北京石油化工学院

BEIJING INSTITUTE OF PETROCHEMICAL TECHNOLOGY

Java 应用开发技术

学生宿舍管理系统报告

组　　长　：　詹振华

成 员 1 ：　李靖亚

2023 年 12 月 22 日

目录

一、实验目的：

1）理解对象和类的概念及交互：通过创建 Student 和 Dormitory 类，学习如何定义类的属性和方法，以及如何通过对象实例化这些类，理解和实践如何在不同对象（如学生和宿舍）之间建立关联和通信，从而模拟现实世界中的复杂关系。

2）学习集合框架的使用：通过使用 HashMap 和 ArrayList 等集合来管理学生和宿舍的数据，提高对 Java 集合框架的理解和应用能力。

3)实践文件的输入/输出操作：学习如何使用 Apache POI 库处理 Excel 文件，包括从 Excel 文件中读取学生数据以及将数据导出到 Excel 文件，增强处理数据和文件的实践技能。

4）加深对异常处理的理解：通过处理文件输入/输出过程中可能出现的异常，加深对 Java 异常处理机制的理解。

5)理解和实现排序算法：通过宿舍评分排名的功能，学习如何应用排序算法对数据进行排序，同时可以改进排序算法如选择排序、双向冒泡排序等，以及如何利用 Java 的流（Stream）API 来进行高效的数据操作。

