

java基础：遍历m取n的所有组合 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_java_E5_9F_BA_E7_A1_80_c104_145504.htm

```
* 求m取n的所有组合。 * m个数分别为0,1,2...m-1. * 算法简述： * 二个组合，若仅有元素顺序不同，视其为同一个组合。 * 左位系低位，右位系高位。 * 按自然的取法取第一个组合(各数位分别是:0,1,2...n-1)，以后的所有组合都经上一个组合变化而来: * 从右至左，找到有增量空间的位,将其加1,使高于该位的所有位，均比其左邻位大1，从而形成新的组合。 * 若所有位均无增量空间，说明所有组合均已被遍历。 * 使用该方法所生成的组合数中：对任意组合int[] c,下标小的数必定小于下标大的数. * */ public class Combination { int n, m. int[] pre;//previous combination. public Combination(int n, int m) { this.n = n. this.m = m. } /** * 取下一个组合。可避免一次性返回所有的组合(数量巨大，浪费资源)。 * if return null,所有组合均已取完。 */ public int[] next() { if (pre == null) { //取第一个组合，以后的所有组合都经上一个组合变化而来。 pre = new int[n]. for (int i = 0. i < n. i++) pre[i] = i. } int[] ret = new int[n]. System.arraycopy(pre, 0, ret, 0, n). return ret. } int ni = n - 1, maxNi = m - 1. while (pre[ni] + 1 > maxNi) { //从右至左，找到有增量空间的位。 ni--. maxNi--. if (ni < 0) return null. //若未找到，说明了所有的组合均已取完。 } pre[ni] += 1. while (pre[ni] > pre[ni - 1]) pre[ni] = pre[ni - 1]. } int[] ret = new int[n]. System.arraycopy(pre, 0, ret, 0, n). return ret. } }
```

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