

餐厅点餐系统

作 者： 曾极涵

学 院： 人文与管理学院

专 业： 信息管理与信息系统

年 级： 2019 级

学 号： 3190707011

指导教师： 黄敏

日 期： 2021年12月20日

目录

一、 系统分析.............................................................................................................................3 系统架构.....................................................................................................................................3 模块设计.....................................................................................................................................4 二、 数据库设计.........................................................................................................................4

1. 商品表 (goods）...............................................................................................................4
2. 订单套餐表 (order\_goods).............................................................................................. 4
3. 订单信息表 (order\_info).................................................................................................. 4
4. 用户信息表 (user).............................................................................................................5 三、 部分关键代码.....................................................................................................................5

1.LogOnFrm.java........................................................................................................................5

2.UserOrderFrm.java................................................................................................................. 7

3.CheckOrderInterFrm.java.....................................................................................................10

1. GoodsAddInterFrm.java..................................................................................................11 四、 系统运行.......................................................................................................................... 13
2. 运行流程图...................................................................................................................... 13
3. 登陆界面（LogOnFrm.java）........................................................................................ 13
4. 注册界面（RegisterFrm.java）......................................................................................14
5. 用户点餐界面（UserOrderFrm.java）..........................................................................14
6. 开始点餐界面（OrderGoodsInterFrm.java）.............................................................. 14
7. 查看订单界面（CheckOrderInterFrm.java）............................................................... 15
8. 信息修改界面（AccountModify.java）........................................................................ 15
9. 点餐系统管理祝界面（AdminFrm.java）.................................................................... 16
10. 新增套餐界面（GoodsAddInterFrm.java）................................................................. 16
11. 套餐维护界面（GoodsManageInterFrm.java）...................................................... 17
12. 订单处理界面（DealOrderInterFrm.java）..............................................................17 五、遇到的问题及总结.................................................................................................................. 18
13. 遇到的问题.................................................................................................................. 18

1.mysql 数据库启动失败................................................................................................ 18

2.数据库驱动版本与数据库版本不匹配.......................................................................19

3.关于 mysql 中[Err] 1451 -Cannot delete or update a parent row: a foreign key constraint fails 问题.........................................................................................................22

4.java.sql 包中的 PreparedStatement 与 Statement 的区别.......................................23

6.用户与管理员注销之后无法弹出登陆界面...............................................................23

1. 总结...............................................................................................................................24

**摘要：**基于Java Swing开发的数独游戏系统。系统集成了用户登录与注册功能，确保只有注册成功的用户才能进入游戏。同时，系统提供了帮助功能，以独立弹窗的形式呈现，提升用户体验。系统采用文件存储用户数据，确保用户信息的持久化。界面设计注重美观与一致性，采用现代化布局和颜色搭配，增强用户交互感受。

# 一、 系统分析

### 1.1 系统功能

* **用户管理**：包括用户注册、登录功能，确保只有认证用户才能进入游戏。
* **数独游戏**：支持多种难度（4x4、6x6、9x9），提供游戏控制、保存与加载、提交答案、提示等功能。
* **帮助功能**：提供独立的帮助弹窗，详细介绍游戏规则与操作说明。
* **界面设计**：美观且统一的用户界面，提升用户体验。

### 1.2 系统架构

系统主要由以下几个模块组成：

* **用户界面模块**：包括登录与注册界面、数独游戏界面、帮助弹窗。
* **数据管理模块**：负责用户数据的读取与写入，确保数据的持久化。
* **游戏逻辑模块**：包含数独生成、验证、提示等核心功能。
* **辅助功能模块**：如计时器、提示功能等，提升游戏互动性。

### 1.3 技术选型

* **编程语言**：Java
* **图形用户界面**：Java Swing
* **数据存储**：文本文件（users.txt）用于存储用户账号与密码，序列化文件（sudoku\_save.dat）用于保存游戏进度。