6)提高系统设计逻辑及能力：通过设计和实现一个宿舍管理系统，提高系统分析、设计和实现的能力，学习如何将一个复杂的问题分解成可管理的小部分。

7)前端页面开发：学习创建和处理表单，通过编写 html、js、css 文件提高前端设计水平，理解客户端和服务器架构，掌握前后端开发。

二、主要方法：

1）基本语言结构

变量定义：使用了基本数据类型（如 double 和 int）和引用数据类型（如 String 和自定义的类 Student 和 Dormitory）。

控制流语句：使用了 switch 和 if-else 语句来根据条件执行不同的代码块。

循环：使用 for 循环来遍历数据集合。

2）面向对象编程（OOP）

类和对象：创建了 Student 和 Dormitory 类，并在这些类中定义了属性和方法。

封装：使用了 private 关键字来隐藏类的内部状态和行为，提供了公共的 getter 和 setter 方法来访问和修改这些属性。

构造方法：为 Student 和 Dormitory 类定义了构造方法来初始化对象。

3）集合框架

List：使用了 ArrayList 来存储和管理 Student 对象。

Map：使用了 HashMap 来存储和管理 Dormitory 对象和 Student 对象，以便快速访问。

4）异常处理

try-catch 语句：在处理文件输入/输出时使用 try-catch 语句来捕获和处理可能发生的 IOException。

5）文件 I/O

FileInputStream 和 FileOutputStream：用于读取和写入文件。

Apache POI 库：使用此库来读取和写入 Microsoft Excel 文件（.xlsx 格式）。

6）Java Stream API

使用了 Stream API 进行数据的排序和收集操作，特别是在宿舍评分排名功能中。

7）Java 标准库的其他部分

Scanner：用于创建一个对象来获取用户的输入。

System.out：用于在控制台打印信息和用户提示。

8）基本输入输出

通过 System.in 和 System.out 来实现用户交互，接收用户输入，并显示结果。

9）排序技术与排序算法

Stream API 和 Comparator：使用 Stream API 结合 lambda 表达式和

Comparator 接口实现自定义排序逻辑，按照宿舍评分降序排列。并改进选择排序和双向冒泡排序等方法。

10）前端界面设计

利用 html 编写 web，结合 java 功能设计，编写 html、css、js 文件设计可视化的界面。

11）系统开发

在前后端开发完成时，接入数据库如 MySQL、Access 等，并部署云端申请域名。

三、系统结构

系统主要功能包括添加查询学生、添加分配宿舍、设置宿舍评分和排名、读取保存数据到本地 excel 等。开发了前端的宿舍评分的 web 表单。

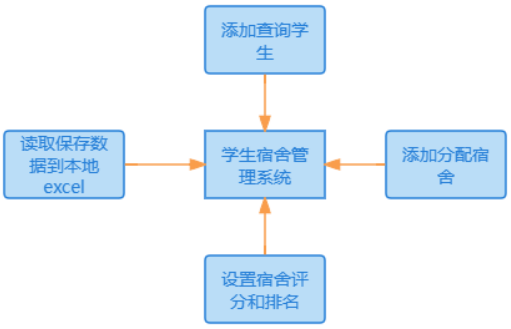


图 1　系统功能图

系统运行命令行界面：

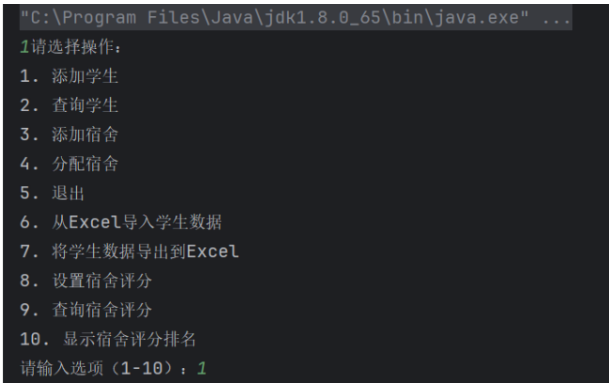
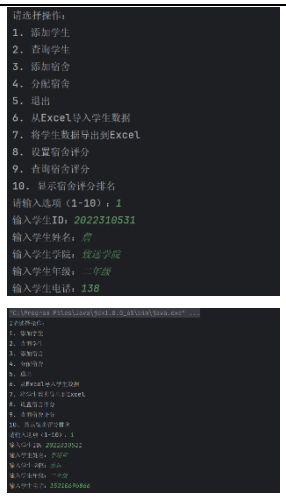


图 2　系统界面

功能	展示图

添加查询学生	 请选择操作： 1. 添加学生 2. 查询学生 3. 添加宿舍 4. 分配宿舍 5. 退出 6. 从Excel导入学生数据 7. 将学生数据导出到Excel 8. 设置宿舍评分 9. 查询宿舍评分 10. 显示宿舍评分排名 请输入选项（1-10）：1 输入学生ID：2022310531 输入学生姓名：xx 输入学生学院：软工学院 输入学生年级：一年级 输入学生电话：138
添加分配宿舍	 请选择操作： 1. 添加学生 2. 查询学生 3. 添加宿舍 4. 分配宿舍 5. 退出 6. 从Excel导入学生数据 7. 将学生数据导出到Excel 8. 设置宿舍评分 9. 查询宿舍评分 10. 显示宿舍评分排名 请输入选项（1-10）：4 输入学生ID：2022310531 输入宿舍楼名：6C 输入房间号：2415
设置宿舍评分排名	 请选择操作： 1. 添加学生 2. 查询学生 3. 添加宿舍 4. 分配宿舍 5. 退出 6. 从Excel导入学生数据 7. 将学生数据导出到Excel 8. 设置宿舍评分 9. 查询宿舍评分 10. 显示宿舍评分排名 请输入选项（1-10）：10 宿舍评分排名： 1. 6C 楼 2415 号室 - 评分: 96.0 2. 6C 楼 2413 号室 - 评分: 94.0 3. 6C 楼 2418 号室 - 评分: 82.0 请选择操作：
读取导入文件	
Web 表单	 宿舍评分系统 楼号 分数 添加

四、项目结构（源代码见附录）：

4.1 java 后端

4.1.1 Student 类

1）定义私有属性: id, name, college, grade, phone

2）构造函数(Student): 初始化所有属性

3）Getter 和 Setter 方法: 为每个属性提供访问和更新

类结构:

类定义: class Student { ... }

属性定义: 私有属性 id, name, college, grade, phone

构造器: public Student(...) { ... } 用于初始化对象

方法定义: 为每个属性提供 get 和 set 方法

4.1.2 Dormitory 类

1）定义私有属性: buildingName, roomNumber, students, rating

2）构造函数(Dormitory): 初始化属性，设置初始评分为 0，初始化学生列表

3）方法 addStudent: 将学生添加到学生列表

4）Getter 和 Setter 方法: 为属性提供访问和更新

类结构:

类定义: class Dormitory { ... }

属性定义: 私有属性 buildingName, roomNumber, students, rating

构造器: public Dormitory(...) { ... }

方法定义: 包括 addStudent，get 和 set 方法

4.1.3 DormitoryManagementSystem 类

1）定义私有属性: dormitories, students (映射)

