



输入、输出流



邹国兵

上海大学

计算机学院




第10章 输入输出流

1. File类

2. 文件输入输出流

3. 文件字节输入输出流

4. 文件字符输入输出流



10.1 File类

- 创建一个File对象的常用构造方法有2个:
- `File(String filename);`
- `File(String directoryPath, String filename);`
- 例如:
- `File f=new File("C:\\\\ch10" , "1.txt");`
- `File f=new File("1.txt");`

10.1.1 文件的属性

- ▶ `public String getName()` 获取文件的名字。
- ▶ `public boolean canRead()` 判断文件是否是可读的。
- ▶ `public boolean canWrite()` 判断文件是否可被写入。
- ▶ `public boolean exists()` 判断文件是否存在。
- ▶ `public long length()` 获取文件的长度（单位是字节）。
- ▶ `public String getAbsolutePath()` 获取文件的绝对路径。
- ▶ `public String getParent()` 获取文件的父目录。
- ▶ `public boolean isFile()` 判断文件是否是一个普通文件。
- ▶ `public boolean isDirectory()` 判断文件是否是一个目录。
- ▶ `public boolean isHidden()` 判断文件是否是隐藏文件。

10.1.2 文件的创建与删除

- `boolean createNewFile();` \\创建文件
- `boolean delete();` \\删除当前文件
- 当使用File类创建一个文件对象后，例如：
`File file=new File("c:\\myletter","letter.txt");`
`file.createNewFile();`
`file.delete();`

文件属性与操作综合举例

```
1 package shu.ces.java.chap10;
2
3 import java.io.*;
4 public class Example10_1 {
5     public static void main(String args[]) {
6         File f = new File("C:\\ch10", "Example10_1.java");
7         System.out.println(f.getName()+"是可读的吗:"+f.canRead());
8         System.out.println(f.getName()+"的长度:"+f.length());
9         System.out.println(f.getName()+"的绝对路径:"+f.getAbsolutePath());
10
11         File file = new File("new.txt");
12         System.out.println("\n在当前目录下创建新文件"+file.getName());
13         if(!file.exists()) {
14             try {
15                 file.createNewFile();
16                 System.out.println("创建成功！");
17             }
18             catch(IOException exp){}
19         } else {
20             System.out.println("该文件已存在，不需要创建！");
21         }
22     }
23 }
24
```

例10-1演示

10.1.3 运行可执行文件

- `Runtime ce`; 用 `Runtime` 类声明一个对象 (`java.lang` 包)
- `ce=Runtime.getRuntime()`; 创建这个对象:
- `ce.exec(String command)`; 打开可执行文件或操作。

```
1 package org.shu.ces.java.chap10;
2
3 import java.io.*;
4 public class Example10_3 {
5     public static void main(String args[]) {
6         try{
7             Runtime ce=Runtime.getRuntime();
8             File file=new File("c:/windows","Notepad.exe");
9             ce.exec(file.getAbsolutePath());
10            file=new File("C:\\Program Files\\Internet Explorer","IEXPLORE www.shu.edu.cn");
11            ce.exec(file.getAbsolutePath());
12        }
13        catch(Exception e) {
14            System.out.println(e);
15        }
16    }
17 }
```