# 部分关键代码

### 3.1 用户数据加载与保存

Java

// 加载用户数据库从文件private void loadUserDatabase() {

File file = new File(USER\_DATA\_FILE);

if (!file.exists()) {

try {

file.createNewFile(); // 创建文件如果不存在

} catch (IOException e) {

JOptionPane.showMessageDialog(this, "无法创建用户数据文件。", "错误", JOptionPane.ERROR\_MESSAGE);

}

}

try (BufferedReader br = new BufferedReader(new FileReader(USER\_DATA\_FILE))) {

String line;

while ((line = br.readLine()) != null) {

if (line.trim().isEmpty()) continue; // 跳过空行

String[] parts = line.split(":", 2);

if (parts.length >= 2) {

String username = parts[0].trim();

String password = parts[1].trim();

userDatabase.put(username, password);

}

}

} catch (IOException e) {

JOptionPane.showMessageDialog(this, "读取用户数据时出现错误。", "错误", JOptionPane.ERROR\_MESSAGE);

}

}

// 将新用户写入文件private void saveUserToFile(String username, String password) {

try (BufferedWriter bw = new BufferedWriter(new FileWriter(USER\_DATA\_FILE, true))) {

bw.write(username + ":" + password);

bw.newLine();

} catch (IOException e) {

JOptionPane.showMessageDialog(this, "保存用户数据时出现错误。", "错误", JOptionPane.ERROR\_MESSAGE);

}

}

**说明**：上述代码负责从users.txt文件中加载用户数据，并在用户注册成功后将新的账号密码追加到文件中，实现用户数据的持久化。

### 3.2 数独谜题生成与验证

Java

private void generatePuzzle() {

int[][] board = new int[size][size];

fillBoard(board);

puzzle = board;

copyBoard(board, solution);

removeNumbers();

}

private boolean fillBoard(int[][] board) {

for (int row = 0; row < size; row++) {

for (int col = 0; col < size; col++) {

if (board[row][col] == 0) {

for (int num = 1; num <= size; num++) {

if (isSafe(board, row, col, num)) {

board[row][col] = num;

if (fillBoard(board)) {

return true;

}

board[row][col] = 0;

}

}

return false;

}

}

}

return true;

}

private boolean isSafe(int[][] board, int row, int col, int num) {

for (int x = 0; x < size; x++) {

if (board[row][x] == num || board[x][col] == num) {

return false;

}

}

int sqrt = (int) Math.sqrt(size);

int boxRowStart = row - row % sqrt;

int boxColStart = col - col % sqrt;

for (int r = boxRowStart; r < boxRowStart + sqrt; r++) {

for (int d = boxColStart; d < boxColStart + sqrt; d++) {

if (board[r][d] == num) {

return false;

}

}

}

return true;

}

**说明**：此部分代码负责生成数独谜题。generatePuzzle方法初始化棋盘，调用fillBoard递归填充数字，确保每个数字的放置符合数独规则。isSafe方法用于判断在特定位置放置某个数字是否安全。

### 3.3 用户界面美化

java

private void styleButton(JButton button) {

button.setFont(new Font("微软雅黑", Font.BOLD, 14));

button.setBackground(new Color(70, 130, 180));

button.setForeground(Color.WHITE);

button.setFocusPainted(false);

button.setCursor(new Cursor(Cursor.HAND\_CURSOR));

// 添加鼠标悬停效果

button.addMouseListener(new MouseAdapter() {

@Override

public void mouseEntered(MouseEvent e) {

button.setBackground(new Color(100, 149, 237));

}

@Override

public void mouseExited(MouseEvent e) {

button.setBackground(new Color(70, 130, 180));

}

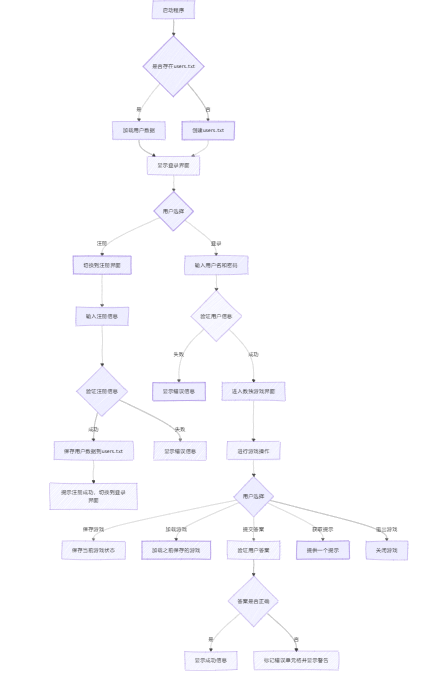
});