2）构造函数: 初始化 dormitories 和 students 映射

3）方法 addStudent: 添加学生信息到映射

方法 assignStudentToDormitory: 分配学生到宿舍

方法 queryStudent: 查询学生信息

方法 addDormitory: 添加宿舍信息到映射

方法 queryDormitory: 查询宿舍信息

方法 findDormitoryOfStudent: 查找学生所在宿舍

方法 setDormitoryRating: 设置宿舍评分

方法 getDormitoryRating: 获取宿舍评分

方法 loadStudentsFromExcel: 从 Excel 文件加载学生数据
方法 saveStudentsToExcel: 将学生数据保存到 Excel 文件
方法 rankDormitoriesByRating: 根据评分对宿舍进行排名
类结构:

类定义: class DormitoryManagementSystem { ... }
属性定义: 私有映射 dormitories, students
构造器: public DormitoryManagementSystem() { ... }
方法定义: 实现了添加、查询、分配、导入/导出数据等多个方法

4.1.4 main 类

1）创建 DormitoryManagementSystem 实例
2）循环提供菜单选择
3）根据用户输入执行相应操作
添加学生
查询学生
添加宿舍
分配宿舍
导入/导出数据
设置/查询评分
显示评分排名
退出程序

类结构:
类定义: public class Main { ... }
主方法: public static void main(String[] args) { ... } 包含用户交互和选择逻辑

4.2 html 前端

定义 HTML 文档类型
开始 HTML 文档
开始头部
设置标题为 "宿舍评分系统"
开始样式定义
设置 body 样式（字体，背景颜色，边距，填充，布局）
设置.container 样式（宽度，边距，布局，对齐）
设置 h2 样式（颜色，文本对齐）
设置 form 样式（背景，填充，边框半径，阴影，宽度）
设置 label 和 input 通用样式（显示类型，宽度，边距）

设置 input[type="text"]样式（填充，边框，边框半径）
设置 input[type="submit"]样式（背景，颜色，填充，边框，边框半径，光标，宽度）
设置 input[type="submit"]:hover 样式（背景）
结束样式定义
结束头部
开始主体
创建一个容器 div
创建标题 h2，文本内容为 "宿舍评分系统"
创建表单
设置表单提交到 "/submit"，方法为 "post"
创建标签和输入框，用于输入宿舍号
创建标签和输入框，用于输入楼号
创建标签和输入框，用于输入分数
创建提交按钮，文本内容为 "添加"
结束主体
结束 HTML 文档

五、课程问题与知识点结合

5.1 this 的含义（将输入行参赋值给成员变量）

代码段如下：

```
    public Student(String id, String name, String college,
String grade, String phone) {
        this.id = id;
        this.name = name;
        this.college = college;
        this.grade = grade;
        this.phone = phone;
    }
```

其中，this.id = id 中 this 的含义为在 student 这个构造方法中，左侧 id 为变量名，这句代码的含义是将右侧输入形参赋给左侧的成员变量中。目的是为了区分成员变量和参数之间的名称冲突。

以下代码段同理，

```
public void setCollege(String college) { this.college = college; }
public void setGrade(String grade) { this.grade = grade; }
public void setPhone(String phone) { this.phone = phone; }
public void setId(String id) { this.id = id; }
public void setName(String name) { this.name = name; }
```

5.2 return 的含义（把值返回给调用方法的变量）

代码段如下：

```
public String getCollege() { return college; }
public String getGrade() { return grade; }
public String getPhone() { return phone; }
public String getId() { return id; }
public String getName() { return name; }
```

其中，return 会将变量的值赋给函数的调用者，类型有前面的数值类型决定。代码段如下，

```
public double getRating() {
    return rating;
}
Dormitory dorm = new Dormitory("Dorm 1", 101);
double currentRating = dorm.getRating();
```

return 嵌套的代码段同理，

```
public Student queryStudent(String studentId) {
    return students.get(studentId);
}
```

即查询然后返回值。

5.3 实例化（创建实例并分配内存或构造函数初始化对象）

代码段如下，

```
system.addStudent(new Student(studentId, studentName,
college, grade, phone));
```

其中，通过构造方法用 new 实例化对象。有参数的构造方法同理，代码段如下，

```
Scanner scanner = new Scanner(System.in);
```

5.4 对象访问控制（确定类成员的可见性、访问级别和权限）

代码段如下，

```
    private String id;
    private String name;
    private String college;
    private String grade;
    private String phone;

    public Student(String id, String name, String college,
String grade, String phone) {
        this.id = id;
        this.name = name;
        this.college = college;
        this.grade = grade;
        this.phone = phone;
    }
```

其中 private 和 public 为变量与方法的访问权限，private 只允许当前类的其它方法和变量对其访问，而 public 可以对全部类都开放。完善地设置权限很好地为代码进行封装与访问，可以有效避免程序出错，提高代码逻辑。

