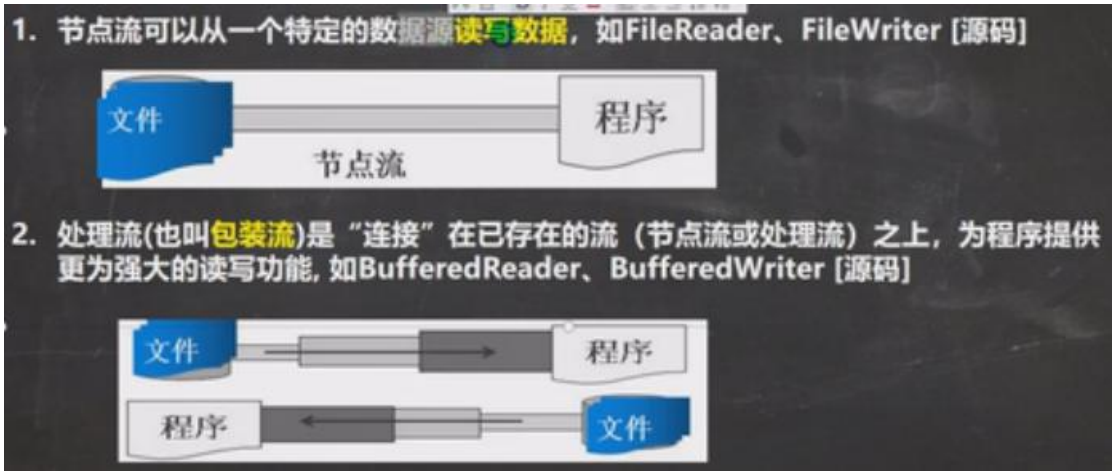
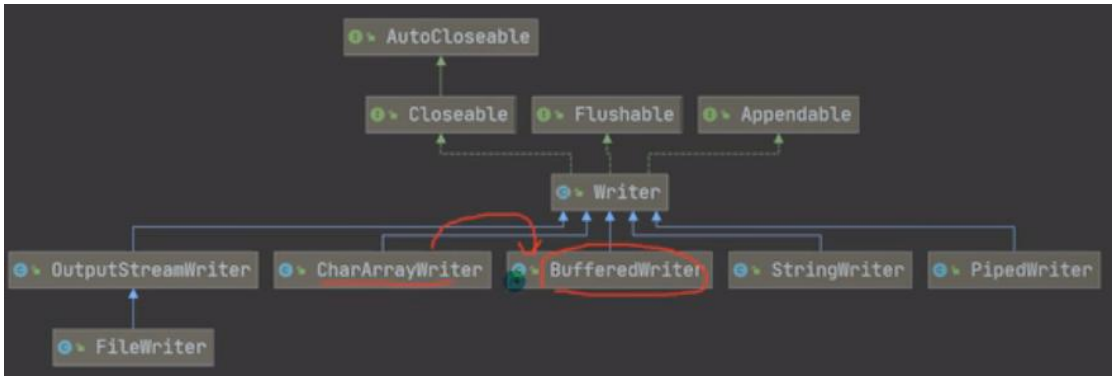
节点流和处理流

1. 节点流可以从一个特定的数据源读取数据，如：FileReader、FileWriter
2. 处理流（也叫包装流）是“连接”在已存在的流（节点流或处理流）之上，为程序提供更为强大的读写功能，如 BufferedReader、BufferedWriter





