

JC2002 Java 编程 - 实践 5（第 6 天）

本实践的目的是练习在 Java 程序中使用递归和异常。

1. **科拉茨猜想**是一个著名的未经证实的数学猜想，其表述如下：如果 n 是偶数，用 n 除以 2 得出下一个数；如果 n 是奇数，用 n 乘以 3 再加上 1（即 $3n+1$ ）得出下一个数。经过一定步数后，数字序列总会到达 1。请编写一个 Java 程序，使用递归法打印从参数 n 开始的科拉茨数列。*提示：*您可以从今天讲座中的斐波那契示例开始。用不同的 n 初始值测试你的程序。
2. 科拉茨猜想只有在初始 n 为正整数时才有效。扩展你的程序，以便在 n 无效时抛出 `IllegalArgumentException`。此外，使用 `try ... catch` 结构来处理异常。例如，如果捕获到异常，可以打印文本 "请使用有效的 n "。
3. 修改前面任务中的程序，使其不打印出序列，而是计算到达终点所需的步数，并在递归结束后打印出计数，例如，"对于初始 $n = 21$ ，用了 7 步到达 1"。用不同的 n 初始值（包括大值和小值）测试程序。
4. 回到上一个实际操作中的计算器程序。你可能已经注意到，当 `Scanner` 对象的 `nextInt()` 方法读取用户输入时，如果输入了无效数据，例如字母而不是数字，程序的执行就会随着 `InputMismatchException` 的抛出而结束。在 `RunApp` 类中添加 `try...catch` 异常处理程序来处理异常（例如，可以在 `catch` 块中打印 "无效输入" 信息）。

请注意，如果要在处理异常后继续循环，则需要在读取用户输入的循环内设置 `try` 块。在这种情况下，还需要在 `catch` 代码块内使用 `nextLine()` 方法清除扫描缓冲区，否则会产生新的异常。

5. 在计算器类中添加第四个方法 `divide()`，如果用户选择除法运算 `/`，该方法将被调用。在 `RunApp` 的方法 `main()` 中添加另一个 `catch` 块，以处理因除数为 0 而抛出的 **算术异常**。例如，您可以打印文本 "除以零"。
6. 由于我们的计算器使用整数操作，因此不会返回小数。定义自定义异常 `NotDivisibleException`，当除法运算的模数不为零时，该异常会在方法 `divide()` 中抛出。在 `RunApp` 的方法 `main()` 中再添加一个 `catch` 块，处理因第一个整数不能被第二个整数整除而抛出的 `NotDivisibleException`。例如，可以打印文本 "不可整除"。