5.5 文件处理 I/O（通过 Apache POI 库实现文件访问）

代码段如下，

```
    FileInputStream file = new FileInputStream(filePath);
                Workbook workbook = new
```

```
XSSFWorkbook(file))
```

其中，通过 Apache POI 库中的类直接打开读取处理 Excel 文件，写入代码段同理，

```
Workbook workbook = new XSSFWorkbook();
            FileOutputStream fileOut = new
FileOutputStream(filePath))
workbook.write(fileOut);
```

5.6 异常处理（可能的错误情况进行处理并输出）

代码段如下，

```
try {

system.loadStudentsFromExcel(importPath);
                        System.out.println("学生数据已从
Excel 导入");
                    } catch (IOException e) {
                        System.out.println("导入 Excel 文件
时出错: " + e.getMessage());
                    }
```

其中，当 try 发生异常时会丢出并跳入 catch 中，输出错误同时处理完异常继续执行代码。

5.7 变量类型（定义数据存储和操作形式）

代码段如下，

```
public Dormitory(String buildingName, int roomNumber)
```

其中，string 为字符串类型，int 为整型，当字符类型设置不对是会报错，类型的正确设置可以节约时间度，代码有更完备逻辑。

5.8 子类与继承（对可能有相近或附属的类进行复用和扩展）

例如，

```
class Person {
    protected String name;
    public Person(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
class Student extends Person {
```

```
    private String studentId;
    public Student(String name, String studentId) {
        super(name); // 调用父类的构造方法
        this.studentId = studentId;
    }
    public String getStudentId() {
        return studentId;
    }
    public void setStudentId(String studentId) {
        this.studentId = studentId;
    }
}
```

其中，子类 Student 继承了父类 Person 的属性，同时扩展了 studentId 的属性，

继承可以极大地减少代码复杂度，增强复用，提高代码逻辑等。在大型系统中可以用人作为父类，不同的职业作为子类，将建筑物作为父类，不同的房屋类型作为子类。

六、项目体会与展望

在这一次项目过程中，我们学习到了一个系统的开发需要经历的过程，尤其是如何利用 java 语言去搭建出一个学生宿舍管理系统，java 是一种面向对象的语言，它与我们所掌握的其它编程语言不同，对象的封装、类的定义和继承、方法和变量的声明等基础语法很高效地让项目结构整体框架具有逻辑性。

我们根据所需要的功能和范围去编写类，然后再按照逻辑继续进行，让整个开发过程十分清晰。在基础的逻辑框架之上，我们也去扩展更多功能更强的 java 技术栈，学习更加高效的技巧。

同时，我们设想搭建出可视化的系统，在编写前端代码的时我们更具后端的主类方法进行 web 页面设计。在前后端互联的过程中，我们尝试了 Java Servlet 和 JavaServe Pages、Spring 框架和 Spring BOOT、Websocket 等，并设想接入 MySQL、Access 等数据库，并将最后的系统上传并部署到云端服务器并创建域名。

同时我们也想创新地利于算法知识结合到学生宿舍评分模块的开发中，我们对选择排序进行改进，也对冒泡算法改写为双向冒泡等，并利用 java 语言进行实现加入到系统中，也是我们从 Visual Basic 语言中学习到并进行应用改进的一个过程。

在这一过程中，我们遇到了各种各样的问题与报错，框架不熟悉、库文件缺失、版本不兼容、运行环境断开、网络协议错误、命令行报错以及电脑环境不完善等等。这一过程中，我们不断解决问题，理解技术点，掌握技术栈，最终我们没有能够将完整的一套学生宿舍管理系统完成，却也学习到了开发系统的经验与知识。

在之后，我们会继续根据后端框架完善我们的前端设计，不断加入新功能并测试，在系统足够完善时，我们将其接入数据库，达到大量数据高效处理的功能，并最终将其传至云端。在这一步之后，我们在有能力的情况下进行系统创新与改进，考虑是否进一步开发并用此系统去申请软件著作权，让我们的项目不断扩大。

在这一次的项目实验中，团队成员极大地提高了计算机应用与开发水平，计算机设计能力，资料检索能力，创新能力，思考问题的能力。一个大的项目离不开每一位团队成员的共同努力，我们也期望最终能够将项目完成落地，提升自己的能力。

