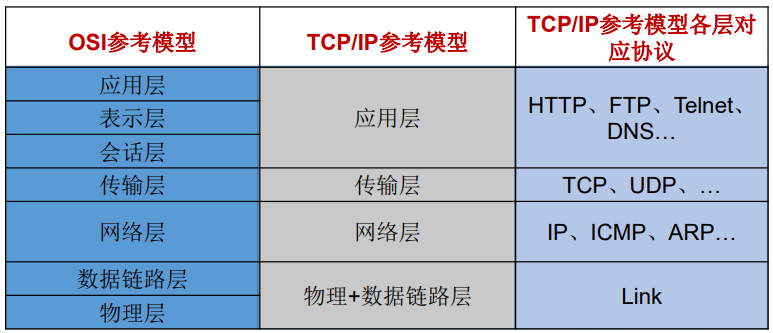
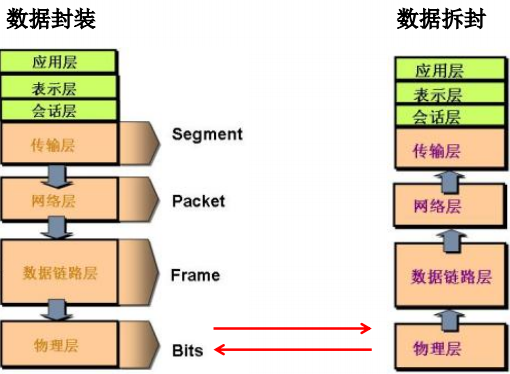
**网络中主机相互通信：**

1. IP、端口号
2. 通信协议（OSI参考模型、TCP/IP参考模型）
3. 
4. 

**IP地址：InetAdress**

**端口号：标识运行的进程**

**InetAdress类：**

1、Internet上的主机有两种方式表示地址：

1、 域名(hostName)：www.atguigu.com

2、 IP 地址(hostAddress)：202.108.35.210

2、两个子类：Inet4Address、Inet6Address

3、InetAddress 类 对 象 含 有 一 个 Internet 主 机 地 址 的 域 名 和 IP 地 址

4、提供了如下几个静态方法来获取InetAddress实例

public static InetAddress getLocalHost()

public static InetAddress getByName(String host)

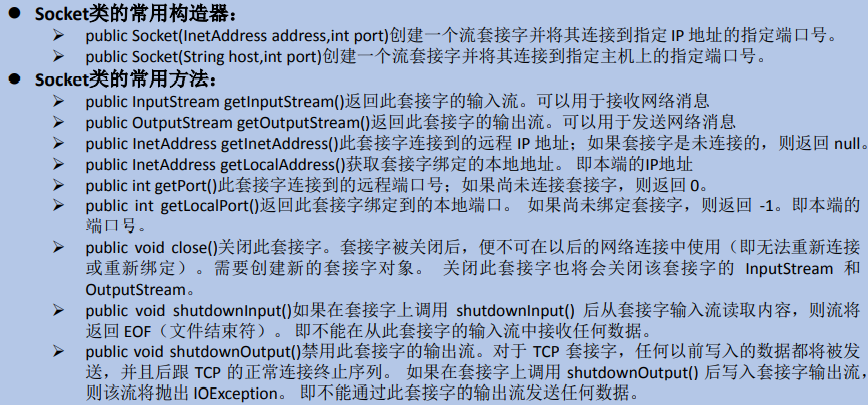
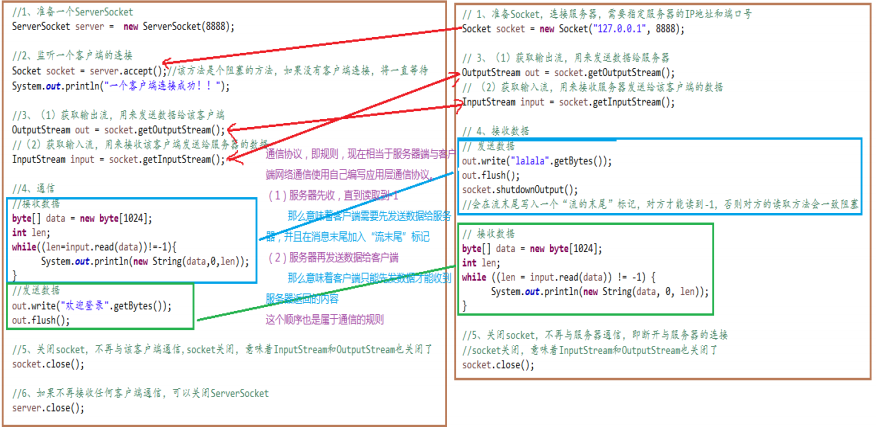
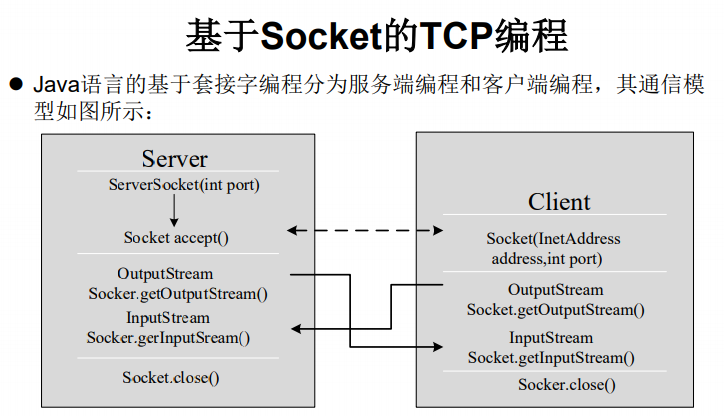
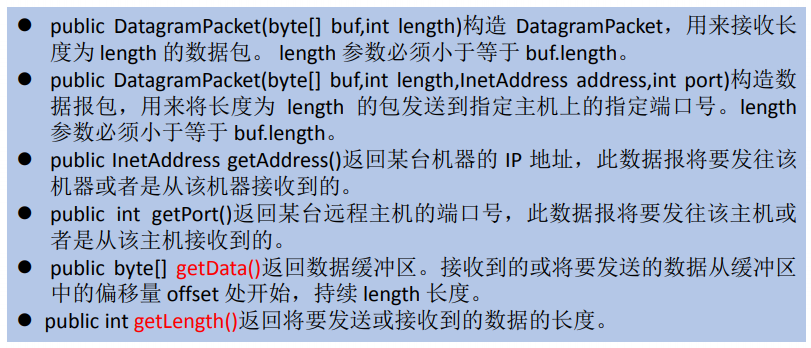
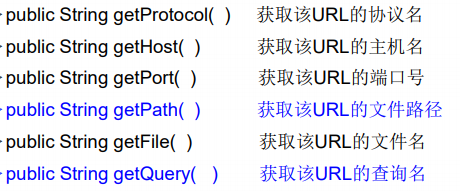
1. 常用方法：

