compact1, compact2, compact3
java.io

Class PrintStream

java.lang.Object
  java.io.OutputStream
    java.io.FilterOutputStream
      java.io.PrintStream

All Implemented Interfaces:

Closeable, Flushable, Appendable, AutoCloseable

已知直接子类:

LogStream

```
public class PrintStream
extends FilterOutputStream
implements Appendable, Closeable
```

一个 PrintStream添加功能到另一个输出流，即打印各种数据值表示的功能。还提供了两个其他功能。不像其他的输出流，一个 PrintStream从不抛出一个 IOException；相反，特殊情况下，只设置一个内部标志，可以通过 checkError测试方法。或者，可以创建一 PrintStream从而自动冲洗；这意味着 flush方法自动调用后一个字节数组写入，其中的 println方法被调用，或者一个换行字符或字节（ '\n'）写。

所有的字符转换成字节PrintStream印刷使用平台的默认字符编码。的PrintWriter类应该使用的情况下，需要写汉字而不是字节。

从以下版本开始:

JDK1.0

Field Summary

Fields inherited from class java.io.FilterOutputStream

out

构造方法摘要

构造方法

Constructor and Description

PrintStream(**File** file)
创建一个新的打印流，不带自动行刷新，用指定的文件。

PrintStream(**File** file, **String** csn)
创建一个新的打印流，无线自动冲洗，用指定的文件和字符集。

PrintStream(**OutputStream** out)
创建一个新的打印流。

PrintStream(**OutputStream** out, boolean autoFlush)
创建一个新的打印流。

PrintStream(**OutputStream** out, boolean autoFlush, **String** encoding)

创建一个新的打印流。

PrintStream(**String** fileName)
创建一个新的打印流，没有自动行刷新，用指定的文件名。

PrintStream(**String** fileName, **String** csn)
创建一个新的打印流，无线自动冲洗，用指定的文件名和字符集。

方法摘要

所有方法 **接口方法** **具体的方法**

Modifier and Type	Method and Description
PrintStream	**append**(char c) 将指定的字符输出流。
PrintStream	**append**(**CharSequence** csq) 将指定的字符序列的输出流。
PrintStream	**append**(**CharSequence** csq, int start, int end) 将指定的字符序列的子序列输出流。
boolean	**checkError**() 冲流和检查错误状态。
protected void	**clearError**() 清除此流的内部错误状态。
void	**close**() 关闭流。
void	**flush**() 冲流。
PrintStream	**format**(**Locale** l, **String** format, **Object**... args) 使用指定的格式字符串和参数将格式化的字符串写入到该输出流中。
PrintStream	**format**(**String** format, **Object**... args) 使用指定的格式字符串和参数将格式化的字符串写入到该输出流中。
void	**print**(boolean b) 打印布尔值。
void	**print**(char c) 打印一个字符。
void	**print**(char[] s) 打印一个字符数组。
void	**print**(double d) 打印一个双精度浮点数。
void	**print**(float f) 打印一个浮点数。
void	**print**(int i) 打印一个整数。
void	**print**(long l) 打印一个长整数。
void	**print**(**Object** obj) 打印一个对象。
void	**print**(**String** s) 打印一个字符串。
PrintStream	**printf**(**Locale** l, **String** format, **Object**... args) 使用指定的格式字符串和参数将格式化的字符串写入该输出流的方便方法。

PrintStream	**printf**(String format, Object... args) 使用指定的格式字符串和参数将格式化的字符串写入该输出流的方便方法。
void	**println**() 通过编写行分隔符终止当前行。
void	**println**(boolean x) 打印一个布尔值，然后终止该行。
void	**println**(char x) 打印一个字符，然后终止行。
void	**println**(char[] x) 打印一个字符数组，然后终止行。
void	**println**(double x) 打印一个双，然后终止线。
void	**println**(float x) 打印一个浮动，然后终止行。
void	**println**(int x) 打印一个整数，然后终止该行。
void	**println**(long x) 打印一个长，然后终止线。
void	**println**(Object x) 打印一个对象，然后终止该行。
void	**println**(String x) 打印一个字符串，然后终止该行。
protected void	**setError**() 集 true流的错误状态。
void	**write**(byte[] buf, int off, int len) 写 len字节指定字节数组中的起始偏移 off这个流。
void	**write**(int b) 将指定的字节写入该流中。

