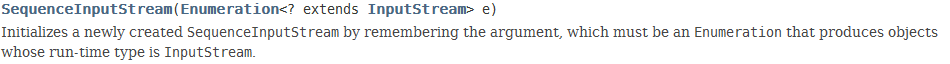
Java中的序列流SequenceInputStream

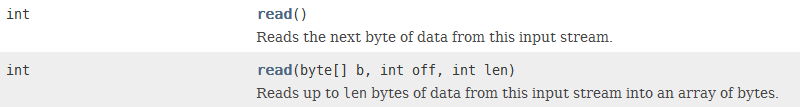
1. **SequenceInputStream**是一个**序列输入流**，存在java.io包中，是**java.io.InputStream类**的直接子类，可以实现**对多个流的合并，同时操作多个字节输入流（InputStream）对象**。（注意：没有SequenceOutputStream。）
2. **concatenation** [kənkætə'neɪʃn] [想启用英文朗读功能吗？请先安装flash插件！](http://www.adobe.com/shockwave/download/download.cgi?P1_Prod_Version=ShockwaveFlash)n. 串联，连结
3. **SequenceInputStream序列字节输入流的构造方法：如果只需要操作两个字节输入流，使用第一个即可，如果操作多个字节输入了，使用第二个构造方法。**
4. **操作两个字节输入流：**



1. **操作多个字节输入流：多个字节输入流利用枚举Enumeration存放。**



1. **SequenceInputStream成员方法：既然是一种字节输入流，所以其操作与普通的字节输入了一样，只不过是把多个流封装到了一起而已。**
2. **read方法：**



1. **close方法**：

**同时把多个流都关闭**。

1. **Enumeration本身**是一个接口（枚举接口），获取Enumeration接口对应对象的方法：
2. 利用**Vector向量**集合：Vector集合中有个elements方法，可以返回一个**Enumeration对象**。

**由于Vector的效率比较低，很少使用了，被ArrayList所取代。**

1. 利用集合工具类**Collections中的方法enumeration（Collection c），集合采用ArrayList即可**。



Collections中的enumeration的方法原理：

ArrayList<FileInputStream> al = new ArrayList<FileInputStream>();

Iterator<FileInputStream> it = al.iterator();

Enumeration<FileInputStream> en = new Enumeration<FileInputStream>() { @Override//Enumeration和Iterator的功能重复

public boolean hasMoreElements() {

return **it.hasNext()**; }

@Override

public FileInputStream nextElement() {

return **it.next()**; }

};//**匿名内部类**

1. **SequenceInputStream序列流的示例见 文件的分割与合并。**