JAVA程序设计

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

目 录 CONTENTS

1 学习指南

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

学习指南

本章首先介绍了输入输出流的原理，由这些原理出发使用实例实现了使用字节输入输出流和字符输入输出流进行的文件操作，最后介绍了使用java的标准类型实现文件的创建过程和访问过程。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

2 难点重点

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

难点重点

- 字节输入输出流
- 字符输入输出流
- 文件的创建与访问

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

3 知识内容

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

1. 输入输出流简介

应用程序都需要与外部设备进行数据交换，比如经常需要从键盘输入数据，在文件中读写数据以及在网络上舆数据。输入和输出（I/O）是程序设计语言的一项重要功能，是程序与用户之间沟通的桥梁。Java语言定义了许多类专门负责处理各种方式的输入输出，这些类都放在java.io包中。

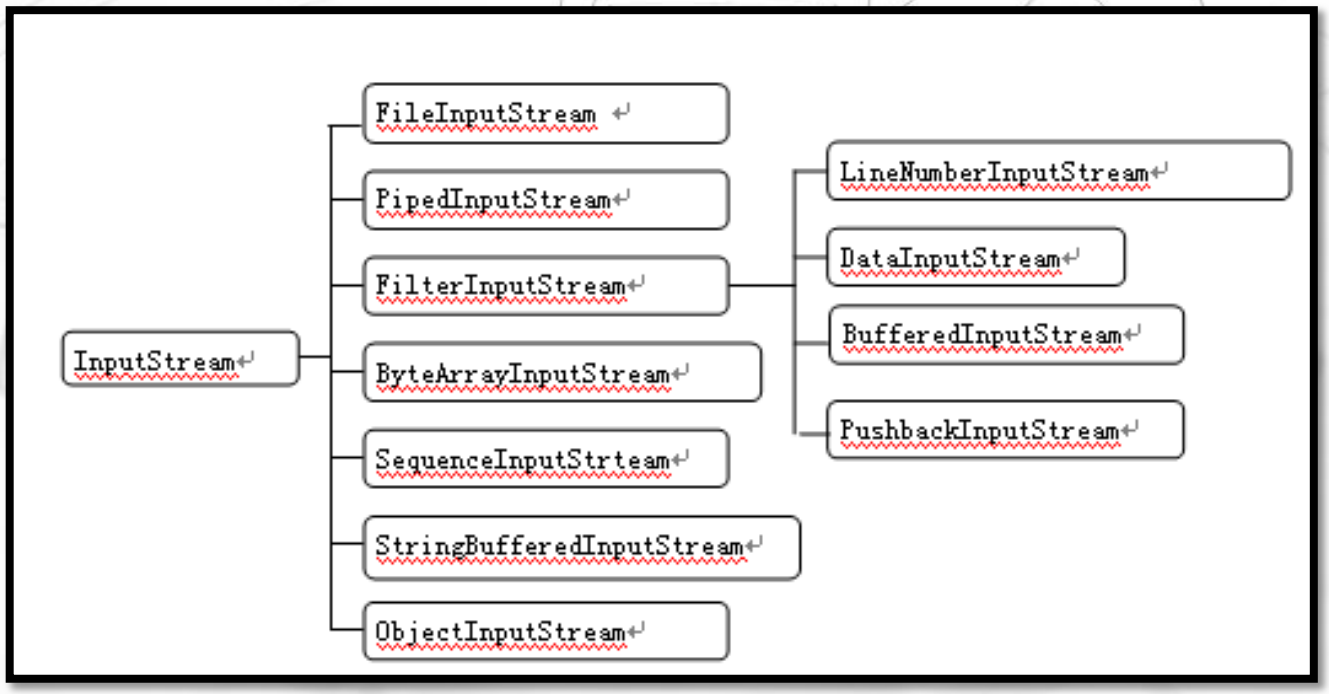
1. 输入输出流简介

所谓“数据流（stream）”，指的是所有数据通信通道之中数据的起点和终点。例如，执行程序通常会输出各种信息到显示器，又例如，一个程序在打开某一文件时，程序和文件之间就建立起一个数据流，文件的内容就是数据流中的数据，若这个文件是程序所要读取的文件，那么数据流的源就是文件，而目的地就是程序；若要对文件进行写入操作时，情况相反。总之，只要是数据从一个地方“流”到另外一个地方，这种数据流动的通道都可以称为数据流。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