例10-3演示




第10章 输入输出流

1. File类

2. 文件输入输出流

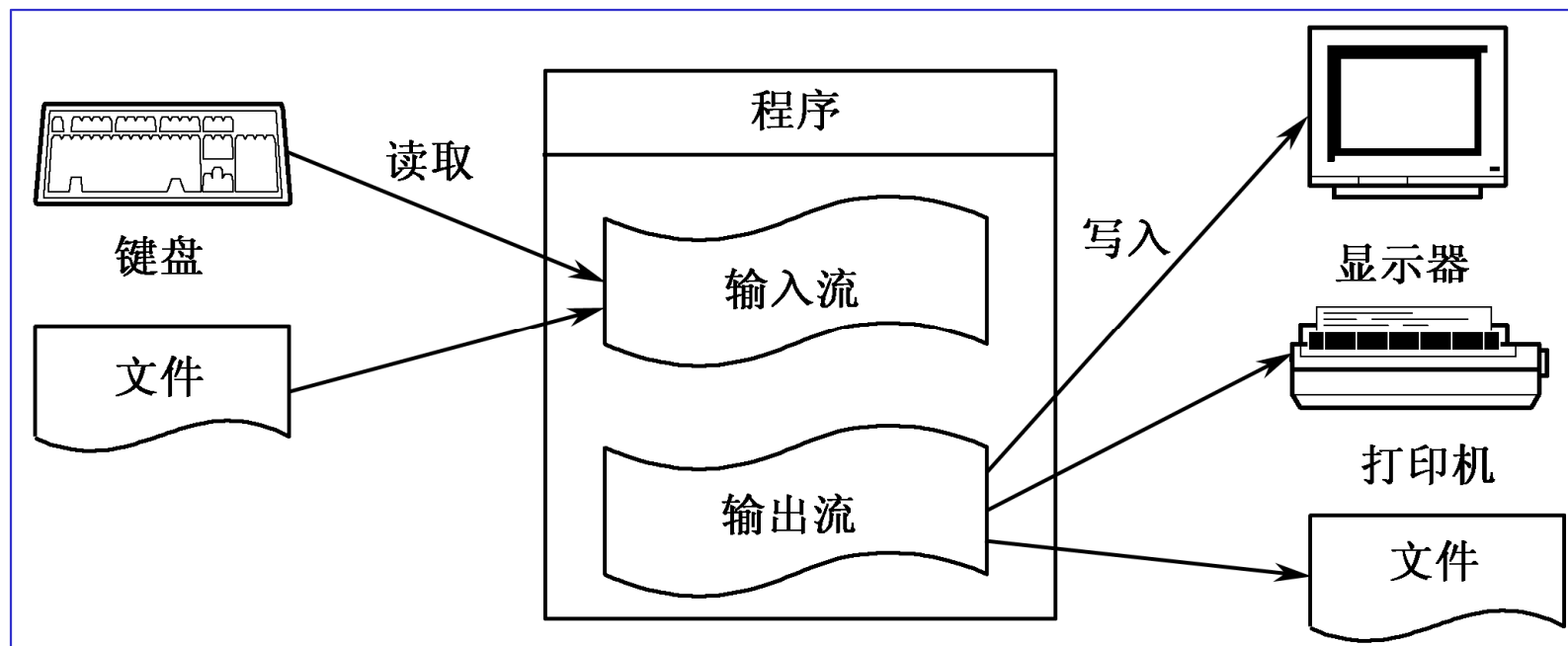
3. 文件字节输入输出流

4. 文件字符输入输出流



10.2 输入流与输出流

- 针对不同的源和目的地，java.io包提供了相应的输入流和输出流。



输入流与输出流

10.2 输入流与输出流

按数据流的方向不同：输入流和输出流

按处理数据单位不同：字节流和字符流

Java所提供的所有流类型位于**java.io**包中，
继承以下四种抽象类型

数据流类	说明	补充说明
InputStream	字节输入流的超类	它是抽象类,子类才能实例化
OutputStream	字节输出流的超类	它是抽象类,子类才能实例化
Reader	字符输入流的超类	它是抽象类,子类才能实例化
Writer	字符输出流的超类	它是抽象类,子类才能实例化


10.2 文件输入流与文件输出流

文件输入流 文件输出流

数据流类	说明	补充说明
FileInputStream	文件字节输入流类	从数据源按字节读取数据
FileOutputStream	文件字节输出流类	按字节写入目的地
FileReader	文件字符输入流类	从数据源按字符读取数据
FileWriter	文件字符输出流类	按字符写入目的地



第10章 输入输出流

1. File类
 2. 文件输入输出流
 3. 文件字节输入输出流
 4. 文件字符输入输出流
- 

10.3 文件字节流（输入和输出）

FileInputStream-文件字节输入流

构造方法:

```
public FileInputStream(String name)
```

```
public FileInputStream(File file)
```

用FileInputStream类创建指向该文件的文件字节输入流。

调用从父类继承read方法顺序地以字节为单位读取文件。

10.3 文件字节流（输入流）

使用输入流通常包括4个基本步骤：

- (1) 设定输入流的源
- (2) 创建指向源的输入流
- (3) 从输入流读取源中的数据
- (4) 关闭输入流

使用输入流read方法读取文件中数据时，通常采用：

```
public int read(byte b[], int off, int len)
```

10.3 文件字节流（输入流）

```
1 package org.shu.ces.java.chap10;
2
3 import java.io.*;
4 public class Example10_4 {
5     public static void main(String args[]) {
6         int n=-1;
7         byte [] a=new byte[100];
8         try{ File f=new File("Example10_4.java");
9             InputStream in = new FileInputStream(f);
10            while ((n=in.read(a,0,100))!=-1) {
11                String s=new String (a,0,n);
12                System.out.print(s);
13            }
14            in.close();
15        }
16        catch(IOException e) {
17            System.out.println("File read Error "+e);
18        }
19    }
20 }
```

