

Java HW2 文档

火柴棒游戏程序

目录

1	设计引言.....	2
1.1	设计要求.....	2
1.2	交互界面.....	2
1.3	设计环境.....	2
2	程序说明.....	2
2.1	Homework2 类.....	2
2.2	Quiz 类.....	3
3	运行情况.....	5
3.1	运行说明.....	5
3.2	运行测试.....	5
4	总结与心得.....	6

1 设计引言

1.1 设计要求

本次作业要求实现一个简单的随机生成的火柴棒游戏，需要支持用户自定义出现数字的最大位数、等式左边数字的数量（2 或 3）、题目类型（移动、移除或添加）和涉及的火柴棒数量，由程序随机生成火柴棒游戏并检验用户输入是否符合预设答案。

1.2 交互界面

程序采用命令行输入输出即可，火柴棒游戏题目可直接用数字表示。

1.3 设计环境

IDE: NetBeans IDE 8.0.2

Java Development Kit: Java SE 8 (v1.8.0_341)

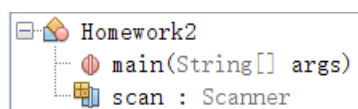
2 程序说明

本程序由一个包（Homework2）内含的两个 java 文件组成，每个 java 文件对应一个类。

2.1 Homework2 类

该类的实现包含在文件 Homework2.java 中。

类的成员类及方法列表如下：



2.1.1 类公有成员

Scanner 类对象 scan。该对象使用 `System.in` 构造，用于从标准输入获取用户响应。

2.1.2 类方法

公有的 main 方法。这是整个程序的入口，实现大体的运行流程。

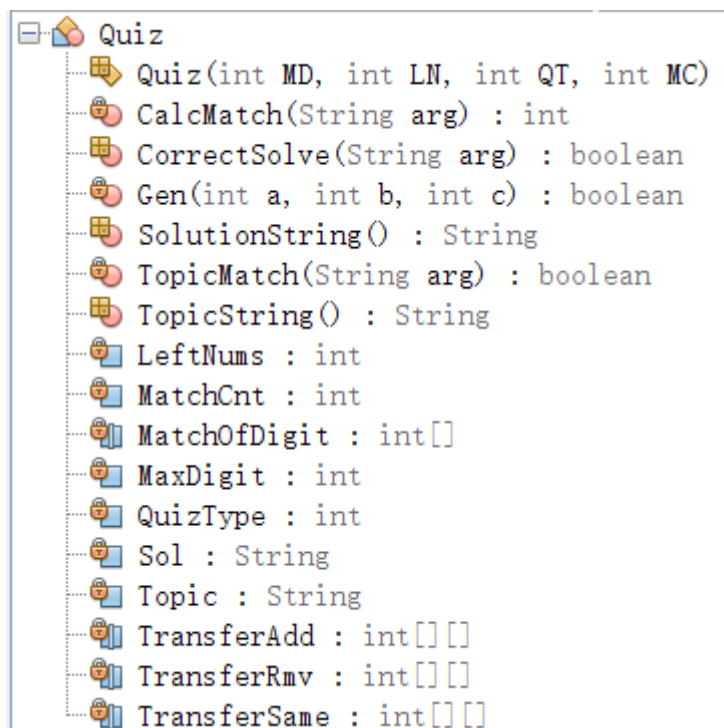
整个 main 方法实现的步骤为：读取用户输入→使用用户输入的参数构建一个 Quiz 类对象→检验 Quiz 类对象生成题目的有效性→给出题目→读取用户输入并对输入进行检验。

详细实现可参考源文件内注释。

2.2 Quiz 类

该类的实现包含在文件 Quiz.java 中。

类的成员类及方法列表如下：



2.2.1 类私有成员

int 类对象 MaxDigit, LeftNums, QuizType 和 MatchCnt。 分别代表构造题目需要的参数（即 Homework2.main() 中用户输入的数字：数字的最大位数，等式左边的数字数量，题目类型和涉及的火柴数量），除 MaxDigit 是由最大位数经过乘算转换成了最大数码外，其余直接从构造函数参数中复制而来。

int 类对象数组 MatchOfDigit。 该数组用 **final** 修饰。MatchOfDigit[i] 代表数字 i 需要用到的火柴数量，用以在之后验证题目生成的正确性。

int 类对象二维数组 TransferAdd, TransferRmv 和 TransferSame。 这些数组都用 **final** 修饰。TransferAdd[i][j] 代表从数码 i 到 j 需要添加火柴棒的数量。同理，TransferRmv[i][j] 代表需要的移除数量，实际上是 TransferAdd 的转置。如果无法通过增添/移除转换，则让该值为-1。TransferSame[i][j] 表示 i 和 j 有多少根火柴位置是相同的，用以确定移动类型题目中涉及火柴棒的数量。

