Collections工具类：

static sort(List l)给List排序；但是List集合中的元素必须是“可排序”的，实现comparable接口。

将List集合转化为线程安全的static synchronizedList(List l);

泛型：

如果集合中不使用泛型，则集合中的元素类型不统一，在遍历集合的时候，只能拿出来Object类型，需要做大量的强制类型转换，麻烦。

优点：统一集合中存储元素的类型，

缺点：集合中只能存储一种元素。

泛型是一个编译阶段的语法，在编译阶段统一集合的类型。

增强for循环：

for(类型 变量 ：数组名/集合名)｛｝

集合要想使用增强for循环这种语法，集合需要使用泛型。

缺点：没有下标。

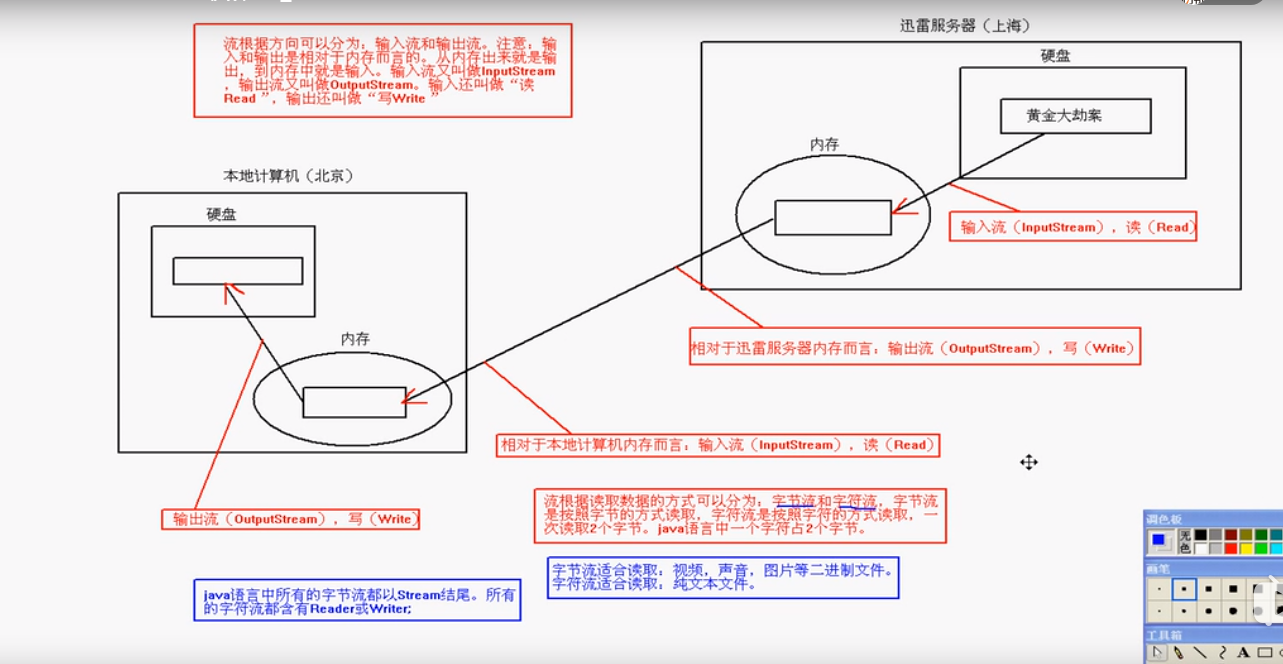
流根据方向可以分为输入流和输出流，从内存出来就是输出，到内存则为输入。输入流叫做InputStream，也叫作“读”即“Read”。输出流叫做“OutputStream”，也叫作“写”即“Write”；