节点流和处理流的区别和联系：

（1）节点流是底层流/低级流，直接跟数据源相连接；

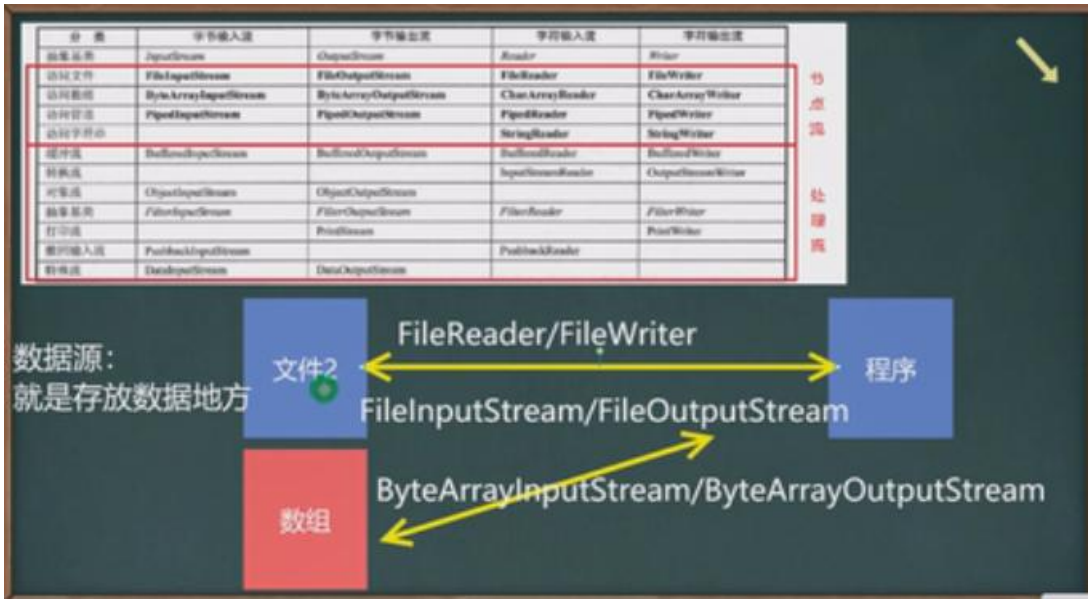
（2）处理流（包装流）包装节点流，既可以消除不同节点流的实现差异，也可以提供更方便的方法来完成输入和输出；

