数独项目生成部分设计文档

一．开发模型

由于项目较小，所以使用迭代式开发方法，先实现数独生成的任务，然后再实现数独求解的任务。

二．需求说明

实现一个sudoku.exe的命令行程序，程序可以:

·生成不重复的数独终局到sudoku.txt文件

·读取文件内的数独问题，并将结果输出到文件

a．生成终局

1. 在命令行中使用-c参数加数字N（1~1000000）控制生成数独终局的数量，例子如下：

sudoku.exe –c 20

2. 将生成的数独终局用一个名为sudoku.exe的文本文件保存起来，文件格式如下，数与数之间有空格隔开，终局之间空一行，行末无空格。

2 6 8 4 7 3 9 5 1

3 4 1 9 6 5 2 7 8

7 9 5 8 1 2 3 6 4

5 7 4 6 2 1 8 3 9

1 3 9 5 4 8 6 2 7

8 2 6 3 9 7 4 1 5

9 1 7 2 8 6 5 4 3

6 8 3 1 5 4 7 9 2

4 5 2 7 3 9 1 8 6

4 5 1 7 8 2 3 6 9

7 8 6 4 9 3 5 2 1

3 9 2 1 5 6 4 8 7

5 2 7 6 4 9 8 1 3

9 6 8 5 3 1 2 7 4

1 3 4 2 7 8 6 9 5

8 1 5 3 6 7 9 4 2

6 7 3 9 2 4 1 5 8

2 4 9 8 1 5 7 3 6

3. 程序在处理命令行参数时，不仅能处理格式正确的参数，还能够处理各种异常的情况，如：

Sudoku.exe –c abc

4. 在生成数独矩阵时，左上角的第一个数为：（学号后两位相加）% 9 + 1。我的学号后2位是89，则该数字为（8+9）% 9 + 1 = 9，那么生成的数独棋盘应如下（x表示满足数独规则的任意数字）：

9 x x x x x x x x

x x x x x x x x x

x x x x x x x x x

x x x x x x x x x

x x x x x x x x x

x x x x x x x x x

x x x x x x x x x

x x x x x x x x x

x x x x x x x x x

b．求解数独

1. 在命令行中使用-s参数加文件名的形式求解数独，并将结果输出至文件，

如：

sudoku.exe -s absolute\_path\_of\_puzzlefile

程序将从路径中读取数独题目，并将数独题目的一个可行解输出至sudoku.exe同目录的sudoku.txt中，要求与生成终局相同。

2. <puzzlefile>也为txt文档，格式与生成数独中的格式相同，0代表谜题中的空格。

3. 求解得到的sudoku.txt格式与生成数独中的sudoku.txt格式相同。

4. < puzzlefile >中数独题目个数N（1<=N<=1000000），保证文件中数独格式正确。

三．功能设计

本次设计暂不考虑用户界面的展示，只实现基本功能，即数独的生成与数独的求解。

四．系统架构设计,

第一个生成项目，我决定用一个类来实现，类名为EndProducer，意为终局生成，其中属性包含一个祖先矩阵以及种子数组，矩阵计划使用struct实现，因为struct中含有九个大小为9的数组分别代表1-9行，种子数组作用是使用1-8全排列（因为首位为9所以不换）索引对原始矩阵进行替换。方法有以下几种：

·行变换方法，分别有132,123,456,465,546,564,654,645,789,798,879,897,987,978等14种变换方式，因此写14个方法。

·索引替换方法。

·初始seed随机化方法，这个方法是为了避免每次开始的原始矩阵都一样，生成一个1-8之间的随机排列，用索引对祖先矩阵进行替换得到原始矩阵。

·对进行一次替换后的原始矩阵进行行变换，具体实现是：生成palaceVary1, palaceVary2, palaceVary3三个随机整数，1<=palaceVary1<=2, 1<= palaceVary2 <= 6, 1<= palaceVary3 <= 6，分别代表相应的变换方式。每次随机后会检查之前有没有相同的组合，若有也重新随机。

·名字暂定为主操作方法，对原始矩阵进行索引替换，之后使用行变换方法输出终局。Seed的全排列则调用next\_permutation()得到。

·查重方法

·交换行方法