Methods inherited from class java.io.FilterOutputStream

write

Methods inherited from class java.lang.Object

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

Constructor Detail

PrintStream

```
public PrintStream(OutputStream out)
```

创建一个新的打印流。这条流不会自动刷新。

参数

out -输出流值和对象将被打印

另请参见:

PrintWriter.PrintWriter(java.io.OutputStream)

PrintStream

```
public PrintStream(OutputStream out,
                   boolean autoFlush)
```

创建一个新的打印流。

参数

`out` -输出流值和对象将被打印

`autoFlush` -布尔；如果属实，输出缓冲区将被刷新时，字节数组写入，其中的 `println`方法被调用，或者一个换行字符或字节（ `'\n'`）写

另请参见:

`PrintWriter.PrintWriter(java.io.OutputStream, boolean)`

PrintStream

```
public PrintStream(OutputStream out,
                   boolean autoFlush,
                   String encoding)
            throws UnsupportedEncodingException
```

创建一个新的打印流。

参数

`out` -输出流值和对象将被打印

`autoFlush` -布尔；如果属实，输出缓冲区将被刷新时，字节数组写入，其中的 `println`方法被调用，或者一个换行字符或字节（ `'\n'`）写

`encoding` -支持的 `character encoding`名称

异常

`UnsupportedEncodingException` -如果指定的编码不支持

从以下版本开始:

一点四

PrintStream

```
public PrintStream(String fileName)
            throws FileNotFoundException
```

创建一个新的打印流，没有自动行刷新，用指定的文件名。这种便利的构造函数创建必要的中间 `OutputStreamWriter`，将编码字符使用 default charset这个java虚拟机实例。

参数

`fileName` -该文件的名称作为本刊流的目的地。如果文件存在，则它将被截断为零；否则，将创建一个新的文件。输出将被写入到文件，并缓冲。

异常

`FileNotFoundException` -如果指定文件对象不表示一个存在，可写普通文件和一个新的名字不能创造规则的文件，或者一些其他的错误发生在打开或创建文件

`SecurityException` -如果一个安全管理是当前和 `checkWrite(fileName)`否认文件的写访问

从以下版本开始:

一点五

PrintStream

```
public PrintStream(String fileName,
```

```
                String csn)
            throws FileNotFoundException,
                UnsupportedEncodingException
```

创建一个新的打印流，无线自动冲洗，用指定的文件名和字符集。这种便利的构造函数创建必要的中间 `OutputStreamWriter`，将编码字符的使用提供的字符集。

参数

`fileName` -该文件的名称作为本刊流的目的地。如果文件存在，则它将被截断为零；否则，将创建一个新的文件。输出将被写入到文件，并缓冲。

`csn` -支持的 `charset`名称

异常

`FileNotFoundException` -如果指定文件对象不表示一个存在，可写普通文件和一个新的名字不能创造规则的文件，或者一些其他的错误发生在打开或创建文件

`SecurityException` -如果一个安全管理是当前和 `checkWrite(fileName)`否认文件的写访问

`UnsupportedEncodingException` -如果指定的字符集不支持

从以下版本开始:

一点五

PrintStream

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

创建一个新的打印流，不带自动行刷新，用指定的文件。这种便利的构造函数创建必要的中间 `OutputStreamWriter`，将编码字符使用 default charset这个java虚拟机实例。

参数

`file` -文件作为本刊流的目的地。如果文件存在，则它将被截断为零；否则，将创建一个新的文件。输出将被写入到文件，并缓冲。

异常

`FileNotFoundException` -如果指定文件对象不表示一个存在，可写普通文件和一个新的名字不能创造规则的文件，或者一些其他的错误发生在打开或创建文件

`SecurityException` -如果一个安全管理是当前和 `checkWrite(file.getPath())`否认文件的写访问

从以下版本开始:

一点五

PrintStream

```
public PrintStream(File file,
                String csn)
            throws FileNotFoundException,
                UnsupportedEncodingException
```

创建一个新的打印流，无线自动冲洗，用指定的文件和字符集。这种便利的构造函数创建必要的中间 `OutputStreamWriter`，将编码字符的使用提供的字符集。