（3）处理流（也叫包装流）对节点流进行包装，使用了修饰器设计模式，不会直接与数据源相连接。

处理流的功能主要体现在以下两个方面：

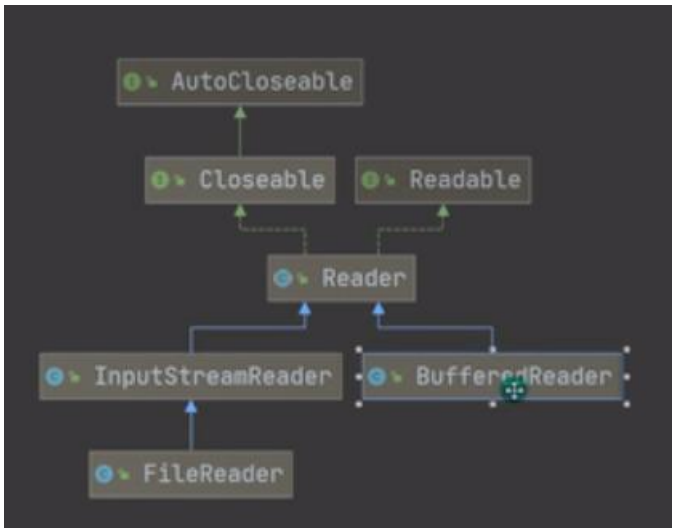
（1）性能的提高：主要以增加缓冲的方式来提高输入输出的效率；

（2）操作的便捷：处理流可能提供了一系列便捷的方法来一次性输入输出大批量的数据，使用更加灵活方便。



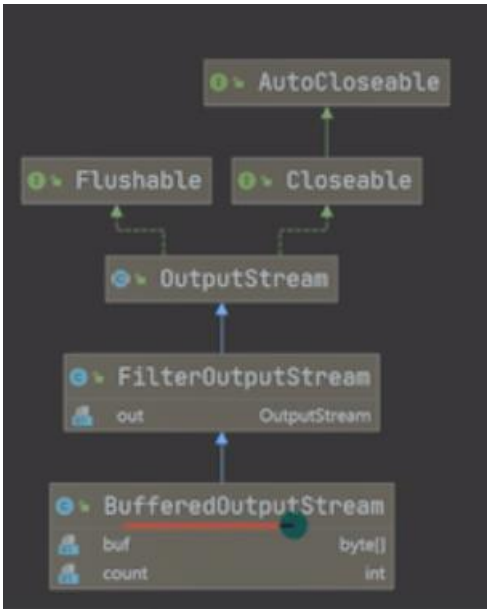
处理流----BufferedReader 和 BufferedWriter

BufferedReader 和 BufferedWriter 属于字符流，是按照字符来读取数据的；其关闭时，只需关闭外层流即可

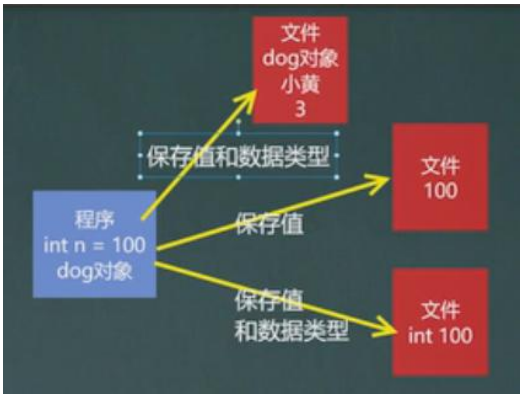


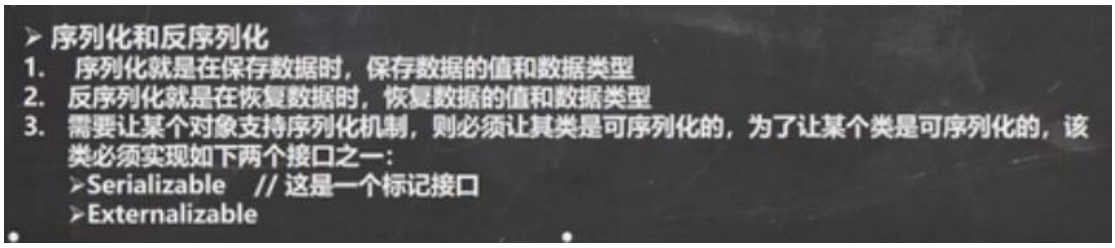
BufferedReader 和 BufferedWriter 是按照字符串操作的，不要去操作二进制文件（视频、图片等），可能会造成文件损坏

BufferedOutputStream 的类关系继承图



对象流----ObjectInputStream 和 ObjectOutputStream





序列化和反序列化的注意事项和细节：
（1）读写顺序要一致；
（2）要求序列化或反序列化的对象，需要实现 Serializable 接口；
（3）序列化的类中建议添加 SerialVersionUID，为了提高版本的兼容性；
（4）序列化对象时，默认将里面所有属性都进行了序列化，但除了 static 或 transient 修饰的成员；
（5）序列化对象时，要求里面属性的类型也要实现序列化接口；
（6）序列化具有可继承性，也就是如果某类已经实现了序列化，则它的所有子类也已经默认实现了序列化。

标准输入输出流：

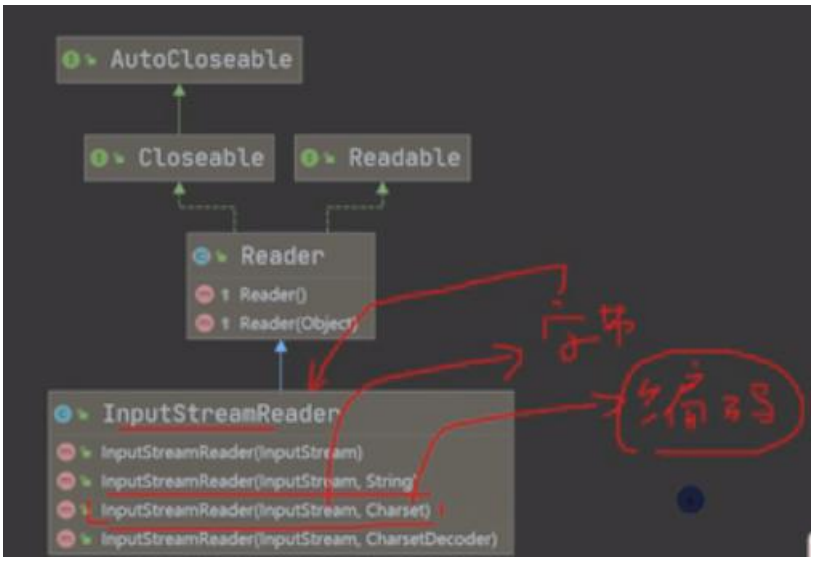
	类型	默认设备
System.in 标准输入	InputStream	键盘
System.out 标准输出	PrintStream	显示器

