**复审过程记录**

本次复审主要是对初审后的代码进行再次的检查，寻找代码中存在的问题，本次复审共发现并进行了三项修改：

1. 由于本代码使用了栈的知识，在使软件更加贴合用户需求时添加了一些代码段进行完善，这时代码就产生了出现空栈的错误。在编写代码时，为了避免产生a\*0、a/0、a-0、a+0、a\*1、a/1等情况的产生，专门编写了代码段对这些情况进行特殊处理，但是在添加此代码段后运行就会产生异常报错，为此我们将整体代码逻辑再次进行了梳理，并没有发现逻辑错误，即我们无法从逻辑上发现问题。

根据报错的类型，我们找到了调用栈的函数，初步猜测是tempEqu中出现的异常，我们对方法进行了再次的思考，生成一个符合数据要求且不含括号的中间式，此段代码进行了上述修改就出现了这次异常，那么试着去看看错误是怎么出现的，我们开始设置断点去看看数据是怎么出现的，刚开始一切照常运行，但是在循环的过程中还是出现了异常，我们将生成的式子打印出来，发现有的式子是正常的，而有的式子就只有一个数，例如单个0出现的概率最多，其次还有其他的随机数。

我们将代码段进行注释，发现代码可以无异常的运行，那么异常就是出现在新添加的代码段处，为了实现我们想达到的要求，我们试着在生成数据处进行修改，因为生成随机数较为简单，我们将最小值设置为1，那么就不会出现有0参与式子的情况，但是代码的编写还是比较辛苦的，我们就试试能不能不将添加的代码段删去，我们把代码顺序进行了调换，代码就不知为何的正常运行了，我们还是不知道异常出现的真正原因，但是代码却可以正常运行，此异常还需要不断去寻找真正出现的原因以避免此异常的再次出现，最终具体的解决方案将在复审后进行选择。

1. 在加括号的代码段进行了修改。我们在对上一个异常处理的过程中思考是否是加括号的代码段出现了异常意外的发现此代码段有重复代码，此代码段中分情况进行了讨论，如只有一个运算符就不添加括号，有两个运算符该怎么进行，有三个运算符该怎么进行，就发现两个和三个运算符所进行的操作基本相同，我们就将此类情况归为一种情况进行处理，就把此处的相同代码提取出来，然后进行了一些修改。

除此之外，我们根据生成算式发现算式中乘除出现的概率较小，因为我们添加括号的原理是基于乘除运算符进行的，那么就会导致添加的括号较少，我们对乘除号进行了思考，发现是因为包含乘除号的运算在判断是否重复的过程中排除掉了，我们对重复的判定是比较成熟的无需进行修改，我们就将修改的地方放在了添加括号处，我们想到在减号后添加括号，那么后面的符号就要进行变号，这样生成的式子和之前不加括号的式子就不是重复的了，因为我们对添加括号的判定是括号中必须包含加减号才可以添加括号，那么先识别减号再在减号后添加括号就是可行的，我们就试着将此类添加括号的形式和之前的添加括号的代码段并行，先识别减号，符号条件的地方一定是可以添加括号的，然后在根据随机数判断是否在此处添加括号，可以保证括号的出现不会过多就是完全随机添加括号了，那么就可以将这个问题解决。

1. 将重复代码进行了整合和简化。本次检查共发现了两处代码冗长，一处是判断运算符个数再进行操作，一处是添加括号的地方也存在一定的可简化空间

此次共发现这三类问题，这些问题还在解决过程中，目前还未发现其他的问题。希望在修改的代码的过程中能发现其他问题，就可以将代码进行不断的完善，但是没有问题更好（doge）。

**复审表**

综合整个代码复审过程，填写了如下代码复审表：

|  |  |
| --- | --- |
| Code Review Checklist | |
| General | |
| Does the code work? Does it perform its intended function, the logic is correct etc. | 代码正常运行，在某些角度来看是符合预期功能的，还缺少一些功能的完善，逻辑正确没有错误，符合规定，但陾需要进行不断的完善； |
| Is all the code easily understood? | 代码尽可能的使用简单的形式，但仍存在一些复杂的代码，在某些难以理解的代码处都写下了注释便于理解； |
| Does it conform to your agreed coding conventions? These will usually cover location of braces, variable and function names, line length, indentations, formatting, and comments. | 所编代码都符合根据现有的Java代码规范制定的小组守则（见表后注释），，严格遵守共同的编码规范； |
| Is there any redundant or duplicate code? | 在编码过程中不断精简代码，已经将重复和冗长的代码进行了一定程度完善和更改，还在想办法进行完善； |
| Is the code as modular as possible? | 是的，代码尽可能的模块化，现有代码已经按模块化进行了分配处理； |
| Is there any commented out code? | 存在注释的代码，这些代码有更好更完善的代码可以替代就注释掉了，还有一些是为了测试功能注释掉过； |
| Do loops have a set length and correct termination conditions? | 循环是符合题目要求的，可以根据用户需要选择循环的长度和正确的终止条件； |
| Security | |
| Are all data inputs checked (for the correct type, length, format, and range) and encoded? | 检查了所有的数据输入(以获得正确的类型、长度、格式和范围)并对其进行编码，确保运行的正确进行； |
| Where third-party utilities are used, are returning errors being caught? | 没有捕获到返回错误； |
| Are output values checked and encoded? | 检查了输出值是否符合题目要求，并尽可能的让输出值变得正确符合题目要求； |
| Are invalid parameter values handled? | 无效的参数值进行了处理，目前代码中不存在无效的参数值； |
| Documentation | |
| Do comments exist and describe the intent of the code? | 注释完全按照编码规范解释了代码的功能理解、意图和参数返回值等； |
| Are all functions commented? | 是的，函数都按照编码规范进行了注释，但由于时间仓促，还有部分代码未编写注释； |
| Is any unusual behavior or edge-case handling described? | 对可能存在的异常情况进行了描述和处理，进一步的修改将在复审中进行； |
| Is there any incomplete code? If so, should it be removed or flagged with a suitable marker like ‘TODO’? | 现有代码都是完整的，如果存在不完整的代码，就会采用“TODO”标识出来； |
| Testing | |
| Is the code testable? i.e. don’t add too many or hide dependencies, unable to initialize objects, test frameworks can use methods etc. | 代码可按照正常流程进行测试； |
| Do tests exist and are they comprehensive? i.e. has at least your agreed on code coverage. | 代码测试全面，覆盖了我们认为需要进行测试的地方； |
| Do unit tests actually test that the code is performing the intended functionality? | 不止如此，还测试了代码是否存在漏洞，并且是否能够按照预期进行运行； |
| Are arrays checked for ‘out-of-bound’ errors? | 检查了是否存在越界行为，并不存在此类行为； |
| Could any test code be replaced with the use of an existing API? | 测试代码尽量精简，并全部调用现有API判断是否通过。 |

注：小组编码规范守则：<https://www.cnblogs.com/Derek-cpp/p/15334278.html>