**网络编程(Socket编程)**

1. **网络编程：用java实现计算机间数据的信息传递和资源共享**
2. **网络编程模型**

**A:TCP/IP参考模型**

**B:OSI参考模型（七层模型）**

**应用层**

**表示层**

**会话层**

**传输层**

**网络层--数据包**

**数据链路层--帧**

**物理层--比特**

1. **网络编程三要素：**

**A:IP地址**（网络号、主机号）、A类、B类、C类

InetAddress类

public **static** InetAddress getByName(String host) 根据主机名或IP地址的字符串表示得到**IP地址对象**

public String getHostName() 获取主机名

public String getHostAddress() 获取IP地址

**B:端口**（16位）

**C:协议**

UDP协议：数据打包、数据有限制、不建立连接、只管发、**速度快**、不可靠

TCP协议：建立数据通道、、数据无限制、速度慢、**可靠**

1. **Socket**（套接字）

Socket=IP地址+端口号

**Socket机制**

A:通信两端都要有Socket对象

B:所有的通信都是通过Socket间的IO进行操作的

1. **UDP协议发送和接收数据**

**发送数据**

A:创建发送端DatagramSocket对象d

B:创建数据和数据包对象dt，并把数据打包

数据包类DatagramPacket

DatagramPacket(byte[] bys,int length,InetAddress address,int port)

C:调用d. send(dt)方法发送数据包

D:释放资源d.close()

**接收数据**

A:创建接收端DatagramSocket(int port)对象d

B:创建一个数据包对象dt（用于接收）

DatagramPacket(byte[] bys,int length)

C:调用d.receive(dt)方法接收数据包

D:解析数据包，并显示在控制台

public byte[] getData() 获取数据缓冲区

public int getLength() 获取数据的实际长度

String s=new String(dt.getData(),0, dt.getLength());

E:释放资源d.close()

1. **TCP协议发送和接收数据**

**客户端：发送数据**

A:创建发送端的Socket(“192.168.0.1”,int port)对象s

创建成功就说明连接已经建立成功了

B:获取输出流，写入数据,并终止写入

public OutputStream getOutputStream() 获取输出流

OutputStream os=s. getOutputStream()

os.write(byte[] bys)

s.dispose()//写完了

C:释放资源s.close()

**服务器端：接收数据**

A:创建接收端的ServerSocket(int port)对象st

B:监听客户端连接，返回一个对应的发送端Socket对象s

Socket s=st.accept();

C:获取输入流，读取数据，并显示在控制台

InputStream is=s.getInputStream();

D:释放资源s.close()