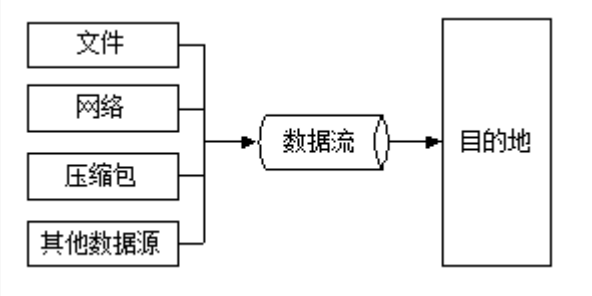
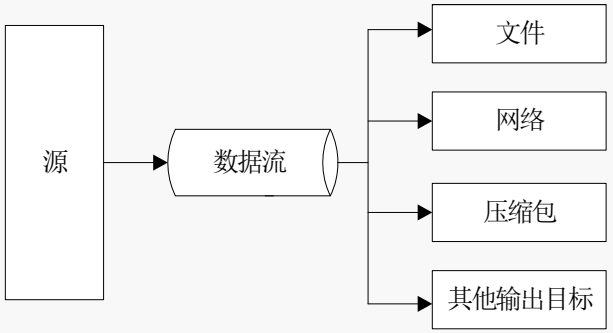
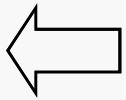
流概述

流是一组有序的数据序列，根据操作的类型，可分为输入流和输出流两种。I/O（Input/Output）流提供了一条通道程序，可以使用这条通道把源中的字节序列送到目的地。虽然I/O流经常与磁盘文件存取有关，但是程序的源和目的地也可以是键盘、鼠标、内存或显示器窗口等。

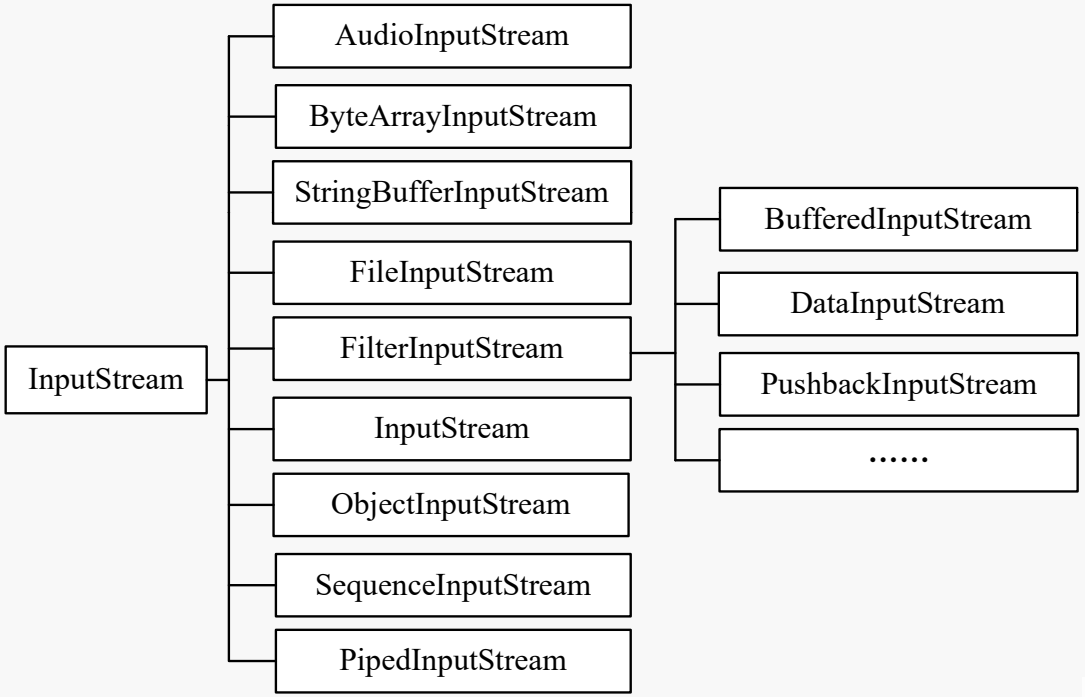






输入流

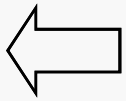
InputStream类是字节输入流的抽象类，是所有字节输入流的父类。InputStream类的具体层次结构如图所示：



