

浙江大学计算机学院

Java 程序设计课程报告

2024—2025 学年秋冬学期

题目	数独游戏的设计与开发
学号	3220104147
学生姓名	余卓耘
所在专业	软件工程
所在班级	软工 2202

目 录

1 引言.....	1
1.1 设计目的.....	1
1.2 设计说明.....	1
2 总体设计.....	2
2.1 功能模块设计.....	2
2.2 流程图设计.....	2
3 详细设计.....	4
3.1 数独的布局设计.....	4
3.2 核心类设计.....	4
4 测试与运行.....	6
4.1 程序测试.....	6
4.2 程序运行.....	6
5 总结.....	10
参考文献.....	11

1 引言

本次开发的是一个数独游戏，包含生成器、求解器，可以让我这样的初学者对 Java 语言中的各项功能有更好的理解和使用，通过具体的程序来加深对 Java 语言的掌握，提高自己的编程水平，为以后的学习、科研与工作打好基础。

1.1 设计目的

数独是一种古老而益智的游戏。本文使用 Java 语言编写一个数独游戏。具体功能如下：

(1)进入游戏后会有欢迎提示语，用户可以依据提示语操作。程序分为“求解器”（模式 1）“生成器”（模式 2）2 种模式。用户输入 1 或 2 选择模式。

(2) 选择生成器后，用户可以指定提示数字的数量（推荐 25-35 个），程序会生成一个符合规则的数独题目，同时生成对应的完整解答，保证生成的题目具有唯一解。

(3) 选择求解器后，用户输入代表数字分布、区域形状的两个矩阵(9x9 的数独矩阵，0 表示空格；9x9 的区域矩，1-9 表示不同区域)，求解器自动给出答案矩阵。

(4) 如果用户输入不符合要求，或数独无解或无穷解，程序会给出相应报错信息，并重新开始。

(5) 结束一轮数独后，程序会询问用户是否继续，按提示输入 y 则继续新一轮游戏，输入 n 则退出。命令行界面按 Ctrl+C 也可以强制退出游戏。

(6) 程序的主要特点：支持不规则数独（通过区域矩阵定义）、采用优化的回溯算法进行求解、生成器保证题目唯一解、具有完善的输入验证和错误处理、提供友好的用户交互界面。

1.2 设计说明

本程序采用 Java 程序设计语言，在 VSCode 平台下编辑、编译与调试。具体程序由余卓耘(学号 3220104147)单人独立完成。

2 总体设计

2.1 功能模块设计

本程序需实现的主要功能有：

- (1) 用户可以选择生成器或求解器模式；
- (2) 系统会自动生成题目或给出解答；
- (3) 一轮结束后用户可以选择是否继续游戏或退出；
- (4) 具有输入检查与报错模块。

