

# 软件工程基础 - 个人项目

---

				8				
	2	7				3	4	
8	9		4		7		1	5
6			2		5			9
	8						3	
2			8		9			4
1	4		7		6		9	3
	6	5				2	7	
				9				

- [软件工程基础 - 个人项目](#)
  - [零、任务：](#)
  - [一、要求：](#)
  - [二、作业提交：](#)
  - [三、需求：](#)
    - [生成终局](#)
    - [求解数独](#)
    - [附加题：](#)
  - [四、测试须知](#)
  - [五、博客撰写要求：](#)
  - [六、评分规则](#)

## 零、任务：

---

实现一个能够生成数独终局并且能求解数独问题的控制台程序。

## 一、要求：

---

1. 阅读个人软件开发流程（PSP）的相关资料。
2. 可选的语言包括:C++, Java, Python。运行环境为64bit Windows 10。
3. 提交的代码要求经过代码质量分析工具的分析并消除所有的警告。如[Code Quality](#)

[Analysis](#)。

4. 完成项目的首个版本之后，请使用**性能分析工具**来找出代码中的性能瓶颈并进行改进。
5. 使用单元测试对项目进行测试，并使用插件查看测试分支覆盖率等指标；并写出至少10个测试用例确保你的程序能够正确处理各种情况。如[Studio Profiling Tools](#)。
6. 使用GitHub来管理源代码和测试用例，**代码有进展即签入GitHub**。签入记录不合理的项目会被抽查询问项目细节。
7. 按照要求发布博客，结合个人项目的实践经历，撰写解决项目的心路历程与收获。**博客与GitHub项目明显不符的作业将取消作业成绩**。

*注意：要求3、4、5在这3种语言中都需要实现，只是使用的工具不同。*

## 二、作业提交：

- 撰写一个博客，要求参见博客作业要求。
- 在个人博客上发布项目源代码（包含单元测试用例）的GitHub链接，将会在测试环境中检查程序的正确性。
- 正确的程序会再进行性能测试，根据性能的好坏进行评分；不正确的程序没有性能的分。

## 三、需求：

实现一个命令行程序，程序能：

1. 生成**不重复**的数独终局至文件
2. 读取文件内的数独问题，求解并将结果输出到文件

### 生成终局

1. 在命令行中使用-c参数加数字N ( $1 \leq N \leq 1000000$ )控制生成数独终局的数量，例如下述命令将生成20个数独终局至文件中：

```
sudoku.exe -c 20
```

2. 将生成的数独终局用一个文本文件（假设名字叫 sudoku.txt）的形式保存起来，每次生成的txt文件需要覆盖上次生成的txt文件，文件内的格式如下，数与数之间由空格分开，终局与终局之间空一行，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
9 5 8 3 6 7 1 2 4
2 3 7 4 5 1 9 6 8
1 4 6 9 2 8 3 5 7
6 1 2 8 7 4 5 9 3
5 7 3 6 1 9 4 8 2
4 8 9 2 3 5 6 7 1
7 2 4 5 9 3 8 1 6
8 9 1 7 4 6 2 3 5
3 6 5 1 8 2 7 4 9
.....
```

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。例如学生A学号后2位是80，则该数字为（8+0）% 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
x x x x x x x x x
```

## 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与sudoku.exe同目录的sudoku.txt中，要求与生成终局相同。

2. 格式如下，其中0代表空格，题目与题目之间空一行，行末无空格，最后一个数独题目后无空行：

```
9 0 8 0 6 0 1 2 4
2 3 7 4 5 1 9 6 8
1 4 6 0 2 0 3 5 7
0 1 2 0 7 0 5 9 3
0 7 3 0 1 0 4 8 2
4 8 0 0 0 5 6 0 1
7 0 4 5 9 0 8 1 6
8 9 0 7 4 6 2 0 0
3 0 5 0 8 0 7 0 9
9 0 0 8 0 0 4 0 0
.....
```

3. sudoku.txt的格式如下（与生成终局的要求相同）：

```
9 5 8 3 6 7 1 2 4
2 3 7 4 5 1 9 6 8
1 4 6 9 2 8 3 5 7
6 1 2 8 7 4 5 9 3
5 7 3 6 1 9 4 8 2
4 8 9 2 3 5 6 7 1
7 2 4 5 9 3 8 1 6
8 9 1 7 4 6 2 3 5
3 6 5 1 8 2 7 4 9
```

4. 数独题目个数N ( $1 \leq N \leq 1000000$ )，保证文件中数独格式正确。

## 附加题：

现在已经有有了一个数独游戏的生成器，如果让大家都能实际使用它，还需要一个简单的游戏界面。为数独游戏的生成器做一个GUI界面，并附上一个简单的使用说明。界面需实现下述功能，会按点给分：

