线程
        new Thread(new ServerThread(client)).start();
        server.close();
    }
}
class ServerThread implements Runnable {

    private Socket client = null;
    public ServerThread(Socket client){
        this.client = client;
    }

    @Override
    public void run() {
        try{
            PrintStream out = new PrintStream(client.getOutputStream());
            BufferedReader buf = new BufferedReader(new InputStreamReader
(client.getInputStream()));
            boolean flag =true;
            while(flag){
                String str =  buf.readLine();
                if(str == null || "".equals(str)){
                    flag = false;
                }else{
                    str=str.toLowerCase();
                    str=str.replaceAll(" +",".");
                    out.println(str);
                }
            }
            out.close();
            client.close();
        }catch(Exception e){
            e.printStackTrace();
        }
    }
```

```
}


import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.PrintStream;
import java.net.Socket;
import java.net.SocketTimeoutException;
import java.util.Scanner;

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

        System.out.print("cutepie> client ");

        Scanner qq = new Scanner(System.in);        //读取服务器输入
        String ip = qq.next();                      //存储域名或ip地址
        int port = qq.nextInt();                    //存储端口号

        //客户端请求与用户设定的地址和端口建立TCP链接
        Socket client = new Socket(ip, port);

        client.setSoTimeout(10000);                 //设置超时时间

        //获取键盘输入
        BufferedReader input = new BufferedReader(new InputStreamReader(S
ystem.in));

        //获取Socket的输出流，用来发送数据到服务端
        PrintStream out = new PrintStream(client.getOutputStream());

        //获取Socket的输入流，用来接收从服务端发送过来的数据
        BufferedReader buf =  new BufferedReader(new InputStreamReader(cl
ient.getInputStream()));
        boolean flag = true;

            System.out.print("Enter text: ");
            String str = input.readLine();

            //发送数据到服务端
            out.println(str);
        String qqq = buf.readLine();
        System.out.println(qqq);
        System.out.print("cutepie> ");

        input.close();
        if(client != null){
            client.close();
        }
    }
}
```

- udpjava

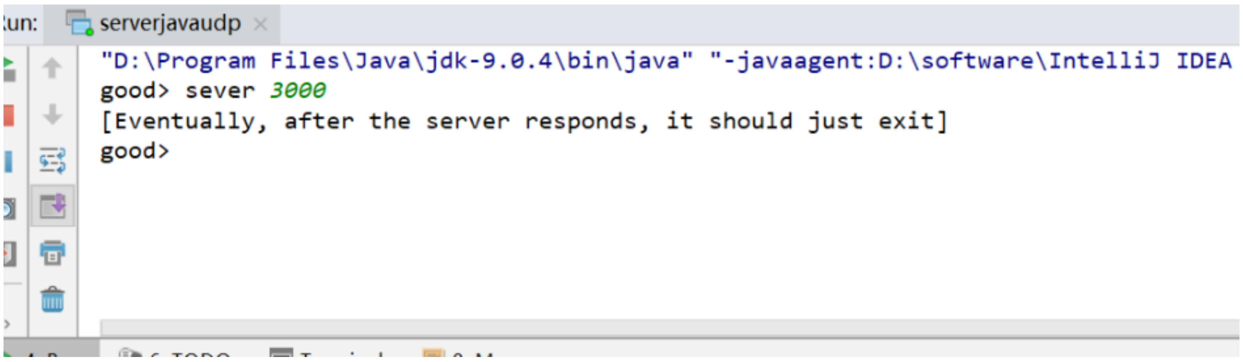
```
import java.io.IOException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.util.Scanner;

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

        String send = new String();
        byte[] buf = new byte[1024];
        System.out.print("good> sever ");
        Scanner sc = new Scanner(System.in);            //获取服务端输入
        int port = sc.nextInt();                          //存储要监听的端
口号
        System.out.println("[Eventually, after the server responds, it sh
ould just exit]");
        DatagramSocket ds = new DatagramSocket(port);
        System.out.print("good> ");
        //接收从客户端发送过来的数据
        DatagramPacket dp_receive = new DatagramPacket(buf, 1024);
        boolean f = true;
        while(f){
            //服务器端接收来自客户端的数据
            ds.receive(dp_receive);
            String str_receive = new String(dp_receive.getData(),0,dp_rec
eive.getLength());

            send= str_receive;
            send= send.toLowerCase();
            send=send.replaceAll(" +",".");
            //数据发动到客户端的port端口
            DatagramPacket dp_send= new DatagramPacket(send.getBytes(),se
nd.length(),dp_receive.getAddress(),2121);
            ds.send(dp_send);