程序的总体功能如图 1 所示：



图 1 总体功能图

2.2 流程图设计

程序总体流程如图 2 所示：

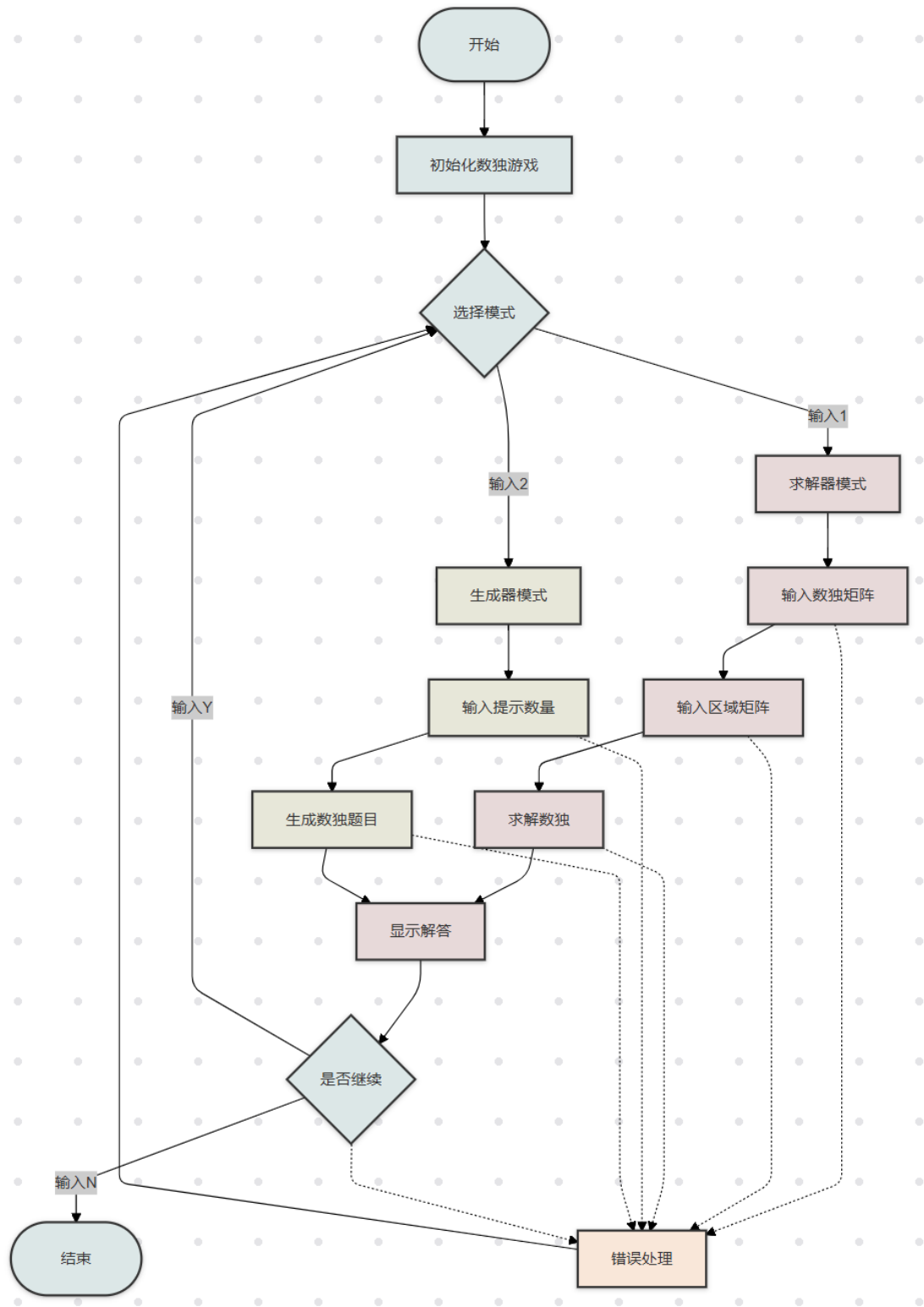


图2 总体流程图

3 详细设计

3.1 数独的布局设计

系统的整体布局采用命令行交互界面，主要包含求解器、生成器两大功能模块，以及错误处理模块。

3.2 核心类设计

Position 类是一个基础工具类，实现了坐标位置的封装。

SudokuHelper 类提供了通用工具方法。

下图是主要类之间的 UML 关系图：

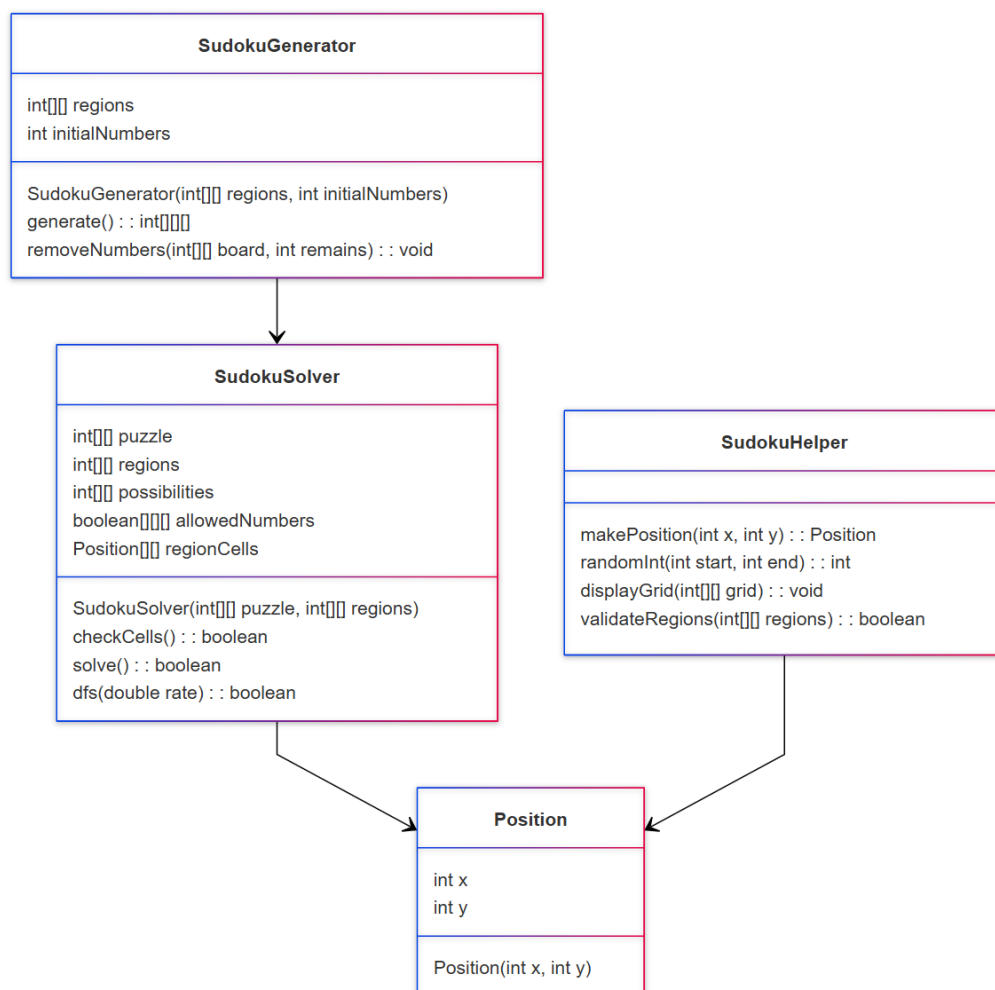


图 3 主要类的 UML 图

以下是 UML 图中有关数据和方法的详细说明：

(1) 成员变量

① Position 类表示数字在矩阵中的位置，x、y 表示二维坐标位置：

② SudokuSolver 类表示数独的解：

puzzle：存储数独题目矩阵

regions：存储区域定义矩阵

possibilities：存储每个位置可能的数字数量

allowedNumbers：三维数组标记每个位置允许填入的数字

