tcp/ip协议
应用层，传输层，网络层，数据+链路层

一:(0-255)
通过IP地址，可以唯一的定位一台主机
inetaddress:IP地址在这个类中，代表一个IP地址，一个inetaddress代表一个IP地址，在Java.net类下:
创建对象:InetAddress inet=InetAddress.get ByName(需要查看IP的域名string host);
inet.getHostName();获取IP地址所在的域名
inet.getHostAddress();获取IP地址(一般访问网站都是要用域名，域名比较形象)dns域名解析服务器

访问本机:InetAddress inet=InetAddress.get LocalHost();

二:端口号(0-65535)端口号标识正在计算机上运行的进程
1，不同的进程拥有不同的端口号
2，端口号在(0-65535)之间，(0-1023)被预先定义的服务通信占用，除非我们要访问这些特定的服务，否则，我们应该使用(1024-65535)这些端口号中的某一个通信，以免端口发生冲突

”端口号和IP地址的组合得出一个网络套接字

三，网络通信协议
tcp:传输控制协议 udp用户数据报协议
tcp使用前，需要先建立tcp连接，形成可靠的数据传输通道，三次握手，保证链接是可靠的，可建立进行大量的数据传输，传输完毕，需释放已经建立的链接，效率低，
udp将数据，源，目的封装成为数据包，不需要建立链接，每个数据报的大小控制在64k以内，是不可靠的，发送完数据后不需要释放资源，速度快。

socket(套接字)通信的俩端都需要有socket，是俩台计算机之间通信的端点，网络通信其实就是socket通信，socket允许程序把网络连接当成一个流，数据在俩个socket之间通过io传输，一般主动发起通信的应用程序属于客户端，等待请求通信的一端称为服务端

客户端:
1，创建一个socket对象，通过构造器指明服务端的IP地址以及接收程序的端口号

2，getoutputstream()，发送数据，方法返回outputstream的对象

3，具体的输出过程

4，关闭相应的流和socket对象

```
Socket socket=new Socket(InetAddress.getByName(‘域名’)，端口号);
OutputStream os=socket.getOutputStream();
os.write("输出的内容");
os.close();
socket.close();
```

服务端:

1，创建起一个serversocket的对象，通过构造器指明自身的端口号

2，调用accept方法，返回一个socket对象

3，调用socket对象，的getinputstream方法获取一个客户端发送出来的输入流

4，对获取的流进行操作

5，关闭相应的流以及socket和serversocket

```
SeverSocket ss=new SeverSocket();
Socket s=ss.accept();
InputStream is=s.getInputStream();
byte[]b=new byte[];
int len;
while((len=is.read(b))!=-1){
string str=new String(b,0,len);
system.out.print(str);
}
is.close();
s.close();
ss.close();
```