public String getHostAddress()：返回 IP 地址字符串（以文本表现形式）。

public String getHostName()：获取此 IP 地址的主机名

public boolean isReachable(int timeout)：测试是否可以达到该地址

网络协议：

1. 在制定协议时，把复杂成份分解成一些简单的成份，再将它们复合起来。最常用的复合方式是层次方式，即同层间可以通信、上一层可以调用下一层，而与再下一层不发生关系。各层互不影响，利于系统的开发和扩展。
   1. 传输控制协议TCP(Transmission Control Protocol)
   2. 用户数据报协议UDP(User Datagram Protocol)。
2. TCP协议：
   1. 使用TCP协议前，须先建立TCP连接，形成传输数据通道
   2. 传输前，采用“三次握手”方式，点对点通信，是可靠的
   3. TCP协议进行通信的两个应用进程：客户端、服务端。
   4. 在连接中可进行大数据量的传输
   5. 传输完毕，需释放已建立的连接，效率低
3. UDP协议：
   1. 将数据、源、目的封装成数据包，不需要建立连接
   2. 每个数据报的大小限制在64K内
   3. 发送不管对方是否准备好，接收方收到也不确认，故是不可靠的
   4. 可以广播发送
   5. 发送数据结束时无需释放资源，开销小，速度快
4. Scoket
   1. 网络上具有唯一标识的IP地址和端口号组合在一起才能构成唯一能识别的标识符套接字。
   2. 通信的两端都要有Socket，是两台机器间通信的端点。
   3. 网络通信其实就是Socket间的通信。
   4. Socket允许程序把网络连接当成一个流，数据在两个Socket间通过IO传输。
   5. 一般主动发起通信的应用程序属客户端，等待通信请求的为服务端。
   6. 分类：
      1. 流套接字（stream socket）：使用TCP提供可依赖的字节流服务
      2. 数据报套接字（datagram socket）：使用UDP提供“尽力而为”的数据报服务
      3. 
5. 
6. UDP：
   1. 类 DatagramSocket 和 DatagramPacket 实现了基于 UDP 协议网络程序。
   2. UDP数据报通过数据报套接字 DatagramSocket 发送和接收，系统不保证UDP数据报一定能够安全送到目的地，也不能确定什么时候可以抵达。
   3. DatagramPacket 对象封装了UDP数据报，在数据报中包含了发送端的IP地址和端口号以及接收端的IP地址和端口号。
   4. UDP协议中每个数据报都给出了完整的地址信息，因此无须建立发送方和接收方的连接。如同发快递包裹一样。
   5. 
   6. 
   7. 
7. URL：
   1. 基本构成：
      1. <传输协议>://<主机名>:<端口号>/<文件名>#片段名?参数列表
      2. URL类构造器：
         1. 
      3. 常用方法：
         1. 
      4. 针对http协议的URLConnection类
         1. URL的方法 openStream()：能从网络上读取数据
         2. 表示到URL所引用的远程对象的连接。当与一个URL建立连接时，首先要在一个 URL 对象上通过方法 openConnection() 生成对应的 URLConnection对象。如果连接过程失败，将产生IOException.
         3. 