

# 01 MAIN 主函数开发

## 主函数

### 1. 概述

本项目是一个基于命令行的宿舍管理系统，旨在通过简单的交互方式，提供学生和管理员分别管理宿舍、用户信息、维修请求等功能。系统主要分为学生和管理员两个角色，分别具有不同的菜单和操作权限。通过一系列功能模块，学生可以查看宿舍信息、申请调整住宿、提交维修请求等；管理员则可以管理宿舍楼、用户、维修请求等。

---

### 2. 项目功能描述

#### 2.1 主程序与菜单设计

主程序的入口是一个主菜单，用户可以选择进入学生菜单、管理员菜单，或退出系统。通过 `getChoice` 函数获取用户输入，并根据选择执行相应的操作。输入检查通过循环实现，确保用户输入合法（如范围检查、类型检查等）。

- **主菜单：**提供三项基本功能选择：
  - 学生菜单
  - 管理员菜单
  - 退出系统

#### 2.2 学生登录与功能模块

学生登录模块包括学生身份验证、密码检查、密码修改等功能。系统首先要求学生输入学号和密码，并通过 `userManager->loginUser` 验证身份。若登录成功，学生将进入个人功能菜单。

- **学生菜单：**
  - 查看宿舍楼和房间信息
  - 申请住宿调整（如入住、退宿、换宿）
  - 提交维修请求
  - 查看系统通知和请求
  - 修改密码
  - 退出登录

在学生登录时，若检测到密码为默认密码（例如学号），系统会提醒学生修改密码。

## 2.3 管理员登录与功能模块

管理员功能模块与学生功能模块有所不同。管理员需要输入管理员账号和密码进行身份验证。管理员登录后，可以执行更复杂的操作，如管理宿舍楼、用户信息、维修请求、住宿安排等。

- **管理员菜单：**
  - 管理宿舍楼信息
  - 用户管理（添加、删除、查看用户）
  - 生成报表（如宿舍使用情况、入住率等）
  - 维修处理中心
  - 住宿管理中心
  - 退出登录

管理员通过各种子菜单来执行任务，包括用户管理、宿舍楼管理、住宿安排等。

## 2.4 住宿调整功能

宿舍调整模块允许学生申请入住、退宿、换宿等操作。每个操作会触发相应的后续处理逻辑，如检查是否有待审批的请求，确认申请是否符合条件等。

## 2.5 管理员权限功能

管理员可以：

- **宿舍管理：**添加、删除宿舍楼，查看和修改宿舍楼信息。
- **用户管理：**通过学号或姓名查看、添加、删除用户，修改用户密码。
- **报表生成：**生成宿舍使用情况报表和入住率报表。

---

## 3. 关键功能实现

### 3.1 输入验证与错误处理

在各个菜单选项中，输入处理是非常关键的部分。通过 `getChoice` 函数进行多次输入验证，确保用户输入的选项在有效范围内，并能够处理错误输入（如输入非数字字符时清空输入缓冲区）。

遇到输入错误时，系统会反复提示用户进行正确输入，保证操作的稳定性和用户体验。

### 3.2 密码安全

在学生和管理员登录时，密码输入采用隐藏模式，用户输入时密码不会显示在屏幕上。系统还支持输入 \* 来查看密码，确保用户在登录过程中能够保密密码。

### 3.3 宿舍管理与申请调整

在学生菜单中，学生可以通过“申请住宿调整”功能提交入住、退宿或换宿的请求。如果存在待审批的请求，系统会阻止学生重复提交申请，避免无效的操作。

管理员拥有权限审批学生的住宿申请并进行相应的住宿安排。通过 `dormManageMenu`，管理员可以安排住宿、处理学生申请等。

### 3.4 用户与宿舍楼管理

管理员可以通过 `manageUsers` 菜单管理用户，包括添加、删除用户、查询用户信息等；同时，管理员还可以通过 `manageDormitories` 菜单进行宿舍楼的管理，添加、删除宿舍楼，或查看宿舍楼的信息。

---

## 4. 开发过程中的问题与解决方案

### 4.1 输入合法性校验

在系统的初步开发过程中，出现了输入时未能正确校验的情况。例如，在获取用户菜单选择时，若用户输入了非法字符，系统会直接崩溃。为此，开发过程中添加了大量的输入校验代码，通过 `cin.fail()` 清除输入缓冲区，确保程序不会因错误输入而异常退出。

### 4.2 密码保护与修改（陈奇乐完成）

在实现密码输入和修改时，遇到了密码显示的问题。最初密码的输入没有加密处理，导致密码显示在终端上。解决方案是通过 `clearScreen` 和 `passwordHide` 等方法隐藏用户输入的密码，并在必要时进行密码修改提醒。

---

## 5. 收获与反思

通过本次宿舍管理系统的开发，深刻认识到了良好的用户体验与输入验证的重要性。特别是在输入部分，通过精心设计输入框架与错误处理机制，极大地提升了系统的稳定性。同时，项目中也加深了对C++中面向对象编程的理解，掌握了如何利用类和对象进行系统功能的模块化。