输入流

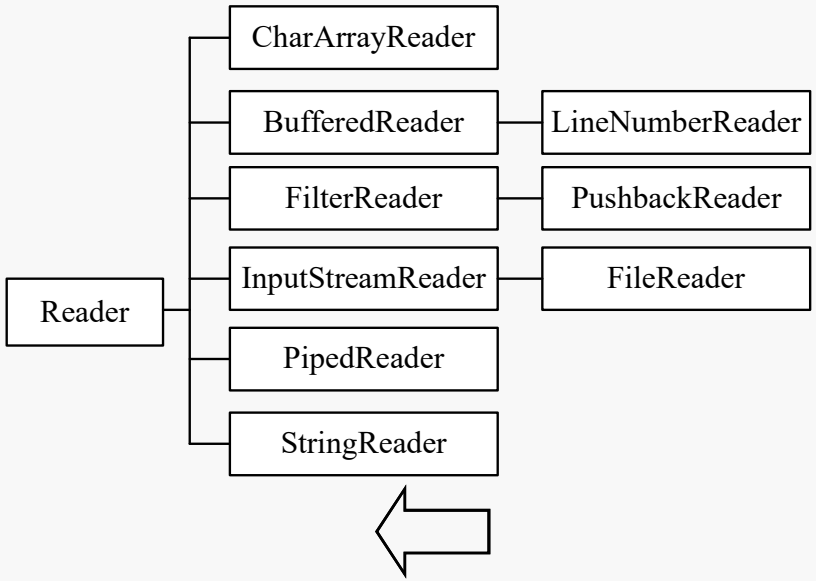
该类中所有方法遇到错误时都会引发IOException异常。下面是对该类中的一些方法的简要说明如下表所示。

方法	说明
read()	从输入流中读取数据的下一个字节。返回 0~255 范围内的 int 字节值。如果因为已经到达流末尾而没有可用的字节，则返回值-1
read(byte[] b)	从输入流中读入一定长度的字节，并以整数的形式返回字节数
mark(int readlimit)	在输入流的当前位置放置一个标记，readlimit 参数告知此输入流在标记位置失效之前允许读取的字节数
reset()	将输入指针返回到当前所做的标记处
skip(long n)	跳过输入流上的 n 个字节并返回实际跳过的字节数
markSupported()	如果当前流支持 mark()/reset()操作就返回 true
close()	关闭此输入流并释放与该流关联的所有系统资源



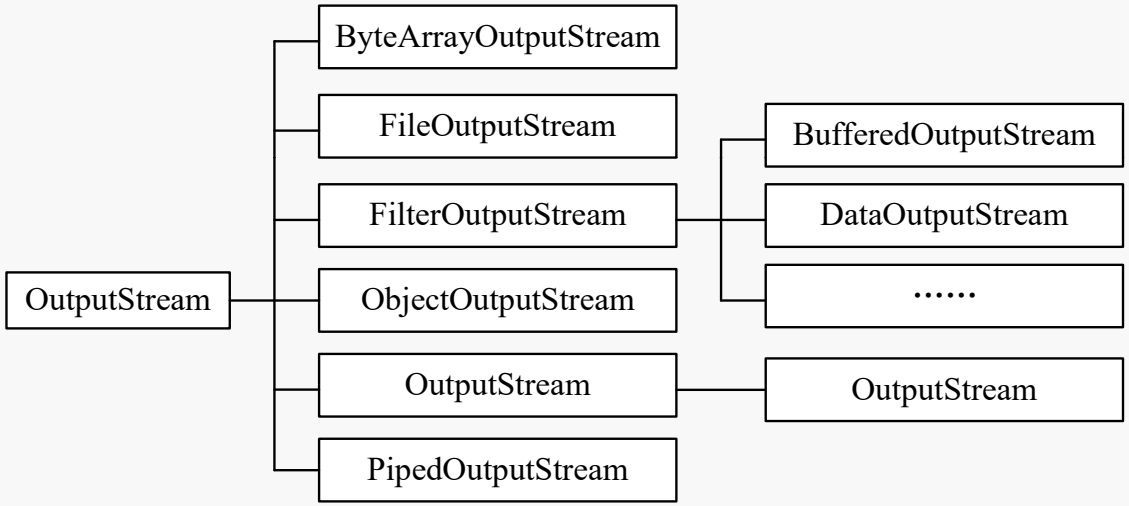
输入流

Java中的字符是Unicode编码，是双字节的。InputStream是用来处理字节的，在处理字符文本时不是很方便。Java为字符文本的输入提供了专门一套单独的类Reader，但Reader类并不是InputStream类的替换者，只是在处理字符串时简化了编程。Reader类是字符输入流的抽象类，所有字符输入流的实现都是它的子类，Reader类的具体层次结构如下图所示：



输出流

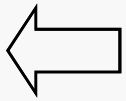
OutputStream类是字节输入流的抽象类，此抽象类是表示输出字节流的所有类的超类。OutputStream类的具体层次如图所示：



