JAVA认证培训辅导：随机整数的生成 PDF转换可能丢失图片或格式，建议阅读原文 https://www.100test.com/kao_ti2020/254/2021_2022_JAVA_E8_AE_A4_E8_AF_81_c67_254810.htm 使用Java 2 SDK基础类库产生随机数的方法很多。但是如果你跟不上这些类库的更新脚步，你有可能正在使用的是一种低效的随机数生成机制，更糟糕的是：你有可能得到的不是均匀分布的随机数。本文将向你展示一种较为可靠的随机数生成方法，同时与其他方法进行比较。 自从JDK最初版本发布起，我们就可以使用java.util.Random类产生随机数了。在JDK1.2中，Random类有了一个名为nextInt()的方法： public int nextInt(int n) 给定一个参数n，nextInt(n)将返回一个大于等于0小于n的随机数，即：0 你所要做的就是先声明一个Random的对象，在调用其nextInt(n)函数以返回随机值。 这里有个示例，下面的代码段将生成很多随机数并输出它们的平均值：以下是引用片段： int count = 1000000. int range = Integer.MAX_VALUE / 3 * 2. double sum = 0. Random rand = new Random(). for (int i=0. i sum = rand.nextInt(range). } System.out.println(sum/count). 执行了1000000次循环之后，得到的平均值基本上就处于随机数范围的中点(midpoint)。 到目前为止，事情还并不复杂，但是我们会问为什么要使用nextInt(n)?考虑一下的随机数生成方法：(1)使用老的方法nextInt()，没有制定数值范围 (2)用Math.abs()静态函数得到(1)中产生值的绝对值 (3)对(2)的结果进行取模运算(%)，得到期望范围类的值 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