compact1, compact2, compact3
java.io

Class BufferedWriter

java.lang.Object
  java.io.Writer
    java.io.BufferedWriter

All Implemented Interfaces:

Closeable, Flushable, Appendable, AutoCloseable

```
public class BufferedWriter
extends Writer
```

将文本写入到字符输出流中，缓冲字符，以便提供对单个字符、数组和字符串的有效写入。

可以指定缓冲区大小，也可以接受默认大小。默认是足够大的用于大多数目的。

提供了一种newline()方法，利用平台自身观念的行分隔符由系统性line.separator定义。并不是所有的平台都使用换行符（'\n'）终止线。调用此方法终止各输出线因此宁愿写一个换行符直接。

在一般情况下，一个作家发送其输出立即到底层字符或字节流。除非及时输出是必需的，最好是用BufferedWriter围绕其write()操作可能是昂贵的任何作家，如filewriters和outputstreamwriters。例如，

```
PrintWriter出来=新PrintWriter（新BufferedWriter（新FileWriter（“foo”））；
```

缓存PrintWriter输出文件。无缓冲，每次调用一个方法会导致print()字符被转换成字节将被立即写入文件，它可以是非常低效的。

从以下版本开始：

JDK1.1

另请参见：

PrintWriter, FileWriter, OutputStreamWriter, Files.newBufferedWriter(java.nio.file.Path, java.nio.charset.Charset, java.nio.file.OpenOption...)

Field Summary

Fields inherited from class java.io.Writer

lock

构造方法摘要

构造方法

Constructor and Description

BufferedWriter(Writer out)
创建一个使用默认大小输出缓冲区的缓冲字符输出流。

BufferedWriter(Writer out, int sz)
创建一个新的缓冲字符输出流，该流使用给定大小的输出缓冲区。

方法摘要

所有方法 **接口方法** **具体的方法**

Modifier and Type	Method and Description
void	**close**() 关闭流，冲洗它。
void	**flush**() 冲流。
void	**newLine**() 写行分隔符。
void	**write**(char[] cbuf, int off, int len) 写入一个字符数组的一部分。
void	**write**(int c) 写一个字符。
void	**write**(**String** s, int off, int len) 写入字符串的一部分。

Methods inherited from class java.io.Writer

append, append, append, write, write

Methods inherited from class java.lang.Object

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

Constructor Detail

BufferedWriter

```
public BufferedWriter(Writer out)
```

创建一个使用默认大小输出缓冲区的缓冲字符输出流。

参数

out -作家

BufferedWriter

```
public BufferedWriter(Writer out,
                      int sz)
```

创建一个新的缓冲字符输出流，该流使用给定大小的输出缓冲区。

参数

out -作家

sz输出缓冲区的大小，一个正整数

异常

如果 sz <= 0 IllegalArgumentException

方法详细信息

write

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

写一个字符。

重写:

`write` 方法重写，继承类 `Writer`

参数

`c` - int指定要写入的字符

异常

`IOException`如果I/O错误发生

write

```
public void write(char[] cbuf,
                  int off,
                  int len)
           throws IOException
```

写入一个字符数组的一部分。

通常，此方法将给定数组中的字符存储到该流的缓冲区中，并根据需要将缓冲区刷新到基础流中。如果所请求的长度至少和缓冲区一样大，那么这个方法将刷新缓冲区，并直接将字符写入到基础流中。因此`BufferedWriter`s不会复制不必要的数据冗余。

Specified by:

`write` 方法重写，继承类 `Writer`

参数

`cbuf` -字符数组

`off`偏移开始读取字符的

`len`数字写

异常

`IOException`如果I/O错误发生

write

```
public void write(String s,
                  int off,
                  int len)
           throws IOException
```

写入字符串的一部分。

如果该`len`参数值为负则没有书写文字。这是违反本法在superclass规范，这需要一个`IndexOutOfBoundsException`扔。

重写:

`write` 方法重写，继承类 `Writer`

参数

`s`字符串被写入

`off`偏移开始读取字符的

len数字要写

异常

IOException如果I/O错误发生

newLine

```
public void newLine()
             throws IOException
```

写行分隔符。行分隔符字符串是由系统性 line.separator定义，而不一定是一个换行符（'\n'）字符。

异常

IOException如果I/O错误发生

flush

```
public void flush()
           throws IOException
```

冲流。

Specified by:

flush 接口 Flushable

Specified by:

flush 方法重写，继承类 Writer

异常

IOException如果I/O错误发生

close

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

从阶级复制的描述： **Writer**
关闭流，冲洗它。一旦流已经关闭，进一步write()或flush()调用将导致资源被。关闭先前关闭的流没有影响。

Specified by:

close 接口 Closeable

Specified by:

close 接口 AutoCloseable

Specified by:

close 方法重写，继承类 Writer

异常

IOException如果I/O错误发生

code examples.