输出流

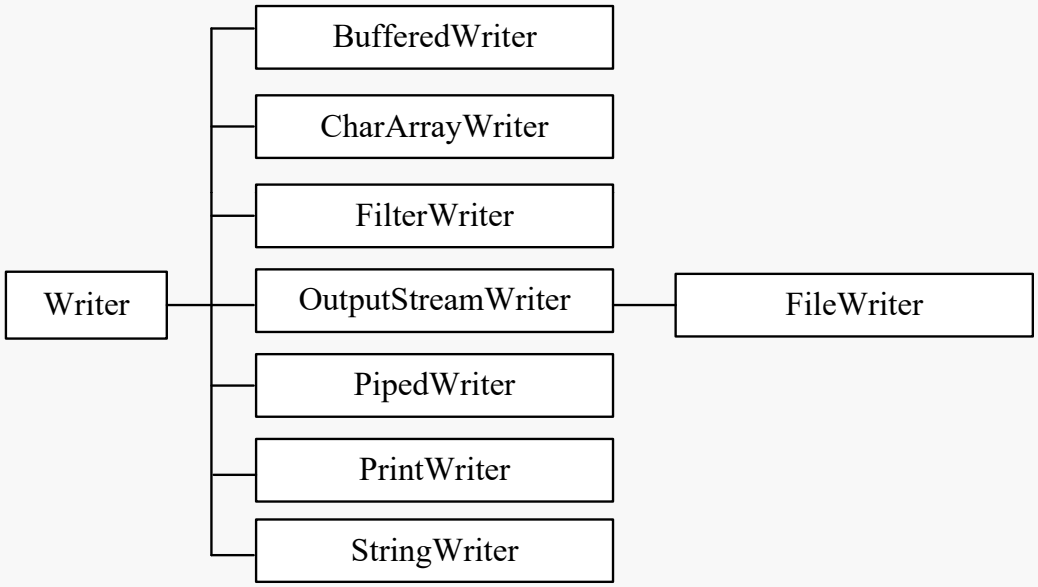
OutputStream类中的所有方法均返回void，在遇到错误时会引发IOException异常。下面对OutputStream类中的方法作一简单的介绍，如下表所示：

方法	说明
write(int b)	将指定的字节写入此输出流
write(byte[] b)	将 b.length 个字节从指定的 byte 数组写入此输出流
write(byte[] b,int off,int len)	将指定 byte 数组中从偏移里 off 开始的 len 个字节写入此输出流
flush()	彻底完成输出并清空缓存区
close()	关闭输出流



输出流

Writer类是字符输出流的抽象类，所有字符输出类的实现都是它的子类，Writer类的层次结构如下图所示：



File类

本讲大纲：

1、文件的创建与删除

2、获取文件信息

支持网站：www.mrbccd.com

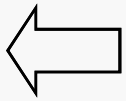
文件的创建与删除

可以使用File类创建一个文件对象，通常使用以下3种构造方法来创建文件对象。

File(String pathname)

File(String parent , String child)

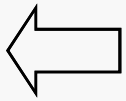
File(File f , String child)



获取文件信息

File类提供了很多方法用于获取文件本身的一些信息，File类的常用方法如下表所示。

方法	说明
getName()	获取文件的名称
canRead()	判断文件是否是可读的
canWrite()	判断文件是否可被写入
exits()	判断文件是否存在
length()	获取文件的长度（以字节为单位）
getAbsolutePath()	获取文件的绝对路径
getParent()	获取文件的父路径
isFile()	判断文件是否存在
isDirectory()	判断文件是否是一个目录
isHidden()	判断文件是否是隐藏文件
lastModified()	获取文件最后修改时间



文件输入输出流

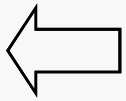
本讲大纲：

1、FileInputStream与FileOutputStream类
2、FileReader类和FileWriter类

支持网站：www.mrbccd.com

FileInputStream与FileOutputStream类

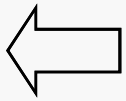
FileInputStream类与FileOutputStream类都是用来操作磁盘文件。如果用户的文件读取需求比较简单，则可以使用FileInputStream类。该类继承自InputStream类。FileOutputStream类与FileInputStream类对应，提供了基本的文件写入能力。FileOutputStream类是OutoputStream类的子类。



FileReader类和FileWriter类

使用FileOutputStream类向文件中写入数据与使用FileInputStream类从文件中将内容读出来，存在一点不足，即这两个类都只提供了对字节或字节数组的读取方法。由于汉字在文件中占用两个字节，如果使用字节流，读取不好可能会出现乱码现象。此时采用字符流Reader或Writer类即可避免这种现象。