附录

附录一

```
1.  package org.example;
2.
3.  import java.util.stream.Collectors;
4.  import java.util.ArrayList;
5.  import java.util.HashMap;
6.  import java.util.List;
7.  import java.util.Map;
8.  import java.util.Scanner;
9.  import org.apache.poi.ss.usermodel.*;
10. import org.apache.poi.xssf.usermodel.XSSFWorkbook;
11. import java.io.FileInputStream;
12. import java.io.FileOutputStream;
13. import java.io.IOException;
14.
15. // 学生类
16. class Student {
17.     private String id;
18.     private String name;
19.     private String college; // 新增学院属性
20.     private String grade;   // 新增年级属性
21.     private String phone;   // 新增电话属性
22.
23.     public Student(String id, String name, String college, String gra
    de, String phone) {
24.         this.id = id;
25.         this.name = name;
26.         this.college = college;
27.         this.grade = grade;
28.         this.phone = phone;
29.     }
30.
31.     // Getter 和 Setter 方法
32.     public String getCollege() { return college; }
33.     public String getGrade() { return grade; }
34.     public String getPhone() { return phone; }
35.     public String getId() { return id; }
36.     public String getName() { return name; }
37.
```

```
    public void setCollege(String college) { this.college = college; }
    public void setGrade(String grade) { this.grade = grade; }
    public void setPhone(String phone) { this.phone = phone; }
    public void setId(String id) { this.id = id; }
    public void setName(String name) { this.name = name; }
}

// 宿舍类
class Dormitory {
    private String buildingName;
    private int roomNumber;
    private List<Student> students;
    private double rating; // 新增宿舍评分属性

    public Dormitory(String buildingName, int roomNumber) {
        this.buildingName = buildingName;
        this.roomNumber = roomNumber;
        this.students = new ArrayList<>();
        this.rating = 0.0; // 初始评分为 0
    }

    // 添加学生到宿舍
    public void addStudent(Student student) {
        students.add(student);
    }
    public double getRating() {
        return rating;
    }

    public void setRating(double rating) {
        this.rating = rating;
    }

    // Getter 和 Setter
    public String getBuildingName() { return buildingName; }
    public int getRoomNumber() { return roomNumber; }
    public List<Student> getStudents() { return students; }
}

// 宿舍管理系统
class DormitoryManagementSystem {
    private Map<String, Dormitory> dormitories; // 用于存储宿舍信息
    private Map<String, Student> students; // 用于存储学生信息
```

```
81.
82.     public DormitoryManagementSystem() {
83.         dormitories = new HashMap<>();
84.         students = new HashMap<>();
85.     }
86.
87.     // 添加学生信息
88.     public void addStudent(Student student) {
89.         students.put(student.getId(), student);
90.     }
91.
92.     // 分配学生到宿舍
93.     public void assignStudentToDormitory(String studentId, String bui
    ldingName, int roomNumber) {
94.         Student student = students.get(studentId);
95.         if (student != null) {
96.             String key = buildingName + "-" + roomNumber;
97.             Dormitory dormitory = dormitories.get(key);
98.             if (dormitory != null) {
99.                 dormitory.addStudent(student);
100.                } else {
101.                    System.out.println("宿舍不存在！");
102.                }
103.            } else {
104.                System.out.println("学生不存在！");
105.            }
106.        }
107.
108.        // 查询学生信息
109.        public Student queryStudent(String studentId) {
110.            return students.get(studentId);
111.        }
112.
113.        // 添加宿舍信息
114.        public void addDormitory(Dormitory dormitory) {
115.            String key = dormitory.getBuildingName() + "-
    " + dormitory.getRoomNumber();
116.            dormitories.put(key, dormitory);
117.        }
118.
119.        // 查询宿舍信息
120.        public Dormitory queryDormitory(String buildingName, int roomN
    umber) {
121.            return dormitories.get(buildingName + "-" + roomNumber);
```

```
122.       }
123.
124.       public String findDormitoryOfStudent(String studentId) {
125.           for (Dormitory dormitory : dormitories.values()) {
126.               for (Student student : dormitory.getStudents()) {
127.                   if (student.getId().equals(studentId)) {
128.                       return dormitory.getBuildingName() + " 楼 " + d
    ormitory.getRoomNumber() + " 号室";
129.                   }
130.               }
131.           }
132.           return "学生未分配宿舍或不存在";
133.       }
134.       //宿舍评分
135.       public void setDormitoryRating(String buildingName, int roomNu
    mber, double rating) {
136.           String key = buildingName + "-" + roomNumber;
137.           Dormitory dormitory = dormitories.get(key);
138.           if (dormitory != null) {
139.               dormitory.setRating(rating);