1. 输入输出流简介

I/O流的层次（InputStream的继承层次结构）



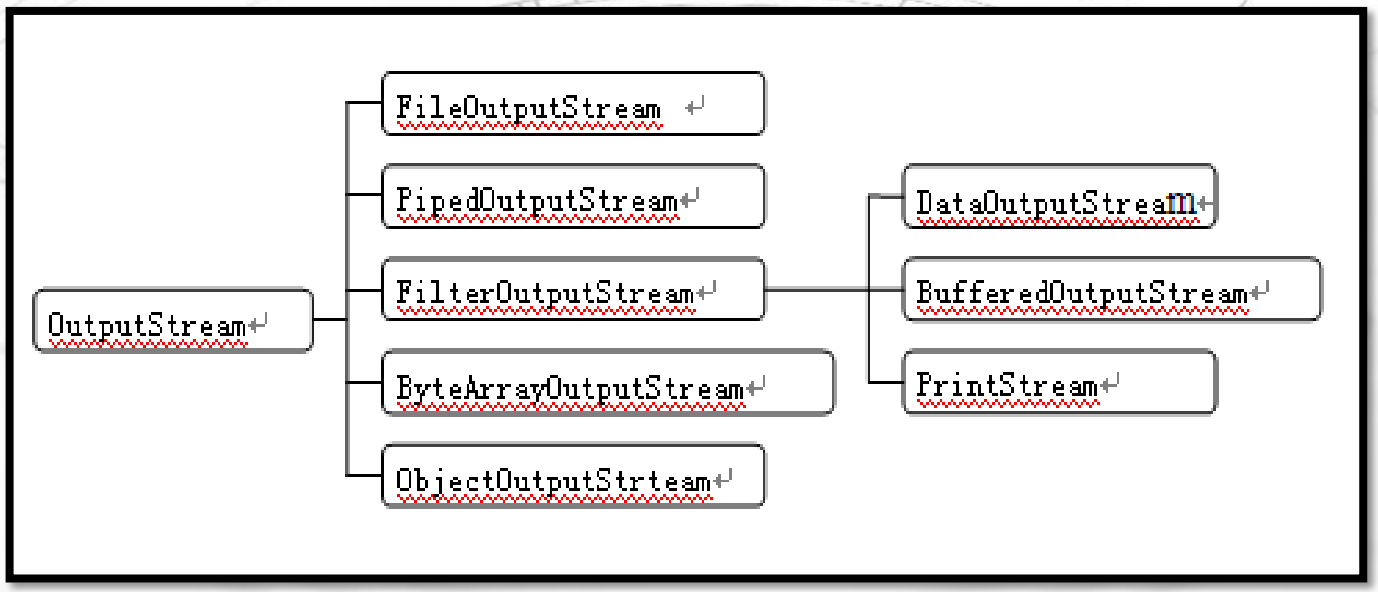
1. 输入输出流简介

I/O流的层次（InputStream的继承层次结构）

类	描述
FileInputStream	从文件系统中的某个文件中获取输入字节
PipedInputStream	传送输入流应该连接到传送输出流；传送输入流会提供要写入传送输出流的所有数据字节
FilterInputStream	包含其它一些输入流，它将这些流用作其基本数据源，它可以直接传输数据或提供一些额外的功能
ByteArrayInputStream	包含一个内部缓冲区，该缓冲区存储从流中读取的字节
SepuenceInputStream	表示其它输入流的逻辑串联
StringBufferInputStream	已过时。 此类未能正确地将字符转换为字节。
ObjectInputStream	对以前使用 ObjectOutputStream 写入的基本数据和对象进行反序列化。
LineNumberInputStream	已过时。 此类错误假定字节能充分表示字符。
DataInputStream	数据输入流允许应用程序以与机器无关方式从基础输入流中读取基本 Java 数据类型。应用程序可以使用数据输出流写入稍后由数据输入流读取的数据。
BufferedInputStream	作为另一种输入流，BufferedInputStream 为InputStream添加了功能，即缓冲输入和支持 mark 和 reset 方法的能力。
PushbackInputStream	向另一个输入流添加“推回 (push back)”或“取消读取 (unread)”一个字节的功能。