}

**说明**：该方法用于统一设置按钮的样式，包括字体、颜色、鼠标悬停效果等，提升界面的美观性和一致性。

# 四、 系统运行

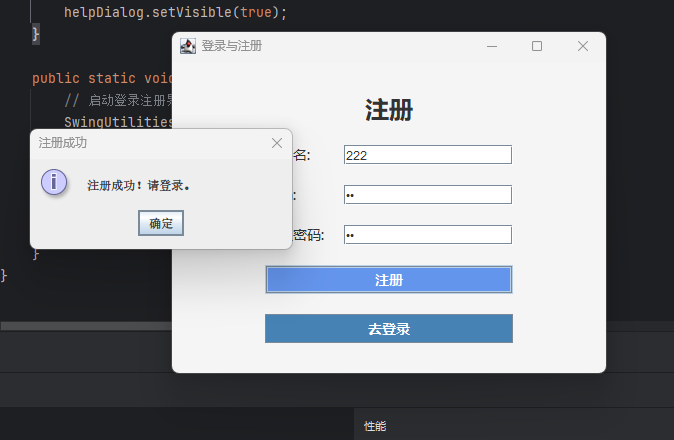
## 1. 运行流程图



## 2.登陆界面



## 3.注册界面



## 4.帮助界面



## 5.游戏界面



# 五、遇到的问题及总结

## （一）遇到的问题

### 1.mysql 数据库启动失败

· **用户数据文件的读写问题**：

* · **问题描述**：在首次运行程序时，users.txt文件可能不存在，导致读取用户数据时出现错误。
* **解决方案**：在加载用户数据库时，先检查users.txt文件是否存在，若不存在则创建新的文件，避免读取错误。

· **递归数独生成导致栈溢出**：

* · **问题描述**：在生成较大规模的数独（如9x9）时，递归调用可能导致栈溢出错误。
* **解决方案**：优化递归算法，限制递归深度，或采用非递归算法生成数独谜题。

· **用户密码以明文形式存储**：

* · **问题描述**：当前实现中，用户密码以明文形式存储在users.txt文件中，存在安全风险。
* **解决方案**：采用密码哈希算法（如SHA-256）对密码进行加密后再存储，提高安全性。

· **GUI界面布局不兼容不同分辨率**：

* · **问题描述**：在不同分辨率下，界面组件可能显示不全或布局紊乱。
* **解决方案**：使用更灵活的布局管理器（如GridBagLayout），并进行界面适配性测试，确保在不同分辨率下均能正常显示。

· **数独提示功能无限递归**：

* · **问题描述**：在用户所有单元格均已填满且正确时，提示功能会进入无限递归，导致程序崩溃。
* **解决方案**：在提示功能中添加判断，若没有空白单元格则不提供提示，避免递归调用。

## （二）总结

通过本次项目开发，深入理解了Java Swing在图形界面开发中的应用，掌握了布局管理器的使用方法，特别是GridBagLayout的灵活性与复杂性。同时，增强了对文件读写操作的掌握，了解了如何实现用户数据的持久化存储。项目中遇到的递归算法优化、安全性提升等问题，使得开发者在实际项目中积累了宝贵的经验。

在用户界面设计方面，注重美观与用户体验，采用现代化的颜色搭配与字体选择，提升了系统的整体视觉效果。同时，通过不断的测试与优化，解决了多分辨率兼容性、递归深度等技术难题，确保系统的稳定性与可靠性。

未来可以进一步优化系统功能，如引入更安全的密码存储机制，增加用户头像、个性化设置等，提升系统的实用性与用户粘性。同时，可以考虑将用户数据存储迁移到数据库中，增强系统的扩展性与安全性。