            dp_receive.setLength(1024);
        }
        ds.close();
    }
}
```

```
import java.io.IOException;
import java.io.InterruptedIOException;
import java.net.DatagramPacket;
```

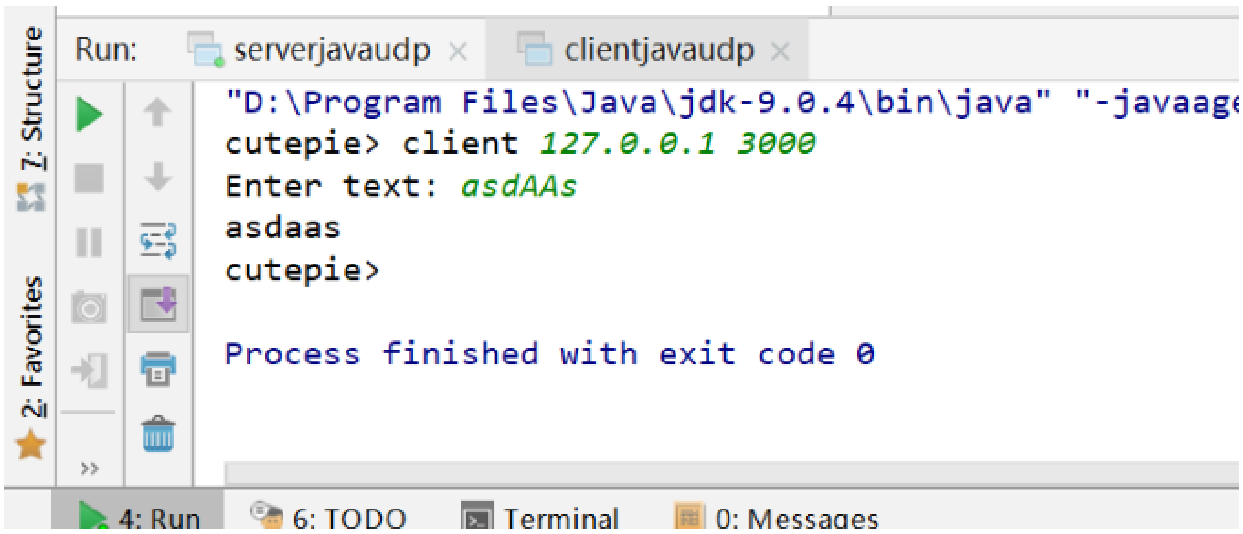
```java
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class clientjavaudp {
    private static final int TIMEOUT = 5000;
    private static final int MAXNUM = 5;
    public static void main(String args[])throws IOException{
        System.out.print("cutepie> client ");
        Scanner in =new Scanner(System.in);               //存储输入的字符
        String adress = in.next();                        //存储ip地址
        int port = in.nextInt();                          //存储端口号
        System.out.print("Enter text: ");
        String send=in.next();                            //存储发送的信息
        byte[] buf = new byte[1024];


        DatagramSocket ds = new DatagramSocket(2121);
        InetAddress loc = InetAddress.getLocalHost();
        //定义用来发送数据的DatagramPacket实例
        DatagramPacket dp_send= new DatagramPacket(send.getBytes(),send.l
ength(),loc,port);
        //定义用来接收数据的DatagramPacket实例
        DatagramPacket dp_receive = new DatagramPacket(buf, 1024);
        //数据发向本地3000端口
        ds.setSoTimeout(TIMEOUT);                   //设置接收数据时阻塞的最长时间