- 生成任意数量的数独题目并将初始数独棋局依次显示。初始数独棋盘需要挖空，要求为：99棋盘上挖空不少于30个，不多于60个。每个33的小棋盘中挖空不少于2个。比如下面这就是一

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

个规范的棋局（2.5'）

- 用户可以在界面上通过点击或输入完成数独题目（1.5'）
- 用户完成数独题目后可以得到反馈，知道自己的题目是否做对（1'）

**【注意】** 选择完成本附加题目的同学，需要将GUI与数独游戏生成器作为两个工程开发，后者可以作为依赖库为前者提供调用接口，但不可以把两个工程直接混在一起。GUI相关的部分也需要提供新的可执行文件，放在根目录的GUIBIN文件夹下。

## 四、测试须知

所有提交到Github上的项目均需要建立一个名字为 `BIN` 的文件夹，里面必须含有可执行文件与相关的依赖库，请注意以下两点：

- 确保可执行文件的名字统一为 `sudoku.exe`。（注：Java和Python程序需要提供可接受参数的执行脚本）
- 确保生成的棋盘文件 `sudoku.txt` 与可执行文件在同一目录下，生成文件时请使用**相对路径**！

一个示例组织目录如下所示：

```

/ SudokuProject(工程名字自行指定即可)
/ main.cpp
/ generator.cpp
/ BIN
  / Lib.dll(exe运行需要的动态链接库文件)
  / sudoku.exe
  / sudoku.txt (运行exe后生成)

```

在测试时，将以命令行运行可执行文件的方式进行批量测试，参数及其约定如下

参数名字	参数意义	用法示例
-c[必选]	需要的棋盘数量	示例：sudoku.exe -c 20 [表示生成20个数独题目]

## 五、博客撰写要求：

- 发表在你的个人博客上，也可以同时转发到你的团队博客上来增加你们团队博客的人气。博客共15分，具体要求如下：
- 1) 在文章开头给出Github项目地址。（1’）
  - 2) 在开始实现程序之前，在下述PSP表格记录下你估计将在程序的各个模块的开发上耗费的时间。（0.5’）
  - 3) 解题思路描述。即刚开始拿到题目后，如何思考，如何找资料的过程。（3’）
  - 4) 设计实现过程。设计包括代码如何组织，比如会有几个类，几个函数，他们之间关系如何，关键函数是否需要画出流程图？单元测试是怎么设计的？（4’）
  - 5) 记录在改进程序性能上所花费的时间，描述你改进的思路，并展示一张性能分析图（由VS 2017的性能分析工具自动生成），并展示你程序中消耗最大的函数。（3’）
  - 6) 代码说明。展示出项目关键代码，并解释思路与注释说明。（3’）
  - 7) 在你实现完程序之后，在下述PSP表格记录下你在程序的各个模块上实际花费的时间。（0.5’）

附：PSP 2.1表格

PSP2.1	Personal Software Process Stages	预估耗时（分钟）	实际耗时（分钟）
Planning	计划		
· Estimate	· 估计这个任务需要多少时间		
Development	开发		
· Analysis	· 需求分析（包括学习新技术）		
· Design Spec	· 生成设计文档		
· Design Review	· 设计复审（和同事审核设计文档）		
· Coding Standard	· 代码规范（为目前的开发制定合适的规范）		
· Design	· 具体设计		
· Coding	· 具体编码		
· Code Review	· 代码复审		
· Test	· 测试（自我测试，修改代码，提交修改）		
Reporting	报告		

· Test Report	· 测试报告		
· Size Measurement	· 计算工作量		
· Postmortem & Process Improvement Plan	· 事后总结, 并提出过程改进计划		
	合计		

## 六、评分规则

本次个人项目分数由三部分组成，分别是

- (1) 博客 — 15分，分数组成在博文规范中。
- (2) 程序 — 35分

5分为源代码管理评分，该评分主要通过源代码管理中的commit注释信息，增量修改的内容，是否有运行说明等给分。

20分为正确性评分，正确性测试中输入范围限制在 1-1000，要求程序在 60 s 内给出结果，超时则认定运行结果无效。

10分为性能评分，性能测试中输入范围限制在 10000-1000000，没有时间的最小要求限制。

当程序的正确性评分等于20分时才可以参与性能评分环节，所以请各位同学务必保证自己程序的正确性。

性能评分将采取档级评分制度，助教将根据同学们的程序跑同一数据耗费的时间长度将程序分为若干档，每一档的同学得到的分数为 10/档级数。

- (3) 附加题 — 5分。
- (4) 注意事项：

按时间完成并提交—正常评分

晚交一周以内—0分

晚交一周以上或不交—倒扣本次作业分数

抄袭—倒扣2倍本次作业分数【严禁代码与博客等一切形式的抄袭！请各位同学千万不要触碰底线，勿谓言之不预也！】