

网络编程

网络常识

1. 什么是计算机网络

分布在不同地域的计算机，通过硬件等网络设备使用通信线路互相连接形成的一个网络系统。
计算机网络，可以很方便的进行 信息的传递，资源的共享 ！

2. 什么是计算机的IP地址

IP地址 是计算机在互联网中的唯一标识 。 就像人在社会中的身份证号码。
本机IP：
127.0.0.1
localhost

3. 什么是 网络中 网站的域名

域名可以简单的理解为，IP地址的别名。 更方便记忆， 当输入域名后(例如www.baidu.com) ，计算机访问域名解析商 ，然后得到ip地址，再进行访问。

4. 什么是计算机的端口号

端口号的范围 0-65535 之间 。 *****

与ip地址很相似，IP地址是计算机在网络中的唯一标识 。
端口号是计算机中 程序的标识 。 用于在一台计算机中区分不同的应用程序

端口号在使用时 ，应尽量避免0-1024之间的端口号，因为已经被一些知名的软件 和 windows操作系统所占用了。

5. 什么是计算机之间的通信协议

是计算机与计算机之间交流的标准 。
是对数据的 传输速率，传入接口，步骤控制 出错控制 等等 制定的一套标准 ！

常用的通信协议：

1. http协议 ： 超文本传输协议 。 80端口号
2. https协议： 安全的超文本传输协议 443端口号
3. ftp协议： 文件传输协议 21端口号
4. TCP协议： 传输控制协议
5. UDP协议： 数据报协议

网络 编程程序的分类：

1.B/S 程序 ： 浏览器与服务器程序

2.C/S 程序 ： 客户端与服务器程序

TCP协议 - OSI网络模型

指的是 从一台计算机的软件中，将数据发送到另一台计算机的软件中的过程。

七层网络模型： 应用层 / 表现层 / 会话层 / 传输层 / 网络层 / 数据链路层 / 物理层

三次握手 和 四次挥手 熟悉

tcp协议客户端与服务器连接时， 存在三次握手操作，确保消息能准确无误的发送。
断开连接是时 ，存在四次挥手操作

TCP 协议的 C/S程序 ***

需要使用到两个类，来编写TCP协议的 CS程序 。

1.ServerSocket 搭建服务器

2.Socket 搭建客户端

两方使用socket(套接字 ，通信端点) 进行交流

ServerSocket

用于创建服务器 。创建完毕后，会绑定一个端口号。

然后此服务器可以等待客户端连接 。

每连接一个客户端 ，服务器就会得到一个新的Socket对象，用于跟客户端进行通信 。

常用构造方法:

```
ServerSocket(int port); ****
```

创建一个基于TCP/IP协议的服务器 ，并绑定指定的端口号。

注意：参数port的范围是： 0-65535 （建议1025-65535）

常用方法:

```
Socket accept(); ****
```

等待客户端连接 。

此方法会导致线程的阻塞！

直到一个新的客户端连接成功，return Socket对象后，线程在继续执行。

```
void close();
```

释放占用的端口号 ，关闭服务器。

Socket

是两台计算机之间通信的端点，是网络驱动提供给应用程序编程的一种接口 一套标准，一种机制。

构造方法:

```
Socket(String ip,int port) ****
```

创建一个套接字，并连接指定ip和端口号的 服务器。

参数1. 服务器的ip地址

参数2. 服务器软件的端口号..

常用方法:

```
- OutputStream getOutputStream();
```

返回的是，指向通信的另一端点的输出流

```
- InputStream getInputStream();
```

返回的是，指向通信的另一端点的输入流

```
- void close();
```

关闭套接字

注意:

