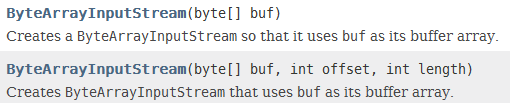
数组流（字节和字符）

1. 操作数组的**源和目的**都是**内存，操作数组的IO流并没有调用底层资源，因此无需关闭这样的流，关闭无效，它们的close方法是空的，调用了close方法，也可以继续使用**。
2. **字符串可以通过方法getBytes方法转换成字节数据。**
3. 操作**字节数组**的IO流：**ByteArrayInputStream和ByteArrayOutputStream。**

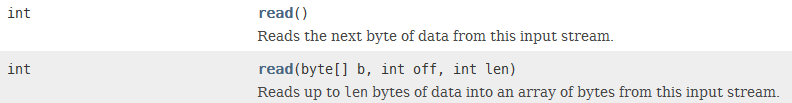
**这两个IO流分别继承于InputStream和OutputStream流。**

1. **ByteArrayInputStream**
2. **构造方法：源是内存，参数是字节数组。**

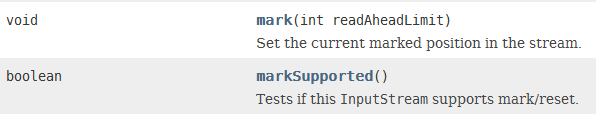


1. **成员方法：**
   1. **read方法：**

**read（byte[] b）：继承于InputStream的。**



* 1. **mark方法：**



* 1. **skip方法：**



* 1. **reset方法：**

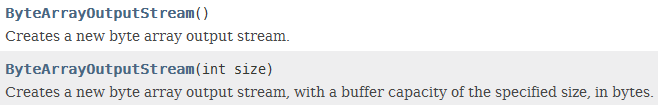


* 1. **available方法：**

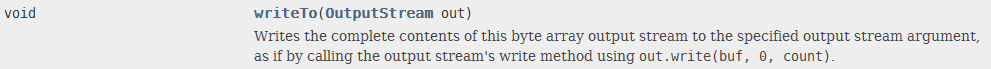


* 1. **close**

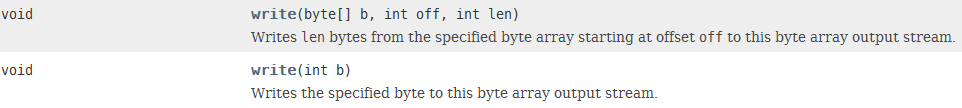
1. **ByteArrayOutputStream：目标也是内存，目的地不需要指明，系统会自动创建一个字节数组。获取该数组的方法就是**toByteArray()方法。
2. **构造方法：向内存中写入数据，不需要指定目的地，默认情况下，在内存中自动创建一个数组。**



1. **成员方法：**
   1. **writeTo方法：把数据写到其他输出流中。**



* 1. **write方法：**



* 1. toString()方法：





* 1. toByteArray()方法：获取**目的数组**。



* 1. size方法：



* 1. reset方法：



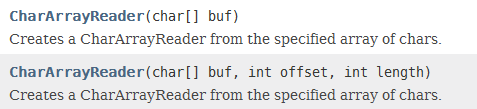
* 1. close方法

1. 操作**字符数组**的IO流：**CharArrayReader和CharArrayWriter。**

**这连个IO流分别继承于Reader和Writer。与字节输入输出流对应。**

* + 1. **CharArrayReader：**

1. **构造方法：**

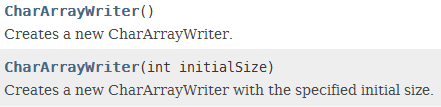


1. **成员方法:**

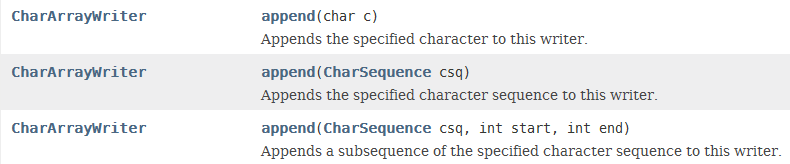
**read方法、reset方法、ready方法、skip方法、mark方法、close方法**

* + 1. **CharArrayWriter：同ByteArrayOutputStream一样，不需要指明目的地，自动创建一个字符数组，获取此结果数组的方法是toCharArray。**

