RandomAccessFile流

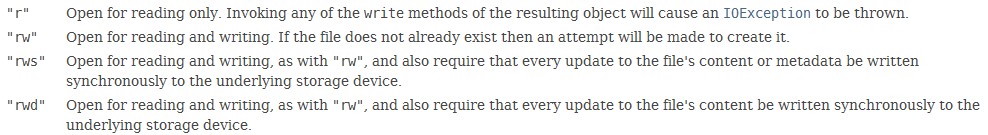
1. **RandomAccessFile流直接继承于Object类，不是常见的普通的IO流，**不是**普通IO体系**的子类；也是一种处理流。
2. **RandomAccessFile流的特点：**
3. 支持对**随机访问文件**的读和写，**既能读又能写**；随机，文件（包含文本文件和媒体文件）；
4. 该对象内部维护了一个**byte数组**，并通过指针可以操作数组中的元素；
5. 可以通过**getFilePoint方法**获取指针的位置，且可以通**过seek方法设置指针的位置**，通过指定指针的位置，就可以实现**随机读取**。 随机性就是由**seek来**完成的。 可以直接设置指针位置，然后写入数据，就可以**修改原来的数据**。
6. 其实**该对象**就是将**字节输入流和输出流**进行了封装；
7. **操作源和操作目的只能是文件File**。（从构造方法看出来）
8. **RandomAccessFile流：在生成一个随机访问对象时，如果文件不存在，就会创建一个文件;如果文件已经存在，就不会再次创建**，原来的数据不改变，如果没有通过seek方法设置指针，则就会默认从0开始写入，写入的位置换成新的数据，但是未重新写入的位置数据不变。这是与一般的IO输出流不同的地方。一般的输出流无论存在不存在都会**直接创建一个新的文件**，内容都会为空。
9. **RandomAccessFile流的构造方法：**
10. **接收文件File对象和模式mode**



1. **接收文件路径字符串和模式mode**



注：mode只有四种**：r、rw、rws、rwd**



1. **RandomAccessFile流的成员方法：**
2. **length方法**

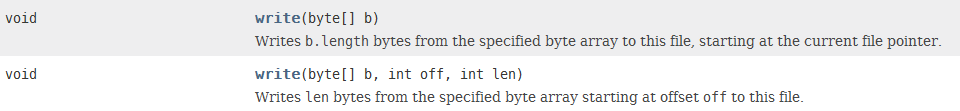


1. **setLength方法：**



1. **write**
2. **write数组：**





1. **writeXxx基本数据类型：**

**还有writeBytes、writeChars**

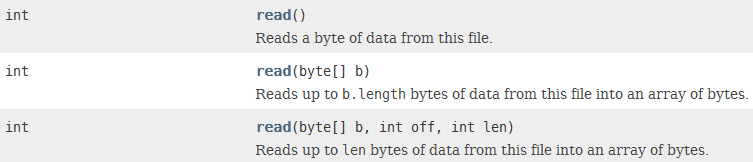




1. **writeUTF**



1. **read**
2. **传统的read方法**



1. **readXxx()方法：基本数据类型、无符号字节**



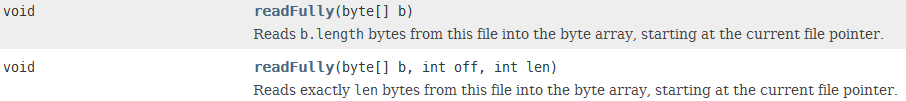
1. **readLine方法**



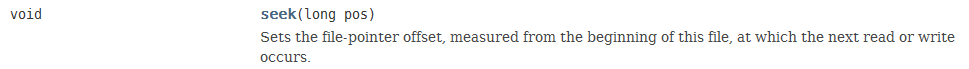
1. **readUTF（）方法：**



1. **readFully方法：**



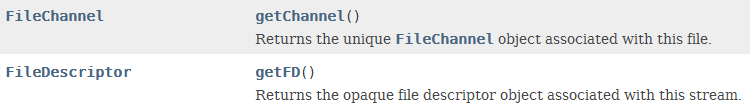
1. **seek（long pos）：**



1. **getFilePointer（）方法：返回long值。**



1. **getChannel()和getFD()方法**:



1. skipBytes



1. **RandomAccessFile流**的简单读写操作：

RandomAccessFile raf = new RandomAccessFile("e:\\raf.txt","rw");

raf.write("赵子龙".getBytes()); raf.writeInt(87); raf.writeInt(85);

raf.close();

RandomAccessFile raf1 = new RandomAccessFile("e:\\raf.txt","rw");

byte[] buf = new byte[6];

raf1.read(buf); String name = new String(buf);

int a = raf1.readInt(); int a2 = raf1.readInt();

System.out.println(name+" "+a+" "+a2);

raf1.close();

1. **随机写入和随机读取**：

**随机访问文件的操作的文件就像一个数组一样，具有索引，可以通过seek方法指定索引位置，从而完成随机读取和随机写入。**

* 1. **随机写入：**

**raf2.seek(14);//先指定指针位置，然后写入数据**

**raf2.write("李自成".getBytes());**

* 1. **随机读取：**

**raf.seek(14);**

**byte[] buf = new byte[6];**

**raf1.read(buf);**

**String name = new String(buf);**

**可以利用getFilePointer方法获取当前的指针值。**

**System.out.println(raf.getFilePointer());**

1. **随机写入和随机读取的应用：**

**可以实现分段读写，同时不同的流向文件的不同段写入数据，可用于多线程中。**