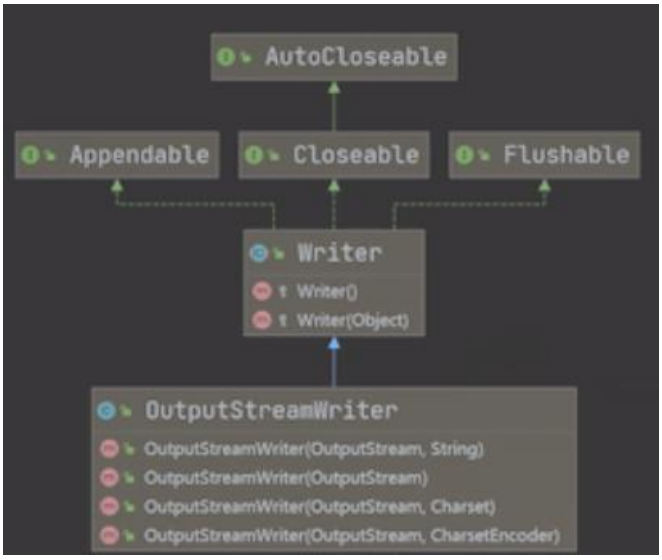
```
//System 类 的 public final static InputStream in = null;
// System.in 编译类型   InputStream
// System.in 运行类型   BufferedInputStream
// 表示的是标准输入 键盘
System.out.println(System.in.getClass());

//老韩解读
//1. System.out public final static PrintStream out = null;
//2. 编译类型 PrintStream
//3. 运行类型 PrintStream
//4. 表示标准输出 显示器
System.out.println(System.out.getClass());
```

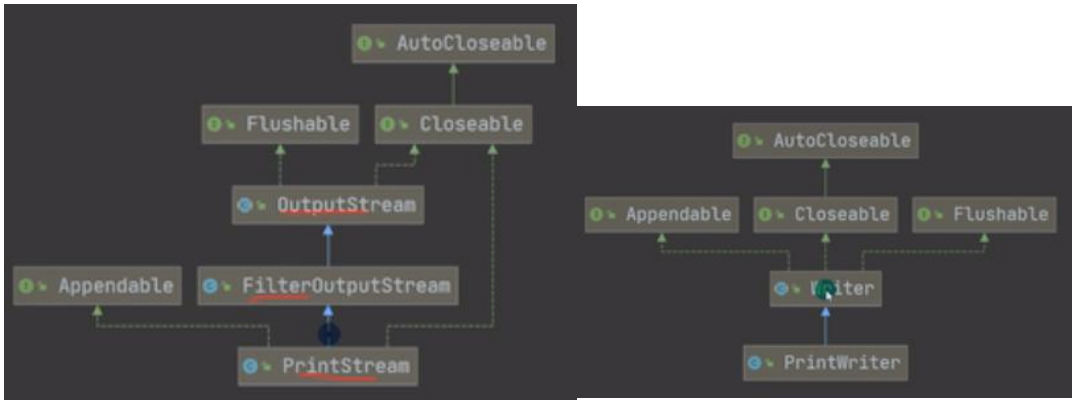
转换流 ---（字节流转换为字符流）InputStreamReader 和 OutputStreamWriter

1. InputStreamReader:Reader的子类，可以将InputStream(字节流)包装成Reader(字符流)
2. OutputStreamWriter:Writer的子类，实现将OutputStream(字节流)包装成Writer(字符流)
3. 当处理纯文本数据时，如果使用字符流效率更高，并且可以有效解决中文问题，所以建议将字节流转换成字符流
4. 可以在使用时指定编码格式(比如 utf-8, gbk , gb2312, ISO8859-1 等)