在网络编程时，获取输入输出流的操作，对于客户端与服务器来说是相对的

客户端的输入流，输入的是服务器的输出流 输出的内容。

客户端的输出流，输出到了服务器的输入流中。

所以 在使用时，需要注意以下一点规则：

客户端与服务器获取流的顺序必须是相反的：

例如：

客户端先得到了输入流，那服务器必须先获取输出流

案例 echo程序

客户端：

```
//1. 连接服务器
Socket socket = new Socket("192.168.102.228",8888);
//2. 得到输出流
//2.1 得到输出流
OutputStream os = socket.getOutputStream();
//2.2 将输出流，转换为打印流
PrintStream ps = new PrintStream(os);
//3. 得到输入流
//3.1 得到输入流
InputStream is = socket.getInputStream();

//3.2 将字节输入流，转换为字符输入流，并转换为逐行读取流
```

```

InputStreamReader isr = new InputStreamReader(is);
BufferedReader br = new BufferedReader(isr);

//4.    循环接收用户输入
Scanner input = new Scanner(System.in);
while(true) {
    System.out.println(""请输入要发送给服务器的内容:");
    String text1 = input.nextLine();
    //5.    将用户输入的内容, 发送给服务器
    ps.println(text1);
    //6.    接收服务器回复的消息
    String text2 = br.readLine();
    System.out.println(text2);
    if("&886&quot;.equals(text1)) {
        break;
    }
}
}

```

服务器:

```

public static void main(String[] args) throws Exception {
    //1.    启动服务器, 并侦听8888端口号
    ServerSocket server = new ServerSocket(8888);
    //2.    打印提示
    System.out.println("&quot;服务器已启动 , 等待客户端连接中...&quot;);
    //3.    等待客户端连接
    Socket socket = server.accept();
    System.out.println("&quot;一个客户端连接成功:&quot;+socket.getInetAddress().toString());
    //4.    获取输入流
    //4.1    获取输入流
    InputStream is = socket.getInputStream();
    //4.2    将输入的字节流 ,转换为字符流
    InputStreamReader isr = new InputStreamReader(is);
    //4.3    将字符流, 转换为逐行读取流
    BufferedReader br = new BufferedReader(isr);

    //5.    获取输出流
    //5.1    获取输出流
    OutputStream os = socket.getOutputStream();
    //5.2    将字节输出流, 转换为打印流
    PrintStream ps = new PrintStream(os);
    while(true) {
        //6.    循环读取一行行的数据 ,读取操作会导致线程的阻塞, 直到客户端真的发送了数据,
        //服务器才能接到, 顺序继续执行下面的代码
        String text = br.readLine();
        //7.    将这个文字, 再打印给客户端
        ps.println("&quot;服务器:&quot;+text);
        if("&886&quot;.equals(text)) {
            break;
        }
    }
}

```

UDP 协议(数据报) 程序 了解

用户数据报协议，与tcp协议不同，UDP的连接是不可信的。数据发送的成功与失败 与 数据报是无关的。

使用到两个类：

1.数据报套接字: DatagramSocket

用于发送 与 接收数据包的Socket

- 构造方法：
 - DatagramSocket(int port);
 - 参数：端口号
- 常用方法：
 - close() : 关闭套接字。
 - send(DatagramPacket dp)
将一个数据包dp 发送出去
 - receive(DatagramPacket dp)
接收一个数据包，并存储到参数dp中。

2.数据包 DatagramPacket

用于发送或接收数据时，盛放数据的对象！

- 构造方法：

1.用于发送数据时，组装数据的 构造方法。

DatagramPacket(byte[] bytes,int startIndex,int len,InetAddress ip,int port);

- 参数1. 要发送的数据，是字节数组的形式
- 参数2. 有效数据 在数组中的起始位置
- 参数3. 有效数据 在数组中的长度
- 参数4. 当前这个数据包，准备发送到的IP地址，InetAddress 这个类的对象，用于描述

IP 。

得到InetAddress对象的方式：

- InetAddress ip = InetAddress.getByName("192.168.102.228");

参数5. 当前这个数据包，准备发送到目标计算机的哪个端口号。

2.用于接收数据时，存储数据的 构造方法。

创建的是 不包含数据的数据包，用于在接收到数据后，存储数据 ！

DatagramPacket(byte[] bytes,int len)

- 参数1. 用于存储数据的 数组
- 参数2. 允许存储的最大长度

- 常用方法：

byte[] getData()

用于获取数据包中的有效字节数组

```
int getLength
```

用于获取数据包中的有效数据的长度。

InetAddress 描述IP地址的类

InetAddress 这个类的对象，用于描述IP 。

得到InetAddress对象的方式：

```
InetAddress ip = InetAddress.getByName("192.168.102.228");
```