regionCells：存储每个区域包含的单元格位置

(2) 方法

① SudokuHelper 类表示数独矩阵的创建与检验：

makePosition：创建位置对象

randomInt：生成指定范围内的随机数

displayGrid：显示数独矩阵

validateRegions：验证区域定义的合法性

② SudokuGenerator 类表示数独生成：

regions：存储区域定义矩阵

initialNumbers：生成题目时保留的提示数字数量

③ SudokuSolver 类：

checkCells：检查当前数独状态是否合法

solve：解决数独问题的入口方法

dfs：使用深度优先搜索求解数独的核心算法

4 测试与运行

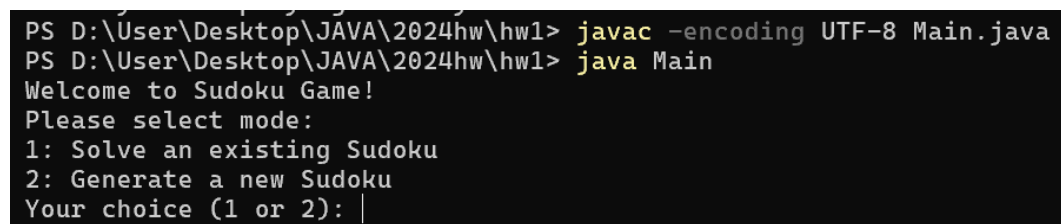
4.1 程序测试

在程序代码基本完成后，经过不断的调试与修改，最后测试本次所设计的数独游戏能够正常运行，没有出现明显的错误和漏洞，但是在一些细节方面仍然需要完善，比如在数独游戏中可以加入更多其他功能增加可玩性。总的来说本次设计在功能上已达到要求。