        int tries = 0;                              //重发数据的次数
        boolean receivedResponse = false;          //是否接收到数据的标志位
        //直到接收到数据，或者重发次数达到预定值，则退出循环
        while(!receivedResponse && tries<MAXNUM){
            ds.send(dp_send);
            try{
                //接收从服务端发送回来的数据
                ds.receive(dp_receive);
                if(!dp_receive.getAddress().equals(loc)){
                    throw new IOException("Received packet from an umknow
n source");
                }
                receivedResponse = true;
            }catch(InterruptedIOException e){
                tries += 1;
                System.out.println("Time out," + (MAXNUM - tries) + " mor
e tries..." );
            }
        }
        if(receivedResponse){
            String str_receive = new String(dp_receive.getData(),0,dp_rec
eive.getLength());
            System.out.println(str_receive);
            System.out.println("cutepie> ");
            dp_receive.setLength(1024);
            }else{
                System.out.println("No response -- give up.");
            }
            ds.close();
        }
    }
```

截图:

UDP java:

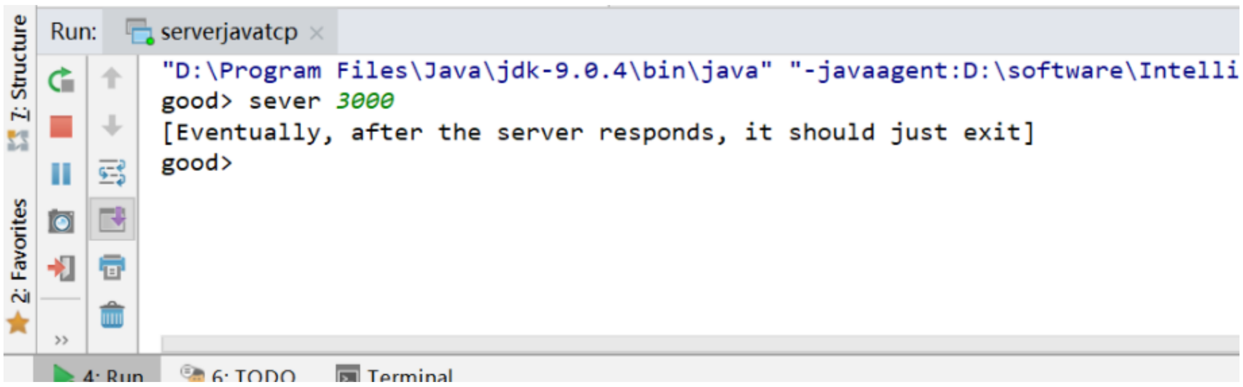


```
"D:\Program Files\Java\jdk-9.0.4\bin\java" "-javaagent:D:\software\IntelliJ IDEA
good> sever 3000
[Eventually, after the server responds, it should just exit]
good>
```

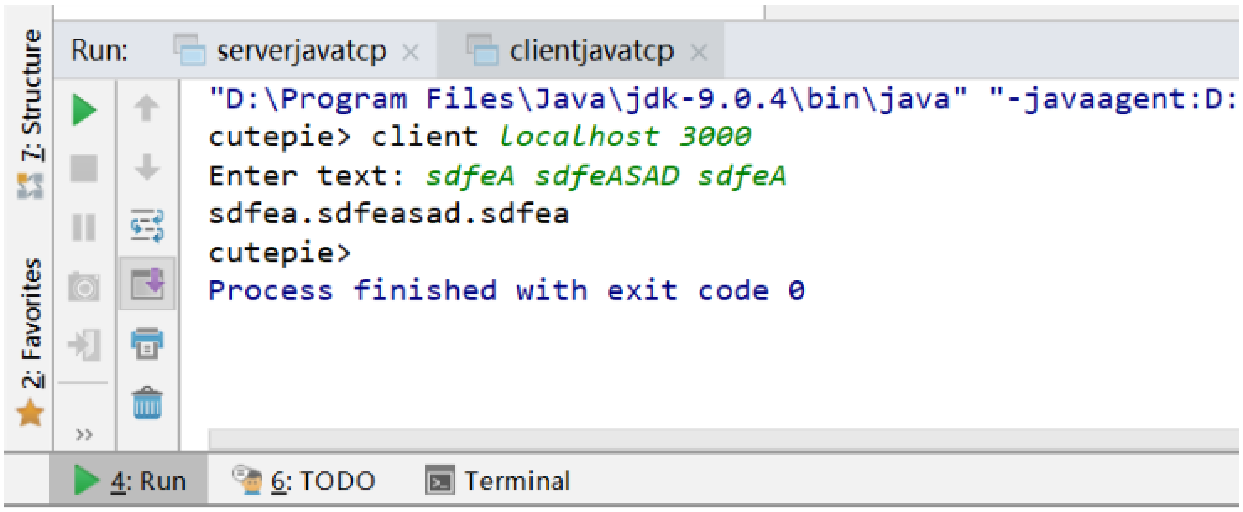


```
"D:\Program Files\Java\jdk-9.0.4\bin\java" "-javaage
cutepie> client 127.0.0.1 3000
Enter text: asdAAs
asdaas
cutepie>

Process finished with exit code 0
```

TCP java:



```
"D:\Program Files\Java\jdk-9.0.4\bin\java" "-javaagent:D:\software\Intelli
good> sever 3000
[Eventually, after the server responds, it should just exit]
good>
```



```
"D:\Program Files\Java\jdk-9.0.4\bin\java" "-javaagent:D:
cutepie> client localhost 3000
Enter text: sdfeA sdfeASAD sdfeA
sdfea.sdfeasad.sdfea
cutepie>
Process finished with exit code 0
```