1. **构造方法：不需要指明目的。**



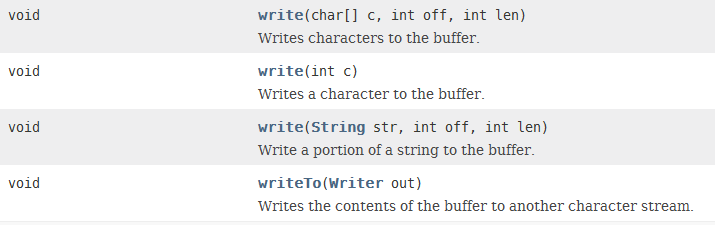
1. **成员方法：**
   1. **append方法：**



* 1. **toCharArray：获取结果的字符数组。**



* 1. **write方法:**



* 1. **reset 方法、flush、close方法等**

1. 示例：

byte[] buf = "abcdeg".getBytes();

ByteArrayInputStream bais = new ByteArrayInputStream(buf);

ByteArrayOutputStream baos = new ByteArrayOutputStream();

int b ;

while((b = bais.read())!= -1) {

baos.write(b);

}

System.out.println(baos.toString());

1. 练习题：当然**输入数组流**也可以利用缓冲流进行包装，但是**输出数组流**不能用缓冲流进行包装。
   * 1. 文件—》程序—》内存（字节数组）

字节操作：

**private** **static** **byte**[] copyFileToByteArray(File file) **throws** IOException {

InputStream is = **new** BufferedInputStream(**new** FileInputStream(file));

ByteArrayOutputStream baos = **new** ByteArrayOutputStream();

// OutputStream baos = new BufferedOutputStream(new ByteArrayOutputStream());

// java.lang.ClassCastException: java.io.BufferedOutputStream cannot be cast to java.io.ByteArrayOutputStream

**byte**[] buff = **new** **byte**[4];

**int** len = 0;

**while**(-1 != (len = is.read(buff))) {

baos.write(buff,0,len);

}

// byte[] result = ((ByteArrayOutputStream) baos).toByteArray();

**byte**[] result = baos.toByteArray();

is.close();

**return** result;

}

字符操作：

**private** **static** **char**[] copyFileToCharArray(File file) **throws** IOException {

Reader reader = **new** BufferedReader(**new** FileReader(file));

// Writer caw = new BufferedWriter(new CharArrayWriter());

// java.lang.ClassCastException: java.io.BufferedWriter cannot be cast to java.io.CharArrayWriter

CharArrayWriter caw = **new** CharArrayWriter();

**char**[] buff = **new** **char**[5];

**int** len = 0;

**while**(-1 !=(len = reader.read(buff))){

caw.write(buff,0,len);

}

**char**[] result = caw.toCharArray();

reader.close();

**return** result;

}测试：

**public** **static** **void** main(String[] args) **throws** IOException {

File file = **new** File("e:\\aaa.txt");

**byte**[] bs = *copyFileToByteArray*(file);

System.***out***.println(Arrays.*toString*(bs));

**char**[] cs = *copyFileToCharArray*(file);

System.***out***.println(**new** String(cs));

}

结果：



* + 1. 字符数组（内存）—》程序—》文件

字节操作：

**private** **static** **void** copyCharArrayToFile(**char**[] cs, File dest2) **throws** IOException {

// CharArrayReader car = new CharArrayReader(cs);

Reader car = **new** BufferedReader(**new** CharArrayReader(cs));

Writer writer = **new** BufferedWriter(**new** FileWriter(dest2));

**char**[] buff = **new** **char**[6];

**int** len = 0;

**while**(-1 != (len = car.read(buff))) {

writer.write(buff,0,len);

}

writer.flush();

writer.close();

}字符操作：

**private** **static** **void** copyByteArrayToFile(**byte**[] bs,File dest) **throws** IOException {

// ByteArrayInputStream bais = new ByteArrayInputStream(bs);

InputStream bais = **new** BufferedInputStream( **new** ByteArrayInputStream(bs));

OutputStream os = **new** BufferedOutputStream(**new** FileOutputStream(dest));

**byte**[] buff = **new** **byte**[4];

**int** len = 0;

**while**(-1 != (len = bais.read(buff))){

os.write(buff,0,len);

}

os.flush();

os.close();

}

测试：

**public** **static** **void** main(String[] args) **throws** IOException {

File dest = **new** File("e:\\aaa.txt");

File dest2 = **new** File("e:\\bbb.txt");

**byte**[] bs = **new** **byte**[] {22,23,22,44,33,22,-3,-43};

**char**[] cs = **new** **char**[] {'a','g','u','i','r','e','w','q','x','w'};

*copyByteArrayToFile*(bs,dest);

*copyCharArrayToFile*(cs,dest2);

}