测试结果：数独求解、数独生成、错误处理等在 4.2 程序运行均有截图。

4.2 程序运行

程序运行主界面如图 4 所示：



```
PS D:\User\Desktop\JAVA\2024hw\hw1> javac -encoding UTF-8 Main.java
PS D:\User\Desktop\JAVA\2024hw\hw1> java Main
Welcome to Sudoku Game!
Please select mode:
1: Solve an existing Sudoku
2: Generate a new Sudoku
Your choice (1 or 2): |
```

图 4 程序主界面

求解模式如图 5 所示：


```

Your choice (1 or 2): 1

Please input the 9x9 Sudoku puzzle (use 0 for empty cells):
5 3 0 0 7 0 0 0 0
6 0 0 1 9 5 0 0 0
0 9 8 0 0 0 0 6 0
8 0 0 0 6 0 0 0 3
4 0 0 8 0 3 0 0 1
7 0 0 0 2 0 0 0 6
0 6 0 0 0 0 2 8 0
0 0 0 4 1 9 0 0 5
0 0 0 0 8 0 0 7 9

Please input the 9x9 region matrix (1-9 for regions):
1 1 1 2 2 2 3 3 3
1 1 1 2 2 2 3 3 3
1 1 1 2 2 2 3 3 3
4 4 4 5 5 5 6 6 6
4 4 4 5 5 5 6 6 6
4 4 4 5 5 5 6 6 6
7 7 7 8 8 8 9 9 9
7 7 7 8 8 8 9 9 9
7 7 7 8 8 8 9 9 9

Input Puzzle:
5 3 0 0 7 0 0 0 0
6 0 0 1 9 5 0 0 0
0 9 8 0 0 0 0 6 0
8 0 0 0 6 0 0 0 3
4 0 0 8 0 3 0 0 1
7 0 0 0 2 0 0 0 6
0 6 0 0 0 0 2 8 0
0 0 0 4 1 9 0 0 5
0 0 0 0 8 0 0 7 9

Region Matrix:
1 1 1 2 2 2 3 3 3
1 1 1 2 2 2 3 3 3
1 1 1 2 2 2 3 3 3
4 4 4 5 5 5 6 6 6
4 4 4 5 5 5 6 6 6
4 4 4 5 5 5 6 6 6
7 7 7 8 8 8 9 9 9
7 7 7 8 8 8 9 9 9
7 7 7 8 8 8 9 9 9

Sudoku solved successfully:
5 3 4 6 7 8 9 1 2
6 7 2 1 9 5 3 4 8
1 9 8 3 4 2 5 6 7
8 5 9 7 6 1 4 2 3
4 2 6 8 5 3 7 9 1
7 1 3 9 2 4 8 5 6
9 6 1 5 3 7 2 8 4
2 8 7 4 1 9 6 3 5
3 4 5 2 8 6 1 7 9

```

图5 求解模式

生成模式如图 6 所示:

```
Your choice (1 or 2): 2

How many hints do you want? (Recommended: 25-35)
Enter number of hints: 30

Please input the 9x9 region matrix (1-9 for regions):
1 1 1 2 2 2 3 3 3
1 1 1 2 2 2 3 3 3
1 1 1 2 2 2 3 3 3
4 4 4 5 5 5 6 6 6
4 4 4 5 5 5 6 6 6
4 4 4 5 5 5 6 6 6
7 7 7 8 8 8 9 9 9
7 7 7 8 8 8 9 9 9
7 7 7 8 8 8 9 9 9

Region Matrix:
1 1 1 2 2 2 3 3 3
1 1 1 2 2 2 3 3 3
1 1 1 2 2 2 3 3 3
4 4 4 5 5 5 6 6 6
4 4 4 5 5 5 6 6 6
4 4 4 5 5 5 6 6 6
7 7 7 8 8 8 9 9 9
7 7 7 8 8 8 9 9 9
7 7 7 8 8 8 9 9 9

Generated Puzzle:
6 0 2 0 3 5 0 0 0
0 8 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
5 0 4 0 1 0 0 0 3
2 3 0 7 5 0 0 0 0
9 0 8 0 0 0 0 5 4
3 2 0 0 8 0 0 0 0
0 0 7 0 0 1 3 8 2
0 0 0 3 7 2 9 4 0

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

图 6 生成模式

报错处理如图 7 所示：