流根据读取数据的方式可以分为字节流和字符流，字节流是按照字节的方式读取，字符流是按照字符的方式读取（一次读取2个字节），java语言中一个字符占2个字节。

字节流适合读取：视频，声音，图片等二进制文件。

字符流适合读取：纯文本文件。

java语言中所有的字节流都以Stream结尾，所有的字符流都含有Reader或者Writer。



需要掌握的16个流：

FileInputStream

FileOutputStream

FileReader

FileWriter

BufferedReader

BufferedWriter

BufferedInputStream

BufferedOutputStream

DataInputStream

DateOutputStream

ObjectInputStream

ObjectOutputStream

转换流（字节流转换成字符流）

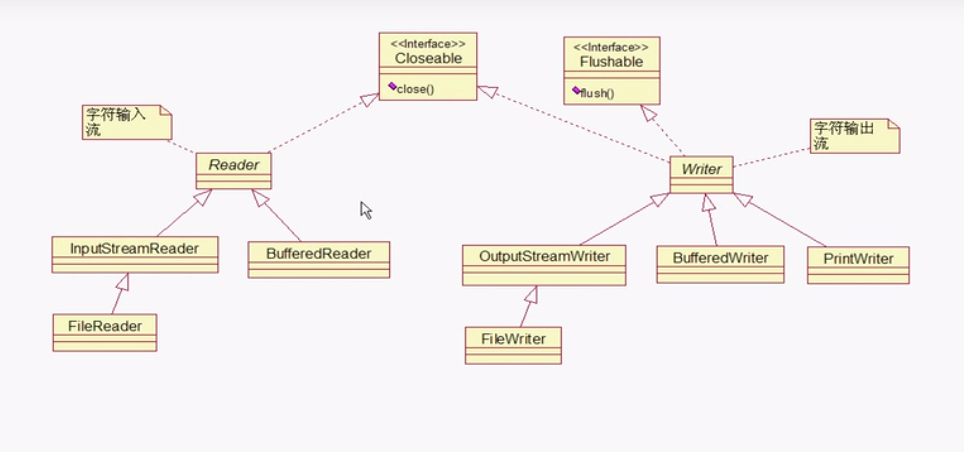
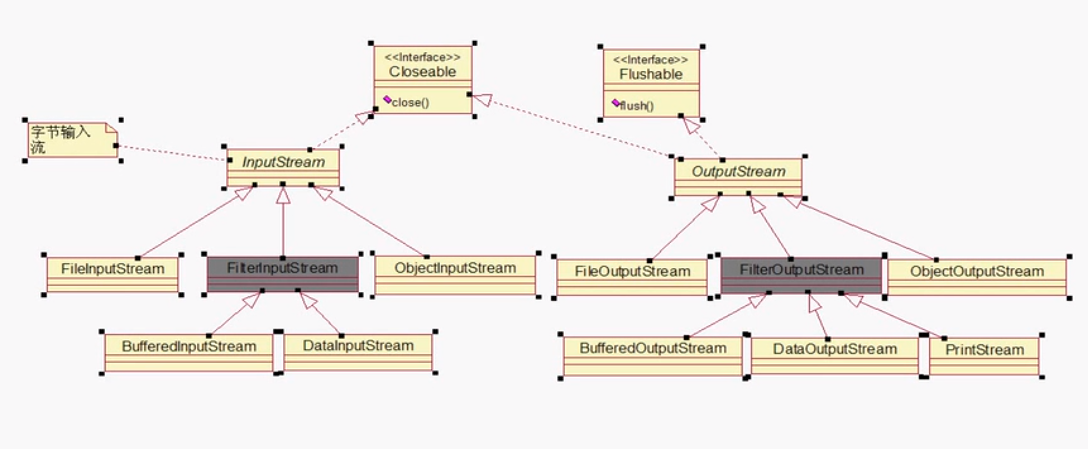
InputStreamReader

OutputStreamWriter

PrintWriter

PrintStream //标准的输出流（默认输出到控制台）

java语言中流分为四大家族：InputStream,OutputStream,Reader,Writer



FileInputStream:读取文件

要读取文件，先要创建一个流：FileInputStream f = new FileInputStream(文件路径);

int read()：读取文件的一个字节，读完之后游标往后移动读下一个。读到文件没有字节则会返回-1。

int read(byte[] bytes):读取文件把文件的数据存到bytes数组中，返回的int表示数组中存了几个字节。

在程序的结尾要关闭流：f.close();

int available();返回流中剩余的估计字节数。

void skip(int n);跳过n个字节不读取。

FileOutputStream:向文件中写入数据

先创建流：FileOutputStream f = new FileOutputStream(文件路径，boolean 是否追加写入);

void write(byte[]);向文件写入byte数组。

flush();强制写入

close();关闭流。

FileReader和FileInputStream类似。

FileWriter和FileOutputStream类似。write方法可以直接传String。