1. 输入输出流简介

I/O流的层次（InputStream的继承层次结构）



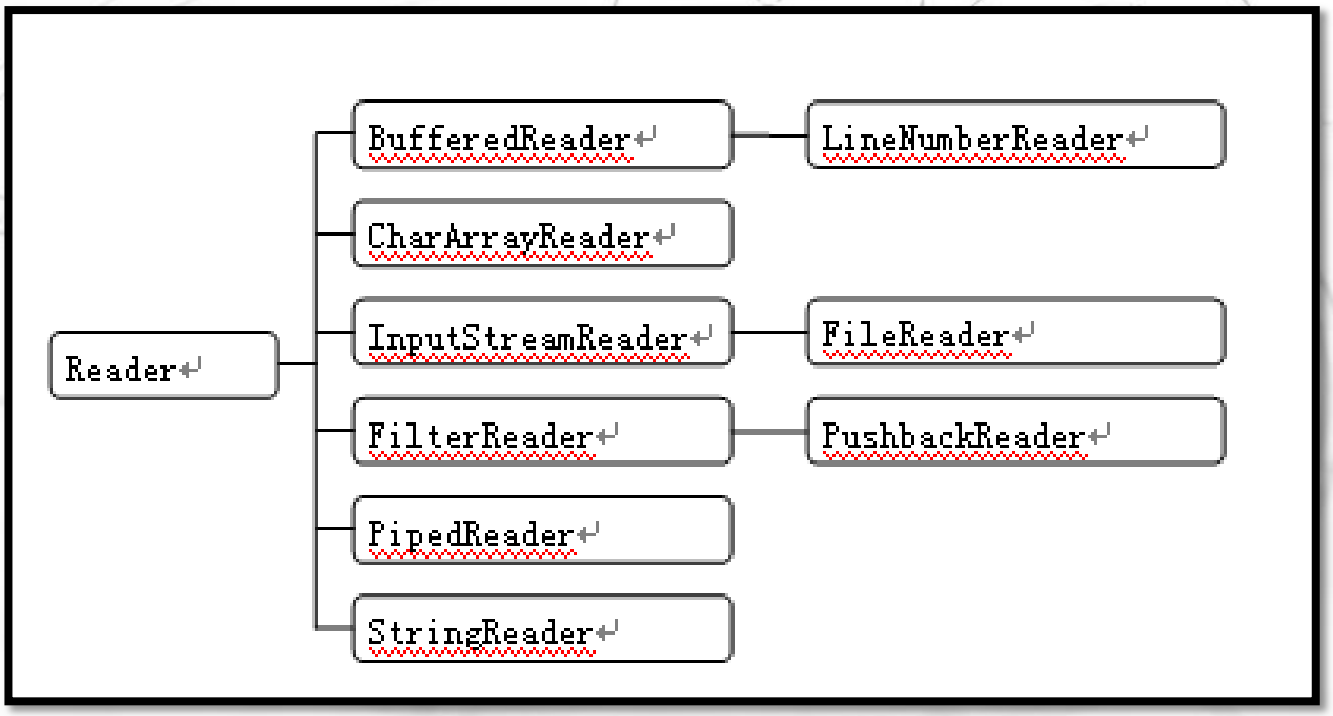
1. 输入输出流简介

I/O流的层次（InputStream的继承层次结构）

类	描述
FileOutputStream	文件输出流是用于将数据写入 File 或 FileDescriptor的输出流
PipedOutputStream	传送输出流可以连接到传送输入流，以创建通信管道。
FilterOutputStream	此类是过滤输出流的所有类的超类。
ByteArrayOutputStream	此类实现了一个输出流，其中的数据被写入一个字节数组。
ObjectOutputStream	将 Java 对象的基本数据类型和图形写入 OutputStream。
DataOutputStream	数据输出流允许应用程序以适当方式将基本 Java 数据类型写入输出流中。
BufferedOutputStream	该类实现缓冲的输出流
PrintStream	为其他输出流添加了功能，使它们能够方便地打印各种数据值表示形式。它还提供其他两项功能。

