

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

网站地址1

网站地址2

阿里云-服务器优惠

腾讯云-服务器优惠

Class **FileOutputStream**

java.lang.Object

java.io.OutputStream

java.io.FileOutputStream

All Implemented Interfaces:

Closeable, Flushable, AutoCloseable

public class **FileOutputStream**

extends OutputStream

文件输出流是一个 `File` 或一个 `FileDescriptor` 数据写入输出流。一个文件是否可用或可创建依赖于底层平台。特别是一些平台，允许文件被打开，只有一个 `FileOutputStream` 写作（或其他文件的写作对象）在一个时间。在这种情况下，如果所涉及的文件已经打开，这个类的构造函数将失败。

`FileOutputStream` 是写作的原始字节的图像数据流。书写的字符流，考虑使用 `FileWriter`。

从以下版本开始:

JDK1.0

另请参见:

`File`, `FileDescriptor`, `FileInputStream`, `Files.newOutputStream(java.nio.file.Path, java.nio.file.OpenOption...)`

构造方法摘要

构造方法

Constructor and Description

- `FileOutputStream(File file)`

创建一个文件输出流写入指定的 `File` 对象表示的文件。
- `FileOutputStream(File file, boolean append)`

创建一个文件输出流写入指定的 `File` 对象表示的文件。
- `FileOutputStream(FileDescriptor fdObj)`

创建一个文件输出流，写入指定的文件描述符，它表示在文件系统实际文件的现有连接。
- `FileOutputStream(String name)`

创建一个文件输出流，用指定的名称写入文件。
- `FileOutputStream(String name, boolean append)`

创建一个文件输出流，用指定的名称写入文件。

方法摘要

所有方法 接口方法 具体的方法

Modifier and Type

Method and Description

<code>void</code>	<code>close()</code> 关闭此文件输出流并释放与此流关联的任何系统资源。
<code>protected void</code>	<code>finalize()</code> 清理文件的连接，并确保此文件输出流的 <code>close</code> 方法被调用时，没有引用此流。
<code>FileChannel</code>	<code>getChannel()</code> 返回唯一 <code>FileChannel</code> 对象与此文件输出流相关。
<code>FileDescriptor</code>	<code>getFD()</code> 返回与此流关联的文件描述符。
<code>void</code>	<code>write(byte[] b)</code> 写 <code>b.length</code> 字节从指定的字节数组来此文件输出流。
<code>void</code>	<code>write(byte[] b, int off, int len)</code> 写 <code>len</code> 字节指定字节数组中的起始偏移 <code>off</code> 此文件输出流。
<code>void</code>	<code>write(int b)</code> 将指定的字节写入该文件输出流中。

Methods inherited from class `java.io.OutputStream`

`flush`

Methods inherited from class `java.lang.Object`

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

Constructor Detail

`FileOutputStream`

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

创建一个文件输出流，用指定的名称写入文件。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`checkWrite`方法被称为`name`作为参数。

如果该文件存在，但是是一个目录而不是常规文件，不存在但不能被创造，也不能打开任何其他理由，然后 `FileNotFoundException`抛出。

参数

`name` -系统依赖的文件名

异常

`FileNotFoundException` -如果文件存在，但是是一个目录而不是常规文件，不存在但不能被创造，也不能打开任何其他原因

`SecurityException` -如果存在一个安全管理及其 `checkWrite`方法拒绝写访问文件。

另请参见：

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

`FileOutputStream`

```
public FileOutputStream(String name,
    boolean append)
    throws FileNotFoundException
```

创建一个文件输出流，用指定的名称写入文件。如果第二 `true`，然后字节将被写入到文件的末尾而不是开头。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`checkWrite`方法被称为`name`作为参数。

如果该文件存在，但是是一个目录而不是常规文件，不存在但不能被创造，也不能打开任何其他理由，然后 `FileNotFoundException`抛出。

参数

`name` -系统相关的文件名称

`append` -如果 `true`，然后字节将被写入到文件的末尾而不是开头

异常

`FileNotFoundException` -如果文件存在，但是是一个目录而不是常规文件，不存在但不能被创造，也不能打开任何其他原因。

`SecurityException` -如果存在一个安全管理及其 `checkWrite`方法拒绝写访问文件。

从以下版本开始：

JDK1.1

另请参见：

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

FileOutputStream

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

创建一个文件输出流写入指定的 `File`对象表示的文件。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`checkWrite`方法调用的参数的`file`参数表示的路径。