```
Please select mode:
1: Solve an existing Sudoku
2: Generate a new Sudoku
Your choice (1 or 2): 1

Please input the 9x9 Sudoku puzzle (use 0 for empty cells):
5 3 0 0 7 0 0 0 0
6 0 0 1 9 5 0 0 0
0 9 8 0 0 0 0 6 0
8 0 0 0 6 0 0 0 3
4 0 0 8 0 3 0 0 1
7 0 0 0 2 0 0 0 6
0 6 0 0 0 0 2 8 0
0 0 0 4 1 9 0 0 5
0 0 0 0 8 0 0 7 9

Please input the 9x9 region matrix (1-9 for regions):
1 1 1 2 2 2 3 3 3
1 1 2 2 2 3 3 3 3
1 2 2 2 3 3 3 3 4
5 5 2 2 3 3 3 4 4
5 5 5 6 6 6 4 4 4
5 5 6 6 6 7 7 7 8
5 6 6 6 7 7 7 8 8
6 6 6 7 7 7 8 8 8
6 6 7 7 7 8 8 8 8
Error: Invalid region matrix. Each region number (1-9) must be used and form a connected area.Let's try again.

Welcome to Sudoku Game!
Please select mode:
1: Solve an existing Sudoku
2: Generate a new Sudoku
Your choice (1 or 2): 3
Error: Invalid choice. Please enter 1 or 2.Let's try again.

Welcome to Sudoku Game!
Please select mode:
1: Solve an existing Sudoku
2: Generate a new Sudoku
Your choice (1 or 2): 2

How many hints do you want? (Recommended: 25-35)
Enter number of hints: 82
Error: Number of hints must be between 17 and 81.Let's try again.

Would you like to play again? (y/n):
s
Error: Please enter 'y' for yes or 'n' for no.

Would you like to play again? (y/n):
|
```

图 7 报错处理

一轮数独结束后如图 8 所示：

```
Would you like to play again? (y/n):
y

Welcome to Sudoku Game!
Please select mode:
1: Solve an existing Sudoku
2: Generate a new Sudoku
```

图 8 一轮数独结束后

选择退出后如图 9 所示：

```
Would you like to play again? (y/n):  
n  
Thank you for playing! Goodbye!
```

图 9 退出

5. 总结

这是我第一次使用 Java 编写大程，之前由于对 Java 语法、特性的陌生，遇到过诸多问题，但在调试代码的过程中，我在实践中学到了很多，逐步提升了自己的技术水平，收益很大。

通过开发这个数独游戏项目，我深入学习了 Java 面向对象编程和错误处理机制。在开发过程中遇到了多个具有挑战性的问题：首先是数独生成算法的实现。为了保证生成的数独题目具有唯一解，我采用了先生成完整解，再随机移除数字的方法。在这个过程中，我学会了如何使用回溯算法，以及如何通过随机化来增加题目的多样性。其次是错误处理机制的设计。我实现了多层次的输入验证：在调试过程中，我学会了使用断言和异常处理来定位问题。比如，在数独生成时，通过在关键节点添加数据验证，及时发现非法状态；通过异常处理机处理各种错误情况，提供清晰的错误提示。

这个项目让我对 Java 的异常处理、面向对象设计有了更深的理解。同时，通过实现回溯算法和约束传播，也提高了我的算法设计能力。最重要的是，我学会了如何将一个复杂问题分解为可管理的模块，并通过清晰的接口组织它们。收获很大，希望在未来学习更多知识，做的更好。

参考文献

- [1] 耿祥义. Java 大学实用教程[M]. 北京: 清华大学出版社, 2009.
- [2] 耿祥义. Java 课程设计[M]. 北京: 清华大学出版社, 2008.
- [3] 丁振凡. Java 语言实验教程[M]. 北京: 北京邮电大学出版社, 2005.
- [4] 郑莉. Java 语言程序设计[M]. 北京: 清华大学出版社, 2006.