1. 输入输出流简介

I/O流的层次（InputStream的继承层次结构）



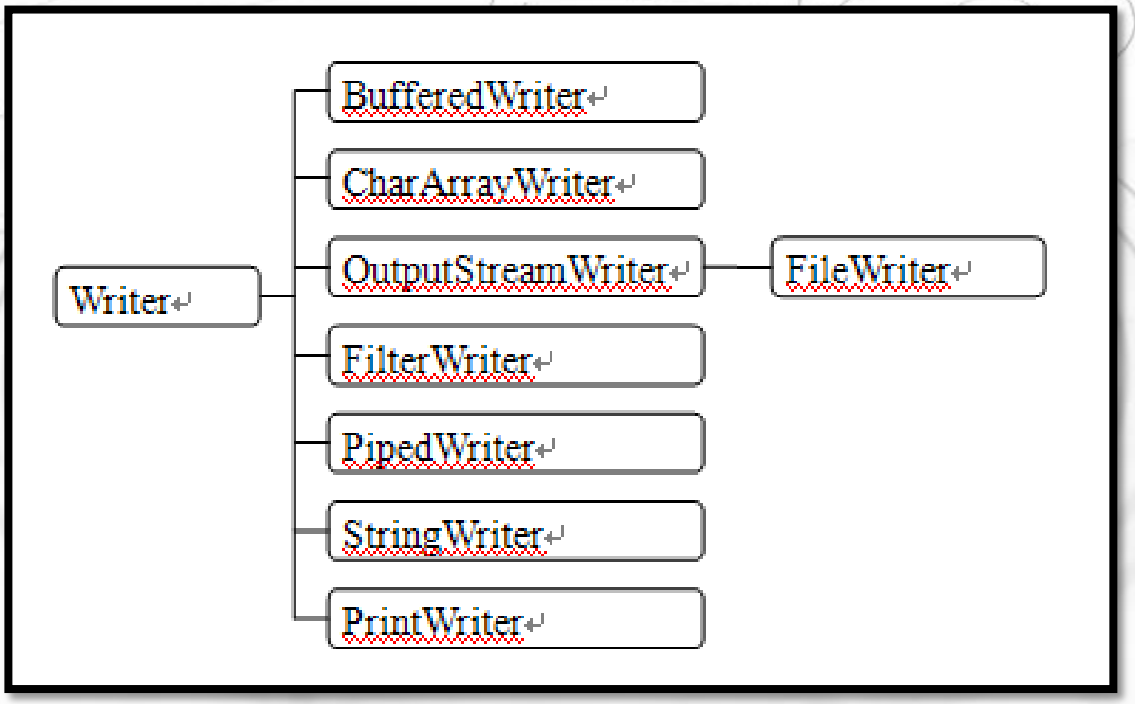
1. 输入输出流简介

I/O流的层次（Reader类）

类	描述
BufferedReader	从字符输入流中读取文本，缓冲各个字符，从而提供字符、数组和行的高效读取。
CharArrayReader	此类实现一个可用作字符输入流的字符缓冲区。
InputStreamReader	是字节流通向字符流的桥梁：它使用指定的 charset 读取字节并将其解码为字符。
FilterReader	用于读取已过滤的字符流的抽象类。
PipedReader	传送的字符输入流。
StringReader	其源为一个字符串的字符流。

1. 输入输出流简介

I/O流的层次（writer类）



1. 输入输出流简介

I/O流的层次 （ writer类列表 ）

类	描述
BufferedWriter	将文本写入字符输出流，缓冲各个字符，从而提供单个字符、数组和字符串的高效写入。
CharArrayWriter	此类实现一个可用作 Writer 的字符缓冲区。
OutputStreamWriter	是字符流通向字节流的桥梁：使用指定的 charset 将要向其写入的字符编码为字节。
FilterWriter	用于写入已过滤的字符流的抽象类。
PipedWriter	传送的字符输出流。
StringWriter	一个字符流，可以用其回收在字符串缓冲区中的输出来构造字符串。
PrintWriter	向文本输出流打印对象的格式化表示形式。

2. 字节输入输出流

一般来说处理字符或字符串时应使用字符流类，处理字节或二进制时应使用字节流类。InputStream类和OutputStream类为字节流设计。Reader类和Writer类则为字符流设计。但在最底层，所有的输入/输出都是字节形式的，基于字符的流只是为了处理字符更方便一些。

