java.io

**Class BufferedWriter**

* [java.lang.Object](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html)
  + [java.io.Writer](https://docs.oracle.com/javase/8/docs/api/java/io/Writer.html)
    - java.io.BufferedWriter
* **All Implemented Interfaces:**

[Closeable](https://docs.oracle.com/javase/8/docs/api/java/io/Closeable.html), [Flushable](https://docs.oracle.com/javase/8/docs/api/java/io/Flushable.html), [Appendable](https://docs.oracle.com/javase/8/docs/api/java/lang/Appendable.html), [AutoCloseable](https://docs.oracle.com/javase/8/docs/api/java/lang/AutoCloseable.html)

BufferedWriter是一个缓存的字符输出流。放到流中的内容不会直接写到目的地（如文件，数组），而是先缓存起来，然后通过flush方法或者在一个时间段，将所有缓存中的内容，一块写入到目的地。这样比使用普通的输出流输入一个字符就马上写到目的地，提高了效率。

举例：

普通输出流：执行一个wirte方法，马上往文件中写一个字符。

Buffered的输出流：执行write方法后的左右字符，都先放到一个缓存中（数组）中。然后通过flush方法一块写到文件中。

缓存的大小可以指定也可以使用默认的大小（8192）。

构造方法

|  |
| --- |
| **Constructor and Description** |
| [**BufferedWriter**](https://docs.oracle.com/javase/8/docs/api/java/io/BufferedWriter.html#BufferedWriter-java.io.Writer-)([**Writer**](https://docs.oracle.com/javase/8/docs/api/java/io/Writer.html) out)  只要是Writer的子类，都可以作为参数来进行构造。多态特性 |
| [**BufferedWriter**](https://docs.oracle.com/javase/8/docs/api/java/io/BufferedWriter.html#BufferedWriter-java.io.Writer-int-)([**Writer**](https://docs.oracle.com/javase/8/docs/api/java/io/Writer.html) out, int sz)  通过子类作为参数进行构造，并且通过sz来指定缓存的大小。 |

|  |
| --- |
| 实例 |
| **public** **static** **void** main(String[] args) **throws** IOException{  **char**[] chars = {'a','b','c','d'};  CharArrayWriter arrayWriter = **new** CharArrayWriter();//Writer的子类  BufferedWriter writer = **new** BufferedWriter(arrayWriter);  } |

|  |  |
| --- | --- |
| void | [**close**](https://docs.oracle.com/javase/8/docs/api/java/io/BufferedWriter.html#close--)()  先将流中的内容写到目的地，然后关闭流 |

|  |
| --- |
| 实例： |
| **public** **static** **void** main(String[] args) **throws** IOException{  //Writer的子类，向这个输出流中写入的内容，自动放到它内部自动构建的一个数组中。  CharArrayWriter arrayWriter = **new** CharArrayWriter();  BufferedWriter writer = **new** BufferedWriter(arrayWriter);  writer.write(65);  writer.close();//先把65（字符A）放到数组中，然后关闭  System.***out***.println(arrayWriter.toString());//A  } |

|  |  |
| --- | --- |
| void | [**flush**](https://docs.oracle.com/javase/8/docs/api/java/io/BufferedWriter.html#flush--)()  将缓存中的内容一并放到目的地。 |

|  |
| --- |
| 实例 |
| **public** **static** **void** main(String[] args) **throws** IOException{  //Writer的子类，向这个输出流中写入的内容，自动放到它内部自动构建的一个数组中。  CharArrayWriter arrayWriter = **new** CharArrayWriter();  BufferedWriter writer = **new** BufferedWriter(arrayWriter);  writer.write(65);  writer.flush();//不用此方法，会发现打印不出A  System.***out***.println(arrayWriter.toString());//A  } |

|  |  |
| --- | --- |
| void | [**newLine**](https://docs.oracle.com/javase/8/docs/api/java/io/BufferedWriter.html#newLine--)()  写入一个换行符，比如向文件中写东西时，使用这个方法，可以是内容换到新的一行，然后再输入内容。 |

|  |
| --- |
| 实例 |
| **public** **static** **void** main(String[] args) **throws** IOException{  //Writer的子类，向这个输出流中写入的内容，自动放到它内部自动构建的一个数组中。  CharArrayWriter arrayWriter = **new** CharArrayWriter();  BufferedWriter writer = **new** BufferedWriter(arrayWriter);  writer.write(65);  //写入一个换行符（从控制台来看就如同使用了printf。）  //如果不用此方法，会发现最后AB在同一行上显示  writer.newLine();  writer.write(66);  writer.flush();//不用此方法，会发现打印不出A  System.***out***.println(arrayWriter.toString());//AB分别各占一行  } |

|  |  |
| --- | --- |
| void | [**write**](https://docs.oracle.com/javase/8/docs/api/java/io/BufferedWriter.html#write-int-)(int c)  向输出流写入一个字符。c表示字符的ASCII码。实例参照上方 |

|  |  |
| --- | --- |
| void | [**write**](https://docs.oracle.com/javase/8/docs/api/java/io/BufferedWriter.html#write-char:A-int-int-)(char[] cbuf, int off, int len)  从数组cbuf中index为off的位置开始，取出len个字符，写入到输出流中。 |

|  |
| --- |
| 实例：可先学习CharArrayWriter后，再看此例 |
| **public** **static** **void** main(String[] args) **throws** IOException{  **char**[] chars ={'a','b','c'};  //Writer的子类，向这个输出流中写入的内容，自动放到它内部自动构建的一个数组中。  CharArrayWriter arrayWriter = **new** CharArrayWriter();  BufferedWriter writer = **new** BufferedWriter(arrayWriter);  writer.write(chars,2,1);  writer.flush();  System.***out***.println(arrayWriter.toString());//c  } |

|  |  |
| --- | --- |
| void | [**write**](https://docs.oracle.com/javase/8/docs/api/java/io/BufferedWriter.html#write-java.lang.String-int-int-)([**String**](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) s, int off, int len)  从字符串s中index为off的位置开始，取len个字符，放到输出流中。 |

|  |
| --- |
| 实例 |
| **public** **static** **void** main(String[] args) **throws** IOException{  String str = "helloworld";  //Writer的子类，向这个输出流中写入的内容，自动放到它内部自动构建的一个数组中。  CharArrayWriter arrayWriter = **new** CharArrayWriter();  BufferedWriter writer = **new** BufferedWriter(arrayWriter);  writer.write(str,5,5);  writer.flush();  System.***out***.println(arrayWriter.toString());//world  } |

其他继承与父类的Writer的方法，请参考Writer的文档