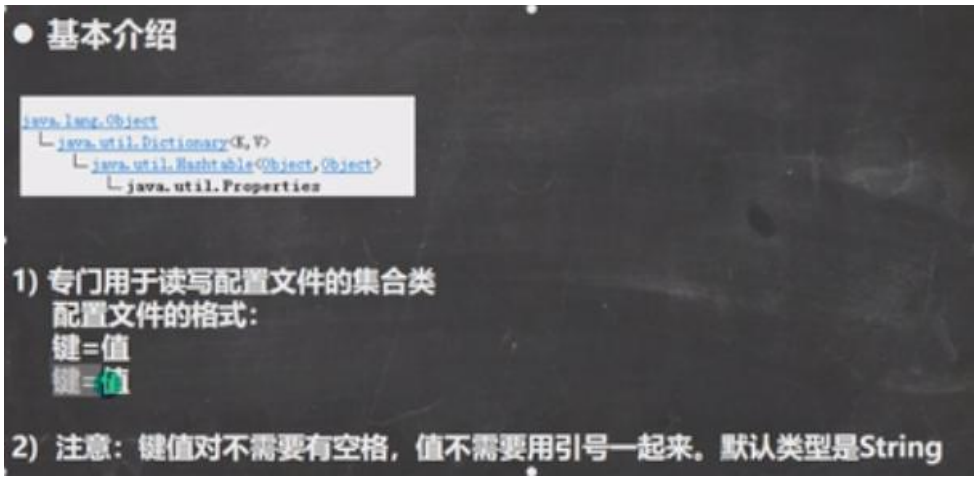




打印流----PrintStream 和 PrintWriter（打印流只有输出流，没有输入流）



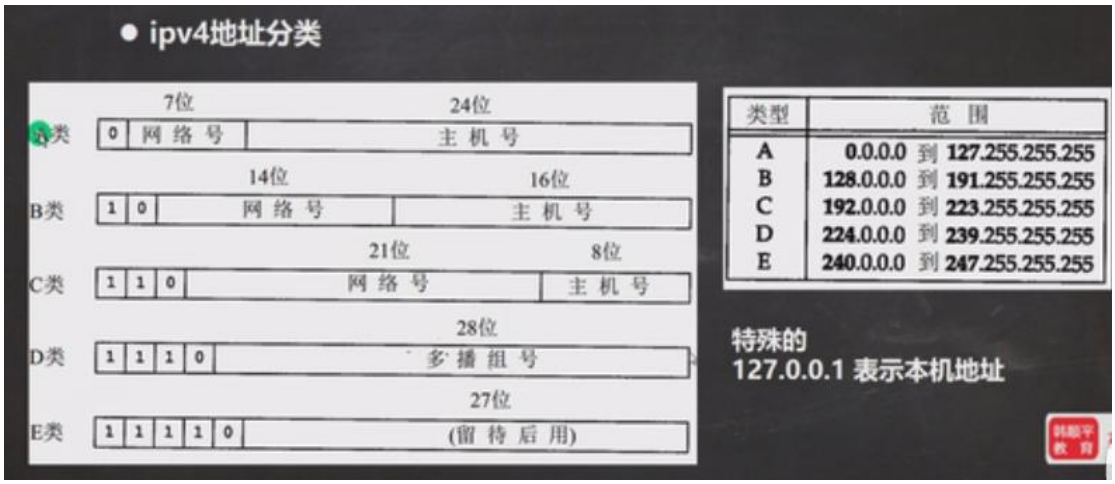
Properties 类



网络的相关概念
网络通信：
（1）概念：两台设备之间通过网络实现数据传输
（2）网络通信：将数据通过网络从一台设备传输到另一台设备
（3）Java.net 包下提供了一系列的类或接口，供使用，完成网络通信

网络：
（1）概念：两台或多台设备通过一定的物理设备连接起来构成了网络
（2）根据网络覆盖范围的不同，对网络进行分类
（3）局域网：覆盖范围最小，仅仅覆盖一个教室或一个机房
（4）城域网：覆盖范围较大，可以范围一个城市
（5）广域网：覆盖范围最大，可以覆盖全国乃至全球，万维网是广域网的代表