参数

`file` -文件作为本刊流的目的地。如果文件存在，则它将被截断为零；否则，将创建一个新的文件。输出将被写入到文件，并缓冲。

`csn` -支持的 `charset`名称

异常

`FileNotFoundException` -如果指定文件对象不表示一个存在，可写普通文件和一个新的名字不能创造规则的文件，或者一些其他的错误发生在打开或创建文件

`SecurityException` -如果一个安全管理是当前和 `checkWrite(file.getPath())`否认文件的写访问

`UnsupportedEncodingException` -如果指定的字符集不支持

从以下版本开始:

一点五

方法详细信息

flush

```
public void flush()
```

冲流。这是通过编写任何缓冲输出字节到下一个输出流，然后刷新该流。

Specified by:

`flush` 接口 `Flushable`

重写:

`flush` 方法重写，继承类 `FilterOutputStream`

另请参见:

`OutputStream.flush()`

close

```
public void close()
```

关闭流。这是通过冲洗流，然后关闭底层的输出流。

Specified by:

`close` 接口 `Closeable`

Specified by:

`close` 接口 `AutoCloseable`

重写:

`close` 方法重写，继承类 `FilterOutputStream`

另请参见:

`OutputStream.close()`

checkError

```
public boolean checkError()
```

冲流和检查错误状态。内部错误状态设置为 `true`当底层输出流将比其他 `InterruptedIOException`一 `IOException`，当 `setError`方法被调用。如果在底层输出流抛出一个 `InterruptedIOException`操作，然后转换回 `PrintStream`异常中断做

线。`currentthread()。interrupt();`

或等效。

结果

`true`当且仅当该流遇到了比其他 `InterruptedIOException`一 `IOException`，或 `setError`方法被调用

setError

```
protected void setError()
```

集 `true`流的错误状态。

println

```
public void println(char x)
```

打印一个字符，然后终止行。这种方法就像是调用 `print(char)`然后 `println()`。

参数

`x` - char要打印。

println

```
public void println(int x)
```

打印一个整数，然后终止该行。这种方法就像是调用 `print(int)`然后 `println()`。

参数

`x` - int要打印。

println

```
public void println(long x)
```

打印一个长，然后终止线。这种方法就像是调用 `print(long)`然后 `println()`。

参数

`x` -一 long要打印。

println

```
public void println(float x)
```

打印一个浮动，然后终止行。这种方法就像是调用 `print(float)`然后 `println()`。

参数

`x` - float要打印。

println

```
public void println(double x)
```

打印一个双，然后终止线。这种方法就像是调用 `print(double)`然后 `println()`。

参数

`x` - double要打印。

println

```
public void println(char[] x)
```

打印一个字符数组，然后终止行。这种方法就像是调用 `print(char[])`然后 `println()`。

参数

`x` -字符打印数组。

println

```
public void println(String x)
```

打印一个字符串，然后终止该行。这种方法就像是调用 print(String)然后 println()。

参数

x - String要打印。

println

```
public void println(Object x)
```

打印一个对象，然后终止该行。该方法在第一个字符串调用。值（x）得到印刷对象的字符串值，然后就像是调用 print(String)然后 println()。

参数

x - Object要打印。

printf

```
public PrintStream printf(String format,
                          Object... args)
```

使用指定的格式字符串和参数将格式化的字符串写入该输出流的方便方法。

这种形式的out.printf(format, args)方法调用的行为一样，调用

出格式（格式，args）。

参数

format -格式字符串中所描述的 Format string syntax

args由格式字符串的格式说明符引用的论据。如果有比格式说明符的更多参数，多余的参数会被忽略。参数的数量是可变的，可能是零。参数的最大数量是由一个java数组的定义由 *The Java™ Virtual Machine Specification*最大尺寸的限制。在null争论的行为取决于 conversion。

结果

输出流

异常

IllegalFormatException如果格式字符串包含一个非法的语法、格式说明符与给定的参数不兼容，论据不足给定的格式字符串，或者其他非法的条件。对于所有可能的格式错误的规范，看到格式化程序类规范的 Details节。

NullPointerException -如果 format是 null

从以下版本开始：

一点五

printf

```
public PrintStream printf(Locale l,
                          String format,
                          Object... args)
```