String 类对象 Solution 和 Topic。 分别代表生成的正确答案和给用户展示的错误等式。

2.2.2 类方法

Quiz(int, int, int, int)构造函数。这个构造函数用来按用户给出的条件生成一个题目和对应的正确答案。

构造函数将 Homework.main() 中读取的四个参数复制到 Quiz 类的私有对象中，然后利用 random() 函数在该限制下随机生成一个正确等式，作为题目的正确答案。对这个正确答案，调用 gen() 来随机生成一个满足设定条件的题目。若不能生成一个题目，重试。如果重复次数达到上限（目前设置为 10 次）仍不能随机生成一个题目，则认为凭借设定的条件无法生成题目。

gen(int, int, int)方法。这个方法根据正确等式左边的数字随机生成题目。

因为枚举所有可能的数字生成题目的时间花销太大，我在实现中假定只变更一个数字或两个数字即可得到一个题目。假定变更一个数字，进行枚举，调用 TopicMatch() 方法进行检查，如果存在能满足设定条件的题目，则使用 random() 随机选择任一，返回 true；否则改为变更两个，同上步骤。如果依然找不到，返回 false，重新生成正确答案。

这个方法利用了 java.util.ArrayList 暂存可能的题目。

通过以上的生成步骤，在最大三位数情况下，程序得到一个题目最大需要运行六亿左右次，基本上能做到秒出结果（况且，根据随机情况，很可能不需要这么大的运行次数）。

TopicMatch(String)方法。这个方法用以验证一个生成的题目是否符合条件。

先检查和正确答案字符串的 length() 相同。之后检查是否数字位和符号位一一对应。若以上检查都通过，则调用 CalcMatch(String) 方法进行下一步检查（详见下一段）。检查都通过后返回 true；否则返回 false。

CalcMatch(String)方法。这个方法检查题目是否能通过移动、移除或添加得到正确答案。

分情况讨论，运用预先设定的 TransferAdd, TransferRmv 和 TransferSame 三个数组即可。添加和移除情况较为简单，只需从数组得到每个数字变更的火柴数量，加和与 MatchCnt 比较，或者遇到不可能变更的情况(-1)直接返回 false。移动情况需要对每个数字所需添加和移除的数量都进行统计，如相同则与

MatchCnt 比较，否则返回 false。检查通过返回 true。

CorrectSolve(String)方法。用以检查用户输入是否正确。

就是直接利用字符串的 `equals()` 方法比较一下正确答案，没什么技术含量。

TopicString()方法。用以返回私有成员 Topic（即题目）。

SolutionString()方法。用以返回私有成员 Solution（即正确答案）。

3 运行情况

3.1 运行说明

使用 javac 编译两个 java 文件，把 Homework2 作为入口直接运行即可。或者将两个 java 文件放进任意 IDE 的项目中，构建并运行即可。

环境可参考 1.3 设计环境。

3.2 运行测试

以下表格中，用红色字代表用户输入和对应的情况，黑色字代表输出和对应的情况。

测试编号	输入/输出情况	说明
1	2 2 1 1 72+10=91 72+16=91 Wrong answer, try another solution? 72+19=91 Correct!	2 位数，移动 1 根火柴 随机生成题目 错误答案 被提示错误 结果正确 输出提示并终止
2	2 2 1 1 21+17=92 21+71=92	2 位数，移动 1 根火柴 随机题目与测试 1 不同 直接回车 给出正确答案并终止
3	2 3 2 2 0+8+45=50 0+5+45=50 Correct!	2 位数，移除 2 根火柴 随机生成题目 结果正确 输出提示并终止

4	2 2 1 10000 Cannot generate a quiz!	不可能生成的题目 提示并终止
---	--	-------------------

4 总结与心得

Java 的一些工具确实很好用，比如 ArrayList。给我编程带来了不少方便。但同时，我依然为了字符串比较的==和 equals()而调试了半天……只能说还是不太熟悉 java 的这种语法模式，还是得多写点东西。

本次作业中，我一开始为如何“随机”生成火柴棒游戏的题目思考了很多，然而要么是算法太过复杂，要么是时间开销太大。最后依然还是用了枚举后检查再随机的方法，以保证“较好”的随机性。

应该说完全的随机性与程序的运行效率是不太可以兼得的。我之后采用了假定只变更一个数字或两个数字的方法，得到了一个姑且满意的结果。

这个火柴棒游戏要是只是命令行的话是真的不太好玩……没法直观看到火柴棒如何排列有点太伤脑筋了。不过现在做到这种程度（交个作业），算法上和运行上都没什么重大瑕疵的情况，姑且也就可以了？也许后续还可以更新嘛。