

Java库谜题60：一行的方法 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/138/2021_2022_Java_E5_BA_93_E8_B0_9C_c97_138639.htm 现在该轮到写一些代码了。

下面的谜题每一个都可以用一个方法来解决，这些方法的方法体都只包含一行代码。各就各位，预备，编码！

A．编写一个方法，它接受一个包含元素的List，并返回一个新的List，它以相同的顺序包含相同的元素，只不过它把第二次以及后续出现的重复元素都剔除了。例如，如果你传递了一个包含 "spam"，"sausage"，"spam"，"spam"，"bacon"，"spam"，"tomato" 和 "spam" 的列表，那么你将得到一个包含 "spam"，"sausage"，"bacon"，"tomato" 的新列表。

B．编写一个方法，它接受一个由0个或多个由逗号分隔的标志所组成的字符串，并返回一个表示这些标志的字符串数组，数组中的元素的顺序与这些标志在输入字符串中出现的顺序相同。每一个逗号后面都可能会跟随0个或多个空格字符，这个方法忽略它们。例如，如果你传递的字符串是 "fear, surprise, ruthless efficiency, an almost fanatical devotion to the Pope, nice red uniforms"，那么你将得到的将是一个包含5个元素的字符串数组，这些元素是 "fear"，"surprise"，"ruthless efficiency"，"an almost fanatical devotion to the Pope" 和 "nice red uniform"。

C．假设你有一个多维数组，出于调试的目的，你想打印它。你不知道这个数组有多少级，以及在数组的每一级中所存储的对象的类型。编写一个方法，它可以向你显示出在每一级上的所有元素。

D．编写一个方法，它接受两个int数值，并在第一个数值与第二个数值以二进制

补码形式进行比较，具有更多的位被置位时，返回true。A . 众所周知，你可以通过把集合（collection）中的元素置于一个Set中将集合中的所有重复元素都消除掉。在本谜题中，你还被要求要保持最初的集合中的元素顺序。幸运的是，有一种Set的实现可以维护其元素被插入的顺序，它提供的导入性能接近HashMap。它就是LinkedHashSet，它是在1.4版本的JDK中被添加到Java平台中的。在内部，它是用一个链接列表来处理的，从而被实现为一个散列表。它还有一个映射表版本可供你使用，以定制缓存。一旦你了解了LinkedHashSet，本谜题就很容易解决了。剩下唯一的关键就是你被要求要返回一个List，因此你必须用LinkedHashSet的内容来初始化一个List。把它们放到一块，就形成了下面的解决方案：

```
static List withoutDuplicates(List original) { return new ArrayList(new LinkedHashSet(original)). }
```

B . 在将字符串解析成标志时，许多程序员都立刻想到了使用StringTokenizer。这是最不幸的事情，自1.4版本开始，由于正则表达式被添加到了Java平台中（java.util.regex），StringTokenizer开始变得过时了。如果你试图通过StringTokenizer来解决本谜题，那么你很快就会意识到它不是非常适合。通过使用正则表达式，它就是小菜一碟。为了在一行代码中解决本谜题，我们要使用很方便的方法String.split，它接受一个描述标志分界符的正则表达式作为参数。如果你以前从来没有使用过正则表达式，那么它们看起来会显得有一点神秘，但是它们惊人地强大，值得我们好好学习一下：

```
static String[] parse(String string) { return string.split(",\\S*"); }
```

C . 这是一个讲究技巧的问题。你甚至不必去编写一个方法。这个方法在5.0或之后的版本中已经提供

了，它就是Arrays.deepToString。如果你传递给它一个对象引用的数组，它将返回一个精密的字符串表示。它可以处理嵌套数组，甚至可以处理循环引用，即一个数组元素直接或间接地引用了其嵌套外层的数组。事实上，5.0版本中的Arrays类提供了一整套的toString、equals和hashCode方法，使你能够打印、比较或散列任何原始类型数组或对象引用数组的内容。

D. 为了在一行代码中解决该谜题，你需要了解在5.0版本中添加到Java平台中的一整套位操作方法。整数类型的包装器类（Integer、Long、Short、Byte和Char）现在支持通用的位处理操作，包括highestOneBit、lowestOneBit、numberOfLeadingZeros、numberOfTrailingZeros、bitCount、rotateLeft、rotateRight、reverse、signum和reverseBytes。在本例中，你需要的是Integer.bitCount，它返回的是一个int数值中被置位的位数：

```
static Boolean hasMoreBitsSet(int i, int j) { return (Integer.bitCount(i) > Integer.bitCount(j)). }
```

总之，Java平台的每一个主版本都在其类库中隐藏了一些宝藏。本谜题的所有4个部分都依赖于这样的宝藏。每当该平台发布一个新版本时，你都应该研究就一下新特性和提高（new features and enhancements）页面，这样你就不会遗漏掉新版本提供的任何惊喜[Features-1.4, Features-5.0]。了解类库中有些什么可以节省你大量的时间和精力，并且可以提高你的程序的速度和质量。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com