使用指定的格式字符串和参数将格式化的字符串写入该输出流的方便方法。

这种形式的out.printf(l, format, args)方法调用的行为一样，调用

出来。格式（L、格式、参数）

参数

l - locale申请过程中的格式。如果 l是 null然后没有本地化的应用。

format -格式字符串中所描述的 Format string syntax

args由格式字符串的格式说明符引用的论据。如果有比格式说明符的更多参数，多余的参数会被忽略。参数的数量是可变的，可能是零。参数的最大数量是由一个java数组的定义由 *The Java™ Virtual Machine Specification*最大尺寸的限制。在null争论的行为取决于 conversion。

结果

输出流

异常

IllegalFormatException如果格式字符串包含一个非法的语法、格式说明符与给定的参数不兼容，论据不足给定的格式字符串，或者其他非法的条件。对于所有可能的格式错误的规范，看到格式化程序类规范的 Details节。

NullPointerException -如果 format是 null

从以下版本开始:

一点五

format

```
public PrintStream format(String format,
                          Object... args)
```

使用指定的格式字符串和参数将格式化的字符串写入到该输出流中。

经常使用的区域是一个由Locale.getDefault()，无论以前的任何调用此对象的其他格式的方法。

参数

format -格式字符串中所描述的 Format string syntax

args由格式字符串的格式说明符引用的论据。如果有比格式说明符的更多参数，多余的参数会被忽略。参数的数量是可变的，可能是零。参数的最大数量是由一个java数组的定义由 *The Java™ Virtual Machine Specification*最大尺寸的限制。在null争论的行为取决于 conversion。

结果

输出流

异常

IllegalFormatException如果格式字符串包含一个非法的语法、格式说明符与给定的参数不兼容，论据不足给定的格式字符串，或者其他非法的条件。对于所有可能的格式错误的规范，看到格式化程序类规范的 Details节。

NullPointerException -如果 format是 null

从以下版本开始:

一点五

format

```
public PrintStream format(Locale l,
                          String format,
                          Object... args)
```

使用指定的格式字符串和参数将格式化的字符串写入到该输出流中。

参数

l - locale申请过程中的格式。如果 l是 null然后不采用定位。

format -格式字符串中所描述的 Format string syntax

args由格式字符串的格式说明符引用的论据。如果有比格式说明符的更多参数，多余的参数会被忽略。参数的数量是可变的，可能是零。参数的最大数量是由一个java数组的定义由 *The Java™ Virtual Machine Specification*最大尺寸限制。在 null争论的行为取决于 conversion。

结果

输出流

异常

IllegalFormatException如果格式字符串包含一个非法的语法、格式说明符与给定的参数不兼容，论据不足给定的格式字符串，或者其他非法的条件。对于所有可能的格式错误的规范，看到格式化程序类规范的 Details节。

NullPointerException -如果 format是 null

从以下版本开始:

一点五

append

```
public PrintStream append(CharSequence csq)
```

将指定的字符序列的输出流。

这种形式的out.append(csq)方法调用的行为一样，调用

出来。打印（CSQ。tostring()）

根据字符序列的csq toString规范，整个序列可能不追加。例如，调用然后字符缓冲区toString方法将返回一个序列的内容取决于缓冲区的位置和范围。

Specified by:

append 接口 Appendable

参数

csq -字符序列添加。如果 csq是 null，然后四个字 "null"附加到输出流。

结果

输出流

从以下版本开始:

一点五

append

```
public PrintStream append(CharSequence csq,
                          int start,
                          int end)
```

将指定的字符序列的子序列输出流。

调用这个方法时，csq形式out.append(csq, start, end)不null，表现在完全相同的方式调用

出来。打印（CSQ。子序列（开始、结束）。tostring()）

Specified by:

append 接口 Appendable

参数

csq -字符序列的子序列将追加。如果 csq是 null，那么人物将追加好像 csq包含四个字符 "null"。

start -子序列中的第一个字符的索引

end -子序列中的最后一个字符后的字符的索引

结果

输出流

异常

IndexOutOfBoundsException -如果 start或 end是负面的， start大于 end，或 end大于 csq.length()

从以下版本开始:

一点五

append

```
public PrintStream append(char c)
```

将指定的字符输出流。

这种形式的out.append(c)方法调用的行为一样，调用

打印（c）

Specified by:

append 接口 Appendable

参数

c - 16位字符追加

结果

输出流

从以下版本开始:

一点五