

Java实现通用组合算法 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/645/2021_2022_Java_E5_AE_9E_E7_8E_B0_c104_645098.htm 编辑特别推荐: #0000ff>Java加载和实例化以及构造函数 #0000ff>关于计算Java程序运行时间 #0000ff>深入理解Java加载类的机制 现在有这样的需求：存在一个类似{31311133,33113330}这样的集合，经过8取5组合，其他位置用非字母数字字符替代，比如使用*号，得到类似{3***1133,***13330,... ..}这样的集合；还要求对于{3***1133,***13330}这样的集合，再次经过5取3组合，其他位置用非字母数字字符替代，比如使用*号，得到类似{*****133,*****330,3***1*3*,... ..}这样的集合。对于这样的要求，实现的思路如下：首先，主要思想是基于信息编码原理，通过扫描字符串，将10组合变为01组合。其次，对于每个数字字符串，设置一个单线程，在单线程类中设置一个List用来存放待处理数字字符串（可能含有*号，或者不含有）中每个数字的（而非*号）索引位置值；再次，设置BitSet来标志每个位置是否被*号替换得到新的组合字符串。最后，在扫描原始待处理数字字符串的过程中，根据设置的字符列表List中索引，来操作BitSet，对于每一个BitSet得到一个新的组合。使用Java语言实现如下：

```
package org.shirdrn. import
java.util.ArrayList. import java.util.BitSet. import java.util.Collection.
import java.util.Collections. import java.util.HashSet. import
java.util.Iterator. import java.util.List. /** * 通用组合拆分类(基于
单线程) * * 可以完成两种功能： * * 第一，可以将完全数字串
拆分成为含有*号的字符串。 * 例如：输入集
```

合{31311133,33113330}，Splitter类会遍历该集合，对每个字符串，创建一个SplitterThread * 线程来处理，如果是2取1组合，即starCount=8-2=6，经过线程处理得到类似*****33,*****1*3等结果 * * 第二，根据从带有*号的字符串经过拆分过滤后得到的字符串集合，对其中每一个字符串进行组合 * 例如：输入集合5取1组合字符串集合{3***1133,***113330} * *

CommonSplitter类会遍历该集合，对每个带有*号的字符串，创建一个SplitterThread * 线程来处理，如果是2串1组合，即starCount=8-3-2=3，经过线程处理得到类

似*****33,*****1*3等结果 * * @author 时延军 * */ public class CommonSplitter { private int starCount. private boolean duplicate. private Collection filteredContainer. public Collection getFilteredContainer() { return filteredContainer. } /** * 构造一个Spilitter实例 * * @param container 输入的待处理字符串集合 * @param starCount 如果对于长度为N的数字字符串，进行M组合（即N取M），则starCount=N-M * @param duplicate 是否去重 */ public CommonSplitter(Collection container, int starCount, boolean duplicate) { this.duplicate = duplicate. this.starCount = starCount. if(this.duplicate) { // 根据指定是否去重的选择，选择创建容器 filteredContainer = Collections.synchronizedSet(new HashSet()). } else { filteredContainer = Collections.synchronizedList(new ArrayList()). } Iterator it = container.iterator(). while(it.hasNext()) { new Thread(new SplitterThread(it.next().trim())).start(). } try { Thread.sleep(50). } catch (InterruptedException e) { e.printStackTrace(). } } 100Test 下载频道开通，各类考试题目直接下载。详细请访问