在字节流中，输入流用InputStream类完成，输出流用OutputStream类完成。

2. 字节输入输出流

2.1 字节输入流 InputStream类

InputStream类中的方法有：

（1）从流中读取数据

int read():读取一个字节，返回值为所读的字节的整形表示。如果为-1，则表明文件结束。

2. 字节输入输出流

2.1 字节输入流 InputStream类

（1）从流中读取数据

int read(byte b[])：读取多个字节，放置到字节数组b中，通常读取的字节数量为b的长度，返回值为实际读取的字节的数量。

int read(byte b[],int off,int len)：读取len个字节，放置到以下标off开始的字节数组b中，返回值为实际读取的字节的数量。

int available():返回值为流中尚未读取的字节的数量。

long skip(long n):读指针跳过n个字节不读，返回值为实际跳过的字节数量。

2. 字节输入输出流

2.1 字节输入流 InputStream类

（2）关闭流

close():流操作完毕后必须关闭。

（3）使用输入流中的标记

void mark(int readlimit):记录当前读指针所在位置，readlimit表示读指针读出readlimit个字节后所标记的指针位置才失效。

void reset():把读指针重新指向用mark方法所记录的位置。

boolean markSupported():当前的流是否支持读指针的记录功能。

2. 字节输入输出流

2.2 字节输出流OutputStream类

OutputStream是一个定义了输出流的抽象类，这个类中的所有方法的返回值均为void，并在遇到错误时引发IOException异常。OutputStream类中的方法有以下几种：

（1）输出数据

void write(int b):向流中写一个字节b，这里的参数是int型，它允许不转成byte型。

void write(byte b[]):向流中写一个字节数组b。

void write(byte b[]，int off,int len):把字节数组b中从下标off开始，长度为len的字节写入流中。

2. 字节输入输出流

2.2 字节输出流OutputStream类

2．刷新

flush():刷空输出流,并输出所有被缓存的字节。由于某些流支持缓存功能，该方法将把缓存中所有内容强制输出到流中。

3．关闭流

close():关闭流操作。

3. 字符输入输出流

java中的字符是Unicode编码，是双字节的，而InputStream类和OutputStream类是处理字节的，在处理字符文本是不太方便，为此java设计了字符流类。Reader和Writer两个抽象类与InputStream类和OutStream类相对应。这两个类是抽象类，只是提供了一系列用于字符流处理的接口，不能生成这两个类的实例，只能通过使用由它们派生出来的子类所生成的对象来处理字符流，它们下面的子类处理具体IO设备的字符输入输出，如FileReader就是用来读取文件流中的字符。

3. 字符输入输出流

3.1 字符输入流Reader类

Reader类是处理所有字符流输入类的父类。该类的所有方法在出错情况下都将引发IOException异常。

Reader类大体的功能和InputStream类相同，Reader类中的方法如下：

3. 字符输入输出流

3.1 字符输入流Reader类

（1）读取字符

public int read() throws IOException

读取一个字符，返回值为读取的字符。

public int read(char cbuf[]) throws IOException

读取一系列字符到cbuf[]中，返回值为实际读取的字符的数量。

public abstract int read(char cbuf[],int off,int len) throws IOException读取len个字符，从数组cbuf[]的下标off处开始存放，返回值为实际读取的字符数量，该方法必须由子类实现。

3. 字符输入输出流

3.1 字符输入流Reader类

（2）标记流

public boolean markSupported()

判断当前流是否支持做标记。

public void mark(int readAheadLimit) throws IOException

给当前流作标记，最多支持readAheadLimit个字符的回溯。

public void reset() throws IOException

将当前流重置到做标记处。

（3）关闭流

public abstract void close() throws IOException

关闭输入流。

3. 字符输入输出流

3.2字符输出流Writer类

Writer类是处理所有字符输出类的父类。该类中所有方法的返回值均为void，并在出错情况下引发IOException异常。

Writer类大体的功能和OutStream类相同，Writer类中的方法如下：

3. 字符输入输出流

3.2字符输出流Writer类

（1）向输出流写入字符

public void writer(int c)throws IOException

将整型值c的低16位写入输出流。

public void writer (char cbuf[]) throws IOException

将字符数组cbuf[]写入输出流。

public abstract void writer (char cbuf[],int off,int,len) throws IOException

将字符数组cbuf[]中的从索引为off的位置处开始的len个字符写入输出流。

3. 字符输入输出流

3.2字符输出流Writer类

（1）向输出流写入字符

public void writer (String str) throws IOException

将字符串str中的字符写入输出流。

public void writer (String str,int off,int,len) throws IOException

将字符串str中从索引为off的位置处开始的len个字符写入输出流。

3. 字符输入输出流

3.2字符输出流Writer类

（2）刷新

public abstract void flush()