FileReader、FileWriter字符流对应了FileInputStream、FileOutputStream类。FileReader流顺序地读取文件，只要不关闭流，每次调用read()方法就顺序地读取源中其余的内容，直到源的末尾或流被关闭。



带缓存的输入输出流

本讲大纲：

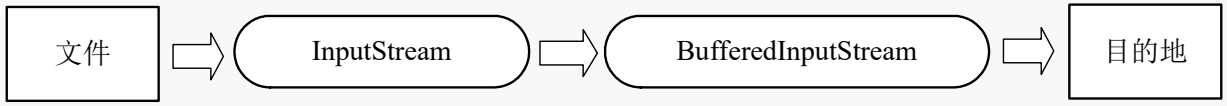
1、BufferedInputStream类与BufferedOutputStream类
2、BufferedReader与BufferedWriter类

支持网站：www.mrbccd.com

BufferedInputStream类与BufferedOutputStream类

BufferedInputStream类可以对任何的InputStream类进行带缓存区的包装以达到性能的优化。

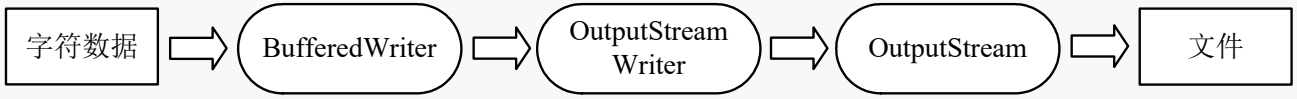
使用BufferedOutputStream输出信息和往OutputStream输出信息完全一样，只不过BufferedOutputStream有一个flush()方法用来将缓存区的数据强制输出完。



BufferedReader与BufferedWriter类

BufferedReader类与BufferedWriter类分别继承Reader类与Writer类。这两个类同样具有内部缓存机制，并可以以行为单位进行输入输出。

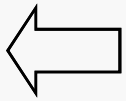
在使用BufferedWriter类的Write()方法时，数据并没有立刻被写入至输出流中，而是首先进入缓存区中。如果想立刻将缓存区中的数据写入输出流中，一定要调用flush()方法。



数据输入输出流

数据输入输出流（DataInputStream类与DataOutputStream类）允许应用程序以与机器无关的方式从底层输入流中读取基本Java数据类型。也就是说，当读取一个数据时，不必再关心这个数值应当是什么字节。

DataInputStream类只提供了一个readUTF()方法返回字符串。这是因为要在一个连续的字节流读取一个字符串，如果没有特殊的标记作为一个字符串的结尾，并且事先也不知道这个字符串的长度，也就无法知道读取到什么位置才是这个字符串的结束。DataOutputStream类中只有writeUTF()方法向目标设备中写入字符串的长度，所以我们也只能准确地读回写入字符串。



ZIP压缩输入输出流

本讲大纲：

1、压缩文件

2、解压缩ZIP文件

支持网站：www.mrbccd.com

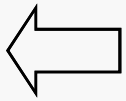
压缩文件

利用ZipOutputStream类对象，可将文件压缩为“.zip”文件。ZipOutputStream类的构造函数如下所示：

ZipOutputStream(OutputStream out);

ZipOutputStream类的常用方法如下表所示：

方法	说明
putNextEntry(ZipEntry e)	开始写一个新的 ZipEntry，并且将流内的位置移至此 entry 所指数据的开头
write(byte[] b, int off, int len)	将字节数组写入当前 ZIP 条目数据
finish()	完成写入 ZIP 输出流的内容，无须关闭它所配合的 OutputStream
setComment(String comment)	可设置此 ZIP 文件的注释文字



解压缩ZIP文件

ZipInputStream类可读取ZIP压缩格式的文件，包括对已压缩和未压缩条目的支持（entry）。ZipInputStream类的构造函数如下所示：

ZipInputStream(InputStream in)

ZipInputStream类的常用方法如下表所示：

方法	说明
read(byte[]b, int off, int len)	读取目标 b 数组内 off 偏移量的位置，长度是 len 字节
available()	判断是否已读完目前 entry 所指定的数据。已读完返回 0，否则返回 1
closeEntry()	关闭当前 ZIP 条目并定位流以读取下一个条目
skip(long n)	跳过当前 ZIP 条目中指定的字节数
getNextEntry()	读取下一个 ZipEntry，并将流内的位置移至该 entry 所指数据的开头
createZipEntry(String name)	以指定的 name 参数新建一个 ZipEntry 对象