

概述

软件包

类

使用

树

已过时的

索引

帮助

上一个

下一个

框架

无框架

所有类

概要:

嵌套字段

CONSTR | 方法

详细信息:

字段

CONSTR | 方法

compact1, compact2, compact3

java.io

Class **FileInputStream**

java.lang.Object

java.io.InputStream

java.io.FileInputStream

All Implemented Interfaces:

Closeable, AutoCloseable

public class **FileInputStream**

extends `InputStream`

一个 `FileInputStream` 获得输入的字节从文件系统中的文件。什么文件是可用的，取决于主机环境。

`FileInputStream` 是读取原始字节的图像数据流。读取字符流，考虑使用 `FileReader`。

从以下版本开始:

JDK1.0

另请参见:

`File`, `FileDescriptor`, `FileOutputStream`, `Files.newInputStream(java.nio.file.Path, java.nio.file.OpenOption...)`

构造方法摘要

构造方法

Constructor and Description

- FileInputStream**(`File` file)
- 打开一个连接到一个实际的文件创建一个 `FileInputStream`，文件在文件系统的 `File` 对象 `file` 命名。
- FileInputStream**(`FileDescriptor` fdObj)
- 创建一个 `FileInputStream` 通过文件描述符 `fdObj`，它代表一个现有连接到文件系统中的实际的文件。
- FileInputStream**(`String` name)
- 打开一个连接到一个实际的文件创建一个 `FileInputStream`，文件由文件系统中的路径名 `name` 命名。

方法摘要

所有方法      接口方法      具体的方法

Modifier and Type

int

void

Method and Description

- available**()
- 返回一个剩余的字节数的估计，可以从这个输入流读取（或跳过），而不阻塞该输入流的方法的下一调用。
- close**()
- 关闭此文件输入流并释放与流关联的任何系统资源。

|                       |  |
|-----------------------|--|
| protected void        | <b>finalize()</b><br>确保该文件输入流的 <code>close</code> 方法被调用时，没有对它的引用。                                      |
| <b>FileChannel</b>    | <b>getChannel()</b><br>返回与此文件输入流有关的独特的 <b>FileChannel</b> 对象。  |
| <b>FileDescriptor</b> | <b>getFD()</b><br>返回表示实际的文件在文件系统中的 <code>FileInputStream</code> 使用的连接的 <code>FileDescriptor</code> 对象。 |
| int                   | <b>read()</b><br>从这个输入流读取一个字节的数据。  |
| int                   | <b>read(byte[] b)</b><br>读到 <code>b.length</code> 从输入流到字节数组数据字节。                                       |
| int                   | <b>read(byte[] b, int off, int len)</b><br>读到 <code>len</code> 从输入流到字节数组数据字节。                          |
| long                  | <b>skip(long n)</b><br>跳过并丢弃 <code>n</code> 字节输入流中的数据。   |

#### Methods inherited from class java.io.InputStream

`mark`, `markSupported`, `reset`

#### Methods inherited from class java.lang.Object

`clone`, `equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Constructor Detail

#### FileInputStream

```
public FileInputStream(String name)
    throws FileNotFoundException
```

打开一个连接到一个实际的文件创建一个 `FileInputStream`，文件由文件系统中的路径名 `name` 命名。一个新的 `FileDescriptor` 对象来表示这个文件连接。

