JAVA基础: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_ti2020/145/2021_2022_JAVA_E5_9F_BA_E7_A1_80_c104_145319.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 amp.lt. n。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：int count = 1000000.int range = Integer.MAX_VALUE / 3 * 2.double sum = 0.Random rand = new Random().for (int i=0. iamp.lt.count. i ) {sum = Math.abs(rand.nextInt()) % range.}System.out.println(sum/count).不难发现，每次循环都多出了几步运算。事实上，这种随机数生成的方法存在着以下三个问题：首先，nextInt()返回的值是趋于均匀分布在Integer.MIN_VALUE 和 Integer.MAX_VALUE之间的。如果你取Integer.MIN_VALUE的绝对值，得到的仍然不是一个正数。事实上，Math.abs(Integer.MIN_VALUE)等于Integer.MIN_VALUE。因此，存在着这样一种情况（虽然很少见）：rand.nextInt()=Integer.MIN_VALUE，经过取绝对

值Math.abs(rand.nextInt())之后，得到是一个负数。这种几率为 1/(2^31)，在我们的测试中不太可能发生循环次数只有1000000次。 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