如果该文件存在，但是是一个目录而不是常规文件，不存在但不能被创造，也不能打开任何其他理由，然后 `FileNotFoundException`抛出。

参数

`file` -要打开以进行写入的文件。

异常

`FileNotFoundException` -如果文件存在，但是是一个目录而不是常规文件，不存在但不能被创造，也不能打开任何其他原因

`SecurityException` -如果存在一个安全管理及其 `checkWrite`方法拒绝写访问文件。

另请参见：

`File.getPath()`， `SecurityException`， `SecurityManager.checkWrite(java.lang.String)`

FileOutputStream

```
public FileOutputStream(File file,
    boolean append)
    throws FileNotFoundException
```

创建一个文件输出流写入指定的 `File`对象表示的文件。如果第二 `true`，然后字节将被写入到文件的末尾而不是开头。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`checkWrite`方法调用的`file`参数作为它的参数表示的路径。

如果该文件存在，但是是一个目录而不是常规文件，不存在但不能被创造，也不能打开任何其他理由，然后 `FileNotFoundException`抛出。

参数

`file` -要打开以进行写入的文件。

`append` -如果 `true`，然后字节将被写入到文件的末尾而不是开头

异常

`FileNotFoundException` -如果文件存在，但是是一个目录而不是常规文件，不存在但不能被创造，也不能打开任何其他原因

`SecurityException` -如果存在一个安全管理及其 `checkWrite`方法拒绝写访问文件。

从以下版本开始：

一点四

另请参见：

`File.getPath()`， `SecurityException`， `SecurityManager.checkWrite(java.lang.String)`

FileOutputStream

```
public FileOutputStream(FileDescriptor fdObj)
```

创建一个文件输出流，写入指定的文件描述符，它表示在文件系统中的实际文件的现有连接。

首先，如果存在安全管理器，它的`checkWrite`方法被调用的文件描述符`fdObj`参数作为参数。

如果`fdObj`为`null`，然后`NullPointerException`抛出。

此构造函数不如果`fdObj`是`invalid`抛出异常。然而，如果调用的方法在流而尝试的I/O流，一个`IOException`抛出。

参数

`fdObj` -文件描述符是打开写作

异常

`SecurityException` -如果存在一个安全管理及其 `checkWrite`方法拒绝写访问的文件描述符

另请参见：

`SecurityManager.checkWrite(java.io.FileDescriptor)`

方法详细信息

write

```
public void write(int b)
    throws IOException
```

将指定的字节写入该文件输出流中。实现的方法 `OutputStream.write`。

Specified by:

`write` 方法重写，继承类 `OutputStream`

参数

`b`的字节被写入。

异常

`IOException`如果I/O错误发生。

write

```
public void write(byte[] b)
    throws IOException
```

写 `b.length`字节从指定的字节数组来此文件输出流。

重写：

`write` 方法重写, 继承类 `OutputStream`

参数

`b` -数据。

异常

`IOException`如果I/O错误发生。

另请参见:

`OutputStream.write(byte[], int, int)`

write

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

写 `len`字节指定字节数组中的起始偏移 `off`此文件输出流。

重写:

`write` 方法重写, 继承类 `OutputStream`

参数

`b` -数据。

`off`的起始偏移量的数据。

`len` -字节数写。

异常

`IOException`如果I/O错误发生。

close

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

关闭此文件输出流并释放与此流关联的任何系统资源。此文件输出流可能不再用于写入字节。

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 `OutputStream`

异常

`IOException`如果I/O错误发生。

getFD

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

返回与此流关联的文件描述符。

结果

表示文件在文件系统中的 `FileOutputStream`对象使用的连接 `FileDescriptor`对象。

异常

`IOException`如果I/O错误发生。

另请参见：

`FileDescriptor`

getChannel

```
public FileChannel getChannel()
```

返回与此文件输出流相关的独特的 `FileChannel`对象。

的返回信道的初始`position`将等于写入文件为止，除非这个流是追加方式的字节数，在这种情况下它将等于该文件的大小。写入该流的字节将相应增加信道的位置。改变通道的位置，无论是显式还是通过写，都会改变这个流的文件位置。

结果

与此文件输出流关联的文件通道

从以下版本开始：

一点四

finalize

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

清理文件的连接，并确保此文件输出流的 `close`方法被调用时，没有引用此流。

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：

`FileInputStream.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[CONSTRI](#)方法 详细信息：字段[CONSTRI](#)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