

# Java IO流详解（三）——RandomAccessFile



Ruheng 关注

2 2017.03.14 09:42:57 字数 1,018 阅读 33,001

本篇文章主要介绍RandomAccessFile，该类是IO流体系中功能最丰富的文件内容访问类，既可以读取文件内容，也可以向文件输出数据。

## 一、RandomAccessFile简介

RandomAccessFile既可以读取文件内容，也可以向文件输出数据。同时，RandomAccessFile支持“随机访问”的方式，程序可以直接跳转到文件的任意地方来读写数据。

由于RandomAccessFile可以自由访问文件的任意位置，所以如果需要访问文件的部分内容，而不是把文件从头读到尾，使用RandomAccessFile将是更好的选择。

与OutputStream、Writer等输出流不同的是，RandomAccessFile允许自定义文件记录指针，RandomAccessFile可以不从开始的地方开始输出，因此RandomAccessFile可以向已存在的文件后追加内容。如果程序需要向已存在的文件后追加内容，则应该使用RandomAccessFile。

RandomAccessFile的方法虽然多，但它有一个最大的局限，就是只能读写文件，不能读写其他IO节点。

RandomAccessFile的一个重要使用场景就是网络请求中的多线程下载及断点续传。

## 二、RandomAccessFile中的方法

### 1.RandomAccessFile的构造函数

RandomAccessFile类有两个构造函数，其实这两个构造函数基本相同，只不过是指定文件的形式不同——一个需要使用String参数来指定文件名，一个使用File参数来指定文件本身。除此之外，创建RandomAccessFile对象时还需要指定一个mode参数，该参数指定RandomAccessFile的访问模式，一共有4种模式。

**"r"**：以只读方式打开。调用结果对象的任何 write 方法都将导致抛出 IOException。

**"rw"**：打开以便读取和写入。

**"rws"**：打开以便读取和写入。相对于 "rw"，"rws" 还要求对“文件的内容”或“元数据”的每个更新都同步写入到基础存储设备。

**"rwd"**：打开以便读取和写入，相对于 "rw"，"rwd" 还要求对“文件的内容”的每个更新都同步写入到基础存储设备。

### 2.RandomAccessFile的重要方法

RandomAccessFile既可以读文件，也可以写文件，所以类似于InputStream的read()方法，以及类似于OutputStream的write()方法，RandomAccessFile都具备。除此之外，RandomAccessFile具备两个特有的方法，来支持其随机访问的特性。

RandomAccessFile对象包含了一个记录指针，用以标识当前读写处的位置，当程序新创建一个RandomAccessFile对象时，该对象的文件指针记录位于文件头（也就是0处），当读/写了n个

字节后，文件记录指针将会后移n个字节。除此之外，RandomAccessFile还可以自由移动该记录指针。下面就是RandomAccessFile具有的两个特殊方法，来操作记录指针，实现随机访问：

long getFilePointer( )：返回文件记录指针的当前位置

void seek(long pos )：将文件指针定位到pos位置

### 三、RandomAccessFile的使用


利用RandomAccessFile实现文件的多线程下载，即多线程下载一个文件时，将文件分成几块，每块用不同的线程进行下载。下面是一个利用多线程在写文件时的例子，其中预先分配文件所需要的空间，然后在所分配的空间中进行分块，然后写入：


```
1  /**
2   * 测试利用多线程进行文件的写操作
3   */
4  public class Test {
5
6      public static void main(String[] args) throws Exception {
7          // 预分配文件所占的磁盘空间，磁盘会创建一个指定大小的文件
8          RandomAccessFile raf = new RandomAccessFile("D://abc.txt", "rw");
9          raf.setLength(1024*1024); // 预分配 1M 的文件空间
10         raf.close();
11
12         // 所要写入的文件内容
13         String s1 = "第一个字符串";
14         String s2 = "第二个字符串";
15         String s3 = "第三个字符串";
16         String s4 = "第四个字符串";
17         String s5 = "第五个字符串";
18
19         // 利用多线程同时写入一个文件
20         new FileWriteThread(1024*1,s1.getBytes()).start(); // 从文件的1024字节之后开始写入
21         new FileWriteThread(1024*2,s2.getBytes()).start(); // 从文件的2048字节之后开始写入
22         new FileWriteThread(1024*3,s3.getBytes()).start(); // 从文件的3072字节之后开始写入
23         new FileWriteThread(1024*4,s4.getBytes()).start(); // 从文件的4096字节之后开始写入
24         new FileWriteThread(1024*5,s5.getBytes()).start(); // 从文件的5120字节之后开始写入
25     }
26
27     // 利用线程在文件的指定位置写入指定数据
28     static class FileWriteThread extends Thread{
29         private int skip;
30         private byte[] content;
31
32         public FileWriteThread(int skip,byte[] content){
33             this.skip = skip;
34             this.content = content;
35         }
36
37         public void run(){
38             RandomAccessFile raf = null;
39             try {
40                 raf = new RandomAccessFile("D://abc.txt", "rw");
41                 raf.seek(skip);
42                 raf.write(content);
43             } catch (FileNotFoundException e) {
44                 e.printStackTrace();
45             } catch (IOException e) {
46                 // TODO Auto-generated catch block
47                 e.printStackTrace();
48             } finally {
49                 try {
50                     raf.close();
51                 } catch (Exception e) {
52                     //
53                 }
54             }
55         }
56     }
57 }
```

当RandomAccessFile向指定文件中插入内容时，将会覆盖掉原有内容。如果不想覆盖掉，则需要将原有内容先读取出来，然后先把插入内容插入后再把原有内容追加到插入内容后。

 51人点赞 >



 java进阶学习





Ruheng

总资产1,568 共写了21.1W字 获得8,864个赞 共8,509个粉丝

关注