首先，如果存在安全管理器，它的 `checkRead` 方法调用作为参数的 `name` 论点。

如果指定的文件不存在，是一个目录而不是常规文件，或者其他原因不能打开，然后 `FileNotFoundException` 抛出。

#### 参数

`name` - 系统相关的文件名称。

#### 异常

`FileNotFoundException` - 如果文件不存在，是一个目录而不是常规文件，或者其他原因不能打开阅读。

`SecurityException` - 如果存在一个安全管理及其 `checkRead` 法否认对文件的读访问。

#### 另请参见：

`SecurityManager.checkRead(java.lang.String)`

#### FileInputStream

```
public FileInputStream(File file)
    throws FileNotFoundException
```

打开一个连接到一个实际的文件创建一个 `FileInputStream`，文件在文件系统的 `File` 对象 `file` 命名。一个新的



throws `IOException`

读到 `b.length`从输入流到字节数组数据字节。该方法块，直到有一个输入可用。

**重写:**

`read` 方法重写，继承类 `InputStream`

**参数**

`b` -缓冲区中读取数据。

**结果**

读入缓冲的字节总数，或 `-1`如果没有更多的数据，因为已经到达文件末尾。

**异常**

`IOException`如果I/O错误发生。

**另请参见:**

`InputStream.read(byte[], int, int)`

**read**

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

读到 `len`字节从输入流到字节数组数据。如果 `len`不为零，该方法阻塞直到输入可用；否则，没有字节读取和 `0`返回。

**重写:**

`read` 方法重写，继承类 `InputStream`

**参数**

`b` -缓冲区中读取数据。

`off` -开始抵消目标数组 `b`

`len` -的最大字节数读。

**结果**

读入缓冲的字节总数，或 `-1`如果没有更多的数据，因为已经到达文件末尾。

**异常**

如果是 `null b` `NullPointerException`。

`IndexOutOfBoundsException` -如果 `off`是负的， `len`是负的，或 `len`大于 `b.length - off`

`IOException`如果I/O错误发生。

**另请参见:**

`InputStream.read()`

**skip**

```
public long skip(long n)
    throws IOException
```

跳过并丢弃 `n`字节输入流中的数据。

`skip`方法的可能，因为种种原因，最终跳过一些较小的字节数，可能`0`。如果`n`是负的，该方法将试图跳过向后。如果备份文件不支持向后跳到目前的位置，一个`IOException`抛出。跳过的字节数实际返回。如果它跳过了，它返回一个积极的价值。如果它跳过向后，它返回一个负值。

此方法可以跳过比备份文件中保留的字节数更多的字节数。这就产生了不例外的字节数不可能包括一些字节数，超出的支持文件的方法。尝试从流中读取，跳过过去结束后将导致`-1`表示该文件的结束。

**重写:**

`skip` 方法重写，继承类 `InputStream`

**参数**

n -字节数被跳过。

**结果**

跳过的实际字节数。

**异常**

`IOException`如果n是负的，如果流不支持查找，或者I/O错误发生。

**available**

```
public int available()
    throws IOException
```

返回一个剩余的字节数的估计，可以从这个输入流读取（或跳过），而不阻塞该输入流的方法的下次调用。o时返回的文件位置是在EOF。下一个调用可能是同一个线程或另一个线程。一个单一的读或跳过这许多字节将不会阻止，但可以读取或跳过更少的字节数。

在某些情况下，一个非阻塞的读（或跳过）可能会出现被阻塞时，它只是慢，例如，当阅读大文件的慢网络。

**重写:**

`available` 方法重写，继承类 `InputStream`

**结果**

可以从这个输入流中读取（或跳过）的剩余字节数的估计。

**异常**

`IOException` -如果这文件输入流被称 `close`或I/O错误关闭时。

**close**

```
public void close()
    throws IOException
```

关闭此文件输入流并释放与流关联的任何系统资源。

如果该流有一个相关的信道，则该信道也被关闭。

**Specified by:**

`close` 接口 `Closeable`

**Specified by:**

`close` 接口 `AutoCloseable`

**重写:**

`close` 方法重写，继承类 `InputStream`

**异常**

`IOException`如果I/O错误发生。

**getFD**

```
public final FileDescriptor getFD()
    throws IOException
```

返回表示实际的文件在文件系统中的 `FileInputStream`使用的连接的 `FileDescriptor`对象。

**结果**

与此流关联的文件描述符对象。

**异常**

`IOException`如果I/O错误发生。

**另请参见：**[FileDescriptor](#)**getChannel**

```
public FileChannel getChannel()
```

返回与此文件输入流有关的独特的 `FileChannel` 对象。

的返回信道的初始 `position` 将等于从文件中读取的字节数量为止。读取该数据流的字节将增加通道的位置。改变通道的位置，无论是显式或通过阅读，将改变这一流的文件位置。

**结果**

与此文件输入流关联的文件通道

**从以下版本开始：**

一点四

**finalize**

```
protected void finalize()
    throws IOException
```

确保 `close` 方法此文件输入流被调用时，没有对它的引用。

**重写：**

`finalize` 方法重写，继承类 `Object`

**异常**

`IOException` 如果 I/O 错误发生。

**另请参见：**[close\(\)](#)

Java™ Platform  
Standard Ed. 8

[概述](#) [软件包](#) [类](#) [使用](#) [树](#) [已过时的](#) [索引](#) [帮助](#)

[上一个](#) [下一个](#) [框架](#) [无框架](#) [所有类](#)

概要：嵌套字段 [CONSTR](#) | 方法      详细信息：字段 [CONSTR](#) | 方法

**[Submit a bug or feature](#)**

For further API reference and developer documentation, see [Java SE Documentation](#). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright © 1993, 2014, Oracle and/or its affiliates. All rights reserved.

本帮助文档是使用 [《百度翻译》](#) 翻译，请与英文版配合使用 by--QQ:654638585