

计算机二级java辅导:浅谈输入输出流计算机等级考试 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_E8_AE_A1_E7_AE_97_E6_9C_BA_E4_c97_645346.htm

Java语言的输入输出功能是十分强大而灵活的，美中不足的是看上去输入输出的代码并不是很简洁，因为你往往需要包装许多不同的对象。在Java类库中，IO部分的内容是很庞大的，因为它涉及的领域很广泛:标准输入输出，文件的操作，网络上的数据流，字符串流，对象流，zip文件流....本文的目的是为大家做一个简要的介绍。 流是一个很形象的概念，当程序需要读取数据的时候，就会开启一个通向数据源的流，这个数据源可以是文件，内存，或是网络连接。类似的，当程序需要写入数据的时候，就会开启一个通向目的地的流。(想:是不是一个东西如果需要从某个地方读取,然后在写到某个地方,需要开读入的流和输出的流)这时候你就可以想象数据好像在这其中“流”动一样，如下图： Java中的流分为两种，一种是字节流，另一种是字符流，分别由四个抽象类来表示（每种流包括输入和输出两种所以一共四个）:InputStream，OutputStream，Reader，Writer。Java中其他多种多样变化的流均是由它们派生出来的: 在这其中InputStream和OutputStream在早期的Java版本中就已经存在了，它们是基于字节流的，而基于字符流的Reader和Writer是后来加入作为补充的。以上的层次图是Java类库中的一个基本的层次体系。 在这四个抽象类中，InputStream和Reader定义了完全相同的接口：
int read ()
int read (char cbuf[])
int read (char cbuf[], int offset, int length)

而OutputStream和Writer也是如此：
int write (int c)
int write

(char cbuf[]) int write (char cbuf[], int offset, int length) 这六个方法都是最基本的方法，read() 和 write() 通过方法的重载来读写一个字节，或者一个字节数组。更多灵活多变的功能是由它们的子类来扩充完成的。知道了 Java 输入输出的基本层次结构以后，本文在这里想给大家一些以后可以反复应用的例子，对于所有子类的细节及其功能并不详细讨论。java 代码对于上面的例子，需要说明的有以下几点：1. BufferedReader 是 Reader 的一个子类，它具有缓冲的作用，避免了频繁的从物理设备中读取信息。它有以下两个构造函数：

BufferedReader (Reader in) BufferedReader (Reader in, int sz)
这里的 sz 是指定缓冲区的大小。它的基本方法：
void close () // 关闭流
void mark (int readAheadLimit) // 标记当前位置
boolean markSupported () // 是否支持标记
int read () // 继承自 Reader 的基本方法
100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com