刷空输出流，并输出所有被缓存的字节。

（3）关闭流

public abstract void close() throws IOException

关闭输出流。

4. 文件的创建与访问

4.1 File类

（1）构造函数

public File(String path): /*如果path是实际存在的路径，则该File对象表示的是目录；如果path是文件名，则该File对象表示的是文件。*/

public File(String path,String name): //path是路径名，name是文件名

public File(File dir,String name): // dir是路径名，name是文件名

4. 文件的创建与访问

4.1 File类

（1）构造函数

例：

(1)File file1=new File("d:/myfile/file.txt");

(2)File file2=new File("d:/myfile","file.txt");

(3)File myDir=new File("d:/myfile");File file3=new File(myDir,"file.txt");

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

4. 文件的创建与访问

4.1 File类

（2）文件名相关的处理方法

String getName(): 得到一个文件的名称(不包括路径)。

String getPath(): 得到一个文件的路径名。

String getAbsolutePath(): 得到一个文件的绝对路径名。

String getParent(): 得到一个文件的上一级目录名

String renameTo(File newName): 将当前文件名更名为给定文件名。

4. 文件的创建与访问

4.1 File类

（3）文件属性测试

boolean exists(): 测试当前File对象所指示的文件是否存在。

boolean canWrite(): 测试当前文件是否可写。

boolean canRead(): 测试当前文件是否可读。

boolean isFile(): 测试当前文件是否是文件(不是目录)。

boolean isDirectory(): 测试当前文件是否是目录。

4. 文件的创建与访问

4.1 File类

（4）普通文件信息和工具

long lastModified(): 得到文件最近一次修改时间。

long length(): 得到文件的长度，以字节为单位。

boolean delete(): 删除当前文件。

4. 文件的创建与访问

4.1 File类

（5）目录操作

boolean mkdir(): 根据当前对象生成一个由该对象指定的路径。

String list(): 列出当前目录下的文件

4. 文件的创建与访问

4.2 输入输出文件流

（1）创建字节输入文件流FileInputStream类对象

若需要以字节为单位顺序读出一个已存在文件的数据，可使用字节输入流FileInputStream。可以用文件名、文件对象或文件描述符建立字节文件流对象。FileInputStream类构造方法有：

①FileInputStream(String name)用文件名name建立流对象。

例如：

```
FileInputStream fis=new FileInputStream("c:/config.sys");
```

4. 文件的创建与访问

4.2 输入输出文件流

（1）创建字节输入文件流FileInputStream类对象

②FileInputStream(File file)用文件对象file建立流对象。

例如：

```
File myfile=new File("c:/config.sys");
FileInputStream fis=new FileInputStream(myFile);
```

4. 文件的创建与访问

4.2 输入输出文件流

（2）读取文件信息

若使用FileInputStream类上述的构造方法创建输入流对像，就在程序和对应文件之间建立了一个通道，并打开了相应文件，现在可以使用该对象的方法从文件里读取数据了。

4. 文件的创建与访问

4.2 输入输出文件流

（2）读取文件信息

读取字节信息，一般使用read()成员方法和它的重载方法：

int read()　读流中一个字节，若流结束则返回-1.

int read(byte b[])　从流中读字节填满字节数组b，返回所读字节数，若流结束则返回-1.

int read(byte b[],int off,int len)　从流中读字节填入b[off]开始处，返回所读字节数，若流结束则返回-1.

