**Class ByteArrayInputStream**

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

[Closeable](https://docs.oracle.com/javase/8/docs/api/java/io/Closeable.html), [AutoCloseable](https://docs.oracle.com/javase/8/docs/api/java/lang/AutoCloseable.html" \o "interface in java.lang)

|  |  |
| --- | --- |
| void | [**mark**](https://docs.oracle.com/javase/8/docs/api/java/io/ByteArrayInputStream.html#mark-int-)(int readAheadLimit) |
| void | **reset()** |

两个方法一般配合使用.mark方法的作用是在当前文职做一个标记,当使用reset方法时,会从mark方法标记的地方开始读取读取流中的内容.

关于mark方法中的参数readAheadLimit,跟mark方法本身没关系,参数值的设置是用来在其他方法中使用的.具体参考其他本章.

|  |
| --- |
| 实例 |
| **public** **class** *Test* {  **public** **static** **void** **main**(*String*[] args) **throws** *IOException* {  **byte**[] b = **new** **byte**[] { 1, 2, 3 ,4 };  *ByteArrayInputStream* bis = **new** **ByteArrayInputStream**(b);  *System*.***out***.**println**(bis.**read**()); //1  *System*.***out***.**println**(bis.**read**()); //2    bis.**mark**(90000);//reset方法后，定位在3上，以后对流中数据的操作会从3开始  *System*.***out***.**println**(bis.**read**()); //3  *System*.***out***.**println**(bis.**read**()); //4    bis.**reset**(); //从mark定位的2开始输出  *System*.***out***.**println**(bis.**read**()); //3  *System*.***out***.**println**(bis.**read**()); //4  }  } |

注意:mark(9000)并不是在流中index为9000的地方做标记。这个参数跟mark方法没关系。

|  |  |
| --- | --- |
| int | [**available**](https://docs.oracle.com/javase/8/docs/api/java/io/ByteArrayInputStream.html#available--)()  返回字节流中还可以被读取的字节的个数。 |

|  |
| --- |
| 实例 |
| **public** **class** *Test* {  **public** **static** **void** **main**(*String*[] args) **throws** *IOException* {  **byte**[] b = **new** **byte**[] { 1, 2,3 };  *ByteArrayInputStream* bis = **new** **ByteArrayInputStream**(b);  *System*.***out***.**println**(bis.**read**()); //1  *System*.***out***.**println**(bis.**available**());//返回1,字节流中还剩2个字节可以被读取  *System*.***out***.**println**(bis.**read**()); //2  *System*.***out***.**println**(bis.**read**()); //3  *System*.***out***.**println**(bis.**available**());//返回0,字节流中已经么有可以被读取的字节了  }  } |

|  |  |
| --- | --- |
| int | [**close**](https://docs.oracle.com/javase/8/docs/api/java/io/ByteArrayInputStream.html#available--)()  ByteArrayInputStream这个方法没有用，因为没有任何实现 |

|  |  |
| --- | --- |
| int | read ()  从流中读取字节。 |

|  |  |
| --- | --- |
| int | read(byte[] b, int off, int len)  从流的当前位置，读取出len个字节，放到新数组b中，  从b中哪个位置（索引值）开始放由参数off指定，  首先Off+len不能超过b的容量，否则回报异常。  然后，如果b的pos（当前位置）等于b的count，也就是说已经到了b的end file位置，返回-1  其他约束看api |

|  |
| --- |
| 实例 |
| **public** **class** *Test* {  **public** **static** **void** **main**(*String*[] args) **throws** *IOException* {  **byte**[] b = **new** **byte**[] { 1, 2,3 };  *ByteArrayInputStream* bis = **new** **ByteArrayInputStream**(b);  bis.**mark**(1);//reset后，可以继续从流的开头开始操作流  bis.**skip**(3);  **byte**[] c = **new** **byte**[5];//初始化后c为[0,0,0,0,0]    //返回值为-1，因为前面skip（3）已经到了流的结尾了。数组c中没有被添加任何东西，因为没有从b中读取到东西。  *System*.***out***.**println**(bis.**read**(c,2, 3)); //-1    bis.**reset**();  *System*.***out***.**println**(bis.**read**(c,2, 3)); //[0,0,1,2,3]    bis.**reset**();  *System*.***out***.**println**(bis.**read**(c,2, 4));//异常。off+len(0+4)大于b的长度了  }  } |

|  |  |
| --- | --- |
| Long | skip (long n)  从流中当前位置跳过n个byte。 |

|  |
| --- |
| 实例 |
| **public** **class** *Test* {  **public** **static** **void** **main**(*String*[] args) **throws** *IOException* {  **byte**[] b = **new** **byte**[] { 1, 2,3 };  *ByteArrayInputStream* bis = **new** **ByteArrayInputStream**(b);  *System*.***out***.**println**(bis.**skip**(2));  *System*.***out***.**println**(bis.**read**()); //3  }  } |