例10-4演示

10.3 文件字节流（输出流）

FileOutputStream-文件字节输出流

构造方法:

```
public FileOutputStream(String name);  
public FileOutputStream(File file)  
public FileOutputStream(String name, boolean append)  
public FileOutputStream(File file, boolean append)
```

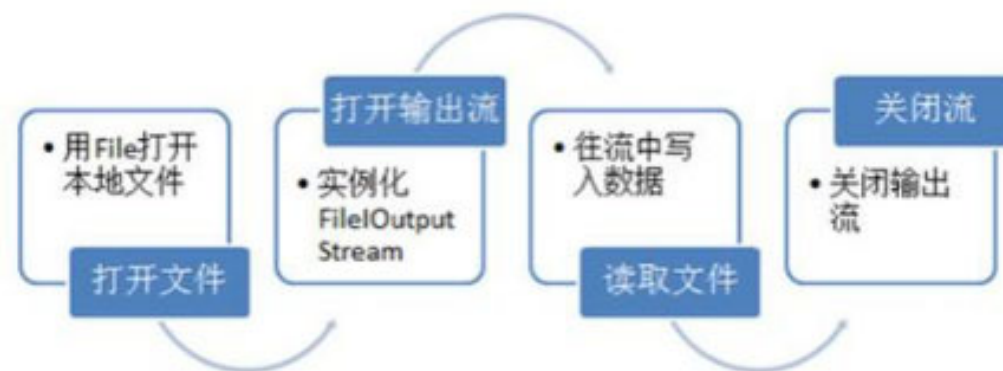
用FileOutputStream类创建指向该文件的文件字节输出流。

调用从父类继承write方法顺序地以字节为单位写入文件。

10.3 文件字节流（输出流）

使用输出流通常包括4个基本步骤：

- (1) 给出输出流目的地
- (2) 创建指向目的地的输出流
- (3) 输出流把数据写入目的地
- (4) 关闭输出流



使用输出流write方法写入文件中数据时，通常采用：

```
public void write(byte b[], int off, int len)
```

10.3 文件字节流（输出流）

```
1 package shu.ces.java.chap10;
2
3 import java.io.*;
4 public class Example10_5 {
5     public static void main(String args[]) {
6         byte [] a = "新年快乐".getBytes();
7         byte [] b = "Happy New Year".getBytes();
8         File file = new File("a.txt");           //输出的目的地
9         try{
10             OutputStream out=new FileOutputStream(file, true); //指向目的地的输出流
11             System.out.println(file.getName()+"的大小:"+file.length()+"字节");
12             out.write(a);                          //向目的地写数据
13             out.close();
14             out=new FileOutputStream(file, true);   //准备向文件尾加内容
15             System.out.println(file.getName()+"的大小:"+file.length()+"字节");
16             out.write(b,0,b.length);
17             System.out.println(file.getName()+"的大小:"+file.length()+"字节");
18             out.close();
19         }
20         catch(IOException e) {
21             System.out.println("Error "+e);
22         }
23     }
24 }
```

例10-5演示



第10章 输入输出流

1. File类
2. 文件输入输出流
3. 文件字节输入输出流
4. 文件字符输入输出流

10.4 文件字符流（输入和输出）

FileReader-文件字符输入流

构造方法:

```
public FileReader(String name)
```

```
public FileReader(File file)
```

用FileReader类创建指向该文件的字符输入流。

调用从父类继承`read`方法顺序地以字符为单位读取文件。

FileWriter-文件字符输出流

构造方法:

```
public FileWriter(String name);
```

```
public FileWriter(File file)
```

```
public FileWriter(String name, boolean append)
```

```
public FileWriter(File file, boolean append)
```

调用从父类继承`write`方法顺序地以字符为单位写入文件。

10.4 文件字符流（输入和输出）

```
1 package org.shu.ces.java.chap10;
2
3 import java.io.*;
4 public class Example10_6 {
5     public static void main(String args[]) {
6         File sourceFile = new File("a.txt"); //读取的文件
7         File targetFile = new File("b.txt"); //写入的文件
8         char c[] =new char[19];             //char型数组
9         try{
10             Writer out = new FileWriter(targetFile,true); //指向目的地的输出流
11             Reader in = new FileReader(sourceFile);        //指向源的输入流
12             int n = -1;
13             while((n=in.read(c))!=-1) {
14                 out.write(c,0,n);
15             }
16             out.flush();
17             out.close();
18         }
19         catch(IOException e) {
20             System.out.println("Error "+e);
21         }
22     }
23 }
24
```

例10-6演示