4. 文件的创建与访问

4.2 输入输出文件流

（3）创建字节输出文件流FileOutputStream类对象

FileOutputStream可表示一种创建并顺序写的文件。在构造此类对象时，若指定路径的文件不存在，会自动创建一个新文件；若指定路径已有一个同名文件，该文件的内容将被保留或删除。

4. 文件的创建与访问

4.2 输入输出文件流

（4）向输出流写信息

向FileOutputStream中写入信息，一般用write()方法，该方法有重载：

void write(int b)将整型数据的低字节写入输出流。

void write(byte b[])将字节数组b中的数据写入输出流。

void write(byte b[],int off,int len)将字节数组b中从off开始的len个字节数据写入输出流。

4. 文件的创建与访问

4.2 输入输出文件流

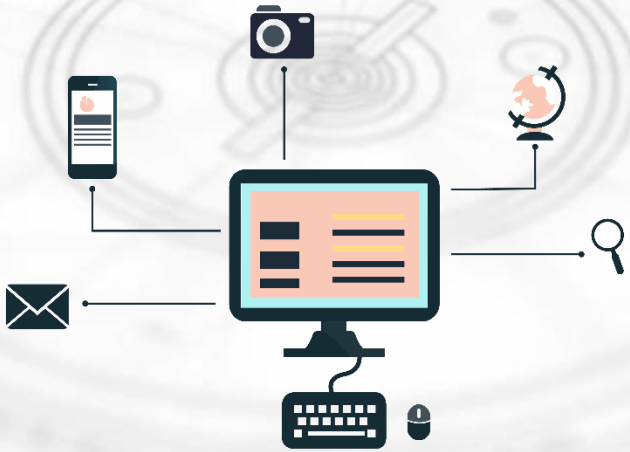
（5）关闭FileInputStream

从文件读取完数据后，可以使用两种方法关闭流，一种是显式的关闭流对象，使用close()方法；另一种是隐式关闭输入流，Java有自动垃圾收集系统，可以自动进行资源回收。在编程过程中，笔者推荐采用第一种方式关闭流。

4. 文件的创建与访问

4.2 输入输出文件流

例7.1是使用FileInputStream与FileOutputStream的一个例子。程序可以复制文件，它会先从来源文件读取数据至一个byte数组中，然后再将byte数组的数据写入目的文件。



4. 文件的创建与访问

4.3 随机读写文件流的输入输出

对InputStream和OutputStream来说，它们的实例都是顺序访问的流，也就是说，只能对文件进行顺序地读写。随机访问文件则允许对文件内容进行随机读写。在Java中，类RandomAccessFile提供了随机访问文件的方法。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

4. 文件的创建与访问

4.3 随机读写文件流的输入输出

（1）构造方法

RandomAccessFile(String name ,String mode); //name是文件名，mode//是打开方式，例如”r”表示只读，” rw”表示可读写

RandomAccessFile(File file,String name); //file是文件对像

4. 文件的创建与访问

4.3 随机读写文件流的输入输出

（2）文件指针的操作

```
long getFilePointer();    //用于得到当前的文件指针
void seek(long pos);    //用于移动文件指针到指定的位置
int skipBytes(int n);    //使文件指针向前移动n个字节
```

4. 文件的创建与访问

4.4 标准输入输出流

为方便使用计算机常用的输入输出设备，各种高级语言与操作系统对应，都规定了可用的标准设备（文件）。所谓标准设备（文件），也称为预定义设备（文件），在程序中使用这些设备（文件）时，可以不用专门的打开操作就能简单地应用。一般地，标准输入设备是键盘，标准输出设备是终端显示器，标准错误输出设备也是显示器。

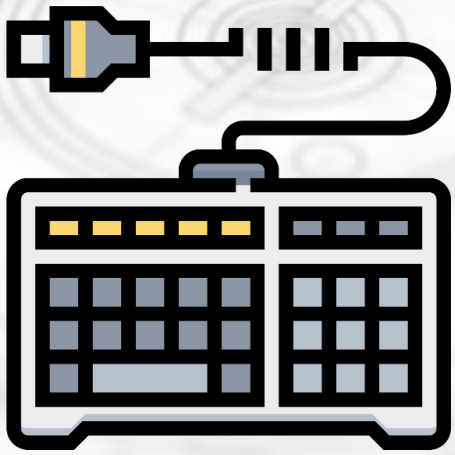
中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

4. 文件的创建与访问

4.4 标准输入输出流

Java语言的系统类System提供访问标准输入输出设备的功能。

System类是继承Object类的终极类，它有3个类变量：in、out和err,分别表示标准输入，标准输出和标准错误输出流。



4 本章小结

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

本章小结

本章主要介绍了输入输出流的基本概念、字节输入流和输出流的实现、字符输入流和输出流、文件的创建与访问等内容。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社


THANK YOU
中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社