IP 地址：
（1）概念：用于唯一标识网络中的每台计算机/主机
（2）查看 IP 地址：ipconfig
（3）IP 地址的表示形式：点分十进制 xx.xx.xx.xx
（4）每个十进制数的范围：0～255
（5）ip 地址的组成 = 网络地址 + 主机地址，比如：192.168.16.69
（6）IPV6 是互联网工程任务组设计的用于替代 IPV4 的下一代 IP 协议，其地址数量号称是可以为全世界的每一粒沙子编上一个地址
（7）由于 IPV4 最大的问题在于网络地址资源有限，严重制约了互联网的应用和发展；IPV6 的使用不仅能解决网络地址资源数量的问题，而且也解决了多种接入设备连入互联网的障碍



域名：

（1）www.baidu.com

（2）好处：为了方便记忆，解决记 IP 困难

（3）概念：将 IP 地址映射成域名

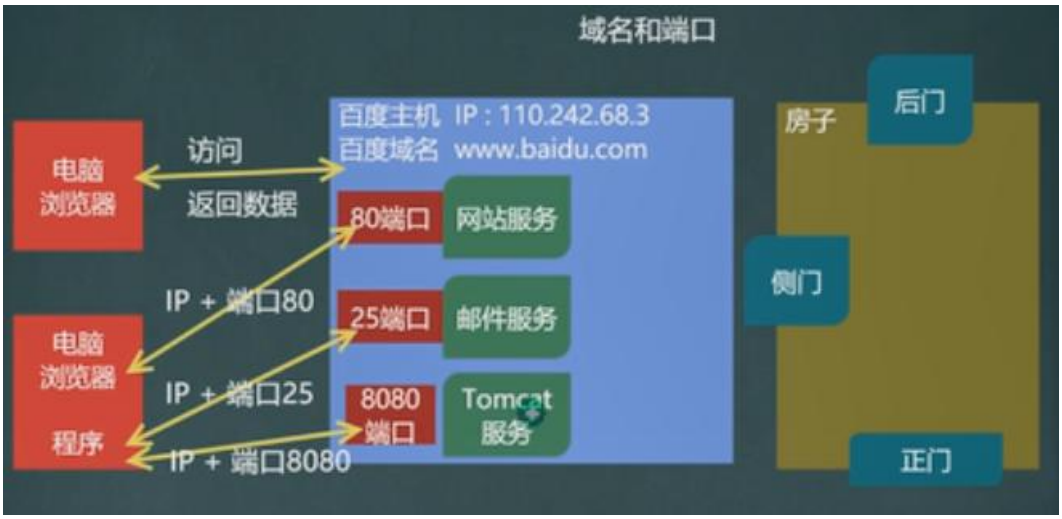
端口号：

（1）概念：用于标识计算机上某个特定的网络程序

（2）表示形式：以整数形式，范围 0~65535[2 个字节表示端口 0~2^16-1]

（3）0~1024 已经被占用，比如 ssh 22，ftp 21，smtp 25，http 80

（4）常见的网络程序端口号：tomcat：8080；mysql：3306；oracle：1521；sqlserver：1433



网络通信协议：

协议（tcp/ip）：TCP/IP（Transmission Control Protocol/Internet Protocol）的简写中文译名为：传输控制协议/因特网互联协议，又叫网络通讯协议，这个协议是 Internet 最基本的协议、Internet 国际互联网的基础，简单来说是由网络层的 IP 协议和传输层的 TCP 协议组成的