在UDP协议中，通过数据包DatagramPacket的getAddress方法， 可以得到数据包来自哪个ip

在TCP协议中，通过套接字Socket的getInetAddress方法，可以得到套接字连接的ip地址。

- 常用方法：

```
1.String getHostAddress()
```

ip地址字符串

```
2.String getHostName()
```

计算机名称， 当名称无法获取时，获取的为ip地址。

URL 类 (统一资源定位符)(网址) 了解

案例1. 下载文件

```
public class ClientDemo {
    public static void main(String[] args) throws Exception {
        Scanner input = new Scanner(System.in);
        System.out.println("欢迎使用嘿嘿雷下载器");
        System.out.println("请输入要下载的文件网址:");
        String urlString = input.nextLine();
        System.out.println("请输入要保存的文件名称:");
        System.out.println("文件默认下载位置:d盘download文件夹中");
        String fileString = input.nextLine();

        //1. 确保文件夹存在
        File dir = new File("d://download");
        if(!dir.exists()) {
            dir.mkdirs();
        }
        //2. 创建一个文件输出流，用于输出数据
        FileOutputStream fos = new FileOutputStream(new File(dir,fileString));

        //3. 今天学习的新内容
```

```

//3.1 创建一个网址对象(统一资源定位符)
URL url = new URL(urlString);
//3.2 打开链接 , 并得到链接对象
URLConnection conn = url.openConnection();
//3.3 通过连接对象, 获取连接到的文件的输入流
InputStream is = conn.getInputStream();
//[3.4] 获取网址指向文件的 大小
long fileLength = conn.getContentLengthLong();
//3.5 循环读取 并写出到fos中
//用于存储 每次读取的数据
byte[] bytes = new byte[1024*1024];
//用于存储每次读取的数据长度
int len = -1;
//用于存储已读取的所有数据的长度
int count = 0;
while((len = is.read(bytes))!=-1) {
    //将每次循环读取的bytes 写出到文件中
    fos.write(bytes,0,len);
    count+=len;
    System.out.println(""下载中:"+(count/(fileLength/100))+"%"");
}
is.close();
System.out.println(""文件下载完毕"");
}
}

```

案例2. 传输参数, 并下载数据

```

static Scanner input = new Scanner(System.in);
public static void main(String[] args) throws Exception {
    System.out.println(""自动P图小程序:"");
    System.out.println(""请选择菜单:"");
    int menu = menu();
    System.out.println(""请输入名字:"");
    String name = input.nextLine();
    String name2 = URLEncoder.encode(name, "&quot;UTF-8&quot;");
    //1. 得到网址, 这个网址指向的内容 是另一个图片的网址
    String urlString = "&quot;http://itdage.cn/B/img?id=&quot;+menu+"&quot;.&amp;s1=&quot;+name2;
    URL url = new URL(urlString);
    //2. 打开链接
    URLConnection conn = url.openConnection();
    //3. 得到输入流
    //3.1 因为我们这个网址的内容 只是一个图片的地址, 也就是一行文字, 所以我们将这个流转换为逐行读取流,
    读取一行文本就可以了
    InputStream is = conn.getInputStream();
    //3.2 转换为字符流
    InputStreamReader isr = new InputStreamReader(is);
    //3.3 转换为逐行读取流
    BufferedReader br = new BufferedReader(isr);
}

```

```

String imgUrlString = br.readLine();
System.out.println(""图片已制作完成 , 地址:"+imgUrlString);

}
public static int menu(){
    System.out.println(""1. 捐*补助");");//1
    System.out.println(""2. 登月插旗");");//2 娃娃
    System.out.println(""3. 娃娃订单");");//3 娃娃
    System.out.println(""4. 相思癌");");//4相思癌
    System.out.println(""5. 孕检证明");");
    System.out.println(""6. 玛莎拉蒂订单");");//6.玛莎拉蒂订单
    System.out.println(""7. 马云湖畔大学");");//
    System.out.println(""122. 男举牌 娶你");");//
    System.out.println(""123. 女举牌 生猴子");");//
    String text = input.nextLine();
    int m = -1;
    try {
        m = Integer.parseInt(text);
    }catch(Exception e) {

    }
    if(m<1 || (m>7&&m<122) || m>123) {
        return menu();
    }
    return m;
}

```