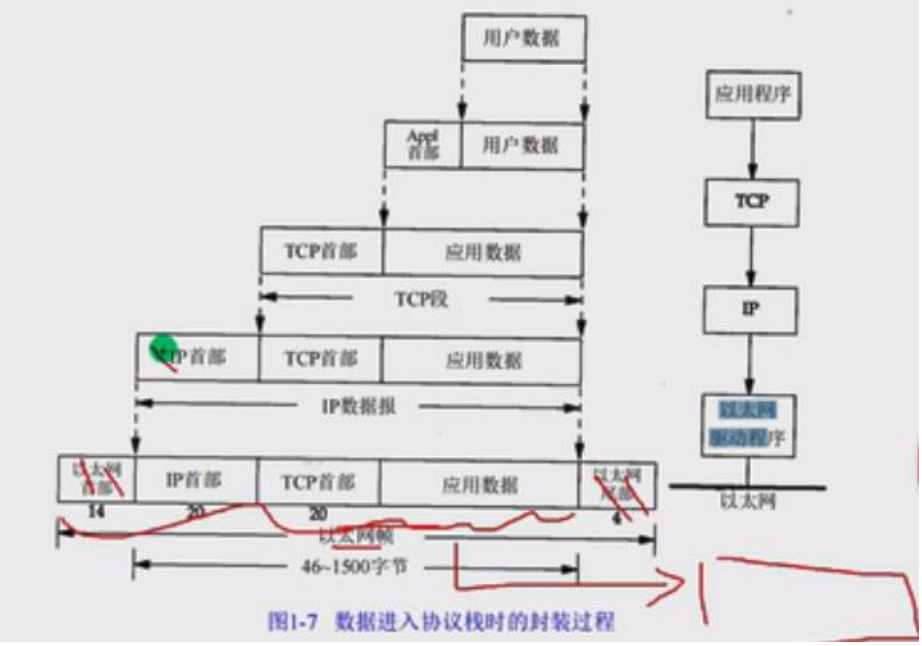


图1-7 数据进入协议栈时的封装过程

TCP 和 UDP：

TCP 协议：传输控制协议

（1）使用 TCP 协议签，需先建立 TCP 连接，形成传输数据通道

（2）传输前，采用“三次握手”方式，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）在连接中可以进行大数据量的传输

（5）传输完毕，需要释放已经建立的连接，效率低

UDP 协议：用户数据协议

（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 以内

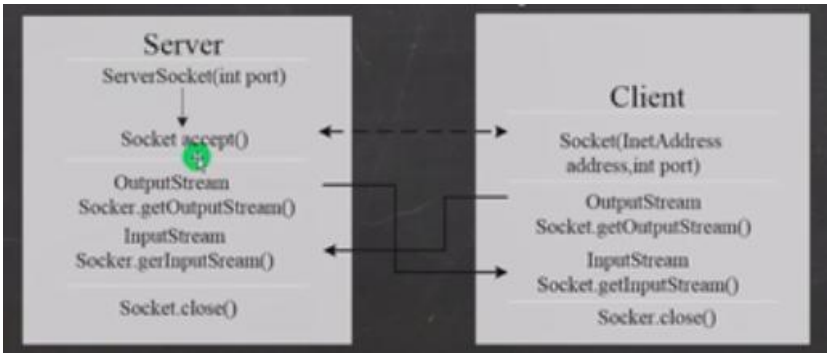
（3）因无需连接，故是不可靠的

（4）发送数据结束时无需释放资源（因为不是面向连接的），速度快

InetAddress 类
相关方法：
（1）获取本机 InetAddress 对象 getLocalHost
（2）根据指定主机名/域名获取 ip 地址对象 getByName
（3）获取 InetAddress 对象的主机名 getHostName
（4）获取 InetAddress 对象的地址 getHostAddress

Socket（套接字）
（1）套接字开发网络应用程序被广泛采用，以至于成为事实上的标准
（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于客户端，等待通信请求的为服务端

TCP 网络编程
（1）基于客户端---服务端的网络通信
（2）底层使用的是 TCP/IP 协议
（3）应用场景举例：客户端发送数据，服务端接受并显示控制台
（4）基于 Socket 的 TCP 编程



TCP 网络通信编程

netstat 指令：

（1）netstat -an 可以查看当前主机网络情况，包括端口监听情况和网络连接情况

（2）netstat -an | more 可以分页显示

（3）要求在 dos 控制台下执行

说明：

（1）Listening 表示某个端口在监听

（2）如果有一个外部程序（客户端）连接到该端口，就会显示一条连接信息

UDP网络通信编程[了解]

- 基本介绍

1. 类 DatagramSocket 和 DatagramPacket[数据包/数据报] 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