140.               System.out.println("宿舍 " + key + " 的评分已更新
    为 " + rating);
141.           } else {
142.               System.out.println("宿舍不存在！");
143.           }
144.       }
145.
146.       // 获取宿舍评分
147.       public double getDormitoryRating(String buildingName, int room
    Number) {
148.           String key = buildingName + "-" + roomNumber;
149.           Dormitory dormitory = dormitories.get(key);
150.           if (dormitory != null) {
151.               return dormitory.getRating();
152.           } else {
153.               System.out.println("宿舍不存在！");
154.               return -1; // 表示宿舍不存在
155.           }
156.       }
157.       public void loadStudentsFromExcel(String filePath) throws IOEx
    ception {
158.           try (FileInputStream file = new FileInputStream(filePath);
159.                Workbook workbook = new XSSFWorkbook(file)) {
```

```
160.                 Sheet sheet = workbook.getSheetAt(0);
161.                 for (Row row : sheet) {
162.                     if (row.getRowNum() == 0) continue; // 跳过标题行
163.                     String id = row.getCell(0).getStringCellValue();
164.                     String name = row.getCell(1).getStringCellValue();
165.                     String college = row.getCell(2).getStringCellValue
    ();
166.                     String grade = row.getCell(3).getStringCellValue()
    ;
167.                     String phone = row.getCell(4).getStringCellValue()
    ;
168.                     addStudent(new Student(id, name, college, grade, p
    hone));
169.                 }
170.             }
171.         }
172.
173.         public void saveStudentsToExcel(String filePath) throws IOExce
    ption {
174.             try (Workbook workbook = new XSSFWorkbook();
175.                  FileOutputStream fileOut = new FileOutputStream(fileP
    ath)) {
176.                 Sheet sheet = workbook.createSheet("Students");
177.                 Row headerRow = sheet.createRow(0);
178.                 String[] columns = {"ID", "Name", "College", "Grade",
    "Phone", "Dormitory", "Dormitory Rating"};
179.                 for (int i = 0; i < columns.length; i++) {
180.                     headerRow.createCell(i).setCellValue(columns[i]);
181.                 }
182.
183.                 int rowNum = 1;
184.                 for (Student student : students.values()) {
185.                     Row row = sheet.createRow(rowNum++);
186.                     row.createCell(0).setCellValue(student.getId());
187.                     row.createCell(1).setCellValue(student.getName());
188.                     row.createCell(2).setCellValue(student.getCollege(
    ));
189.                     row.createCell(3).setCellValue(student.getGrade())
    ;
190.                     row.createCell(4).setCellValue(student.getPhone())
    ;
```

```
191.
192.                     // 获取学生的宿舍信息
193.                     String dormInfo = findDormitoryOfStudent(student.g
    etId());
194.                     row.createCell(5).setCellValue(dormInfo);
195.
196.                     // 获取宿舍评分
197.                     double dormRating = getDormitoryRatingForStudent(s
    tudent.getId());
198.                     row.createCell(6).setCellValue(dormRating);
199.                 }
200.                 workbook.write(fileOut);
201.             }
202.         }
203.
204.         private double getDormitoryRatingForStudent(String studentId)
    {
205.             for (Dormitory dormitory : dormitories.values()) {
206.                 for (Student student : dormitory.getStudents()) {
207.                     if (student.getId().equals(studentId)) {
208.                         return dormitory.getRating();
209.                     }
210.                 }
211.             }
212.             return 0.0; // 默认评分或学生未分配宿舍
213.         }
214.
215.         public void rankDormitoriesByRating() {
216.             List<Dormitory> sortedDormitories = dormitories.values().s
    tream()
217.                     .sorted((d1, d2) -> Double.compare(d2.getRating(),
     d1.getRating())) // 按评分降序排序
218.                     .collect(Collectors.toList());
219.
220.             System.out.println("宿舍评分排名：");
221.             int rank = 1;
222.             for (Dormitory dorm : sortedDormitories) {
223.                 System.out.println(rank + ". " + dorm.getBuildingName(
    ) + " 楼 " + dorm.getRoomNumber() + " 号室 - 评
    分: " + dorm.getRating());
224.                 rank++;
225.             }
226.         }
227.
```

```
228.   }
229.
230.   public class Main {
231.       public static void main(String[] args) {
232.           DormitoryManagementSystem system = new DormitoryManagement
    System();
233.           Scanner scanner = new Scanner(System.in);
234.
235.           while (true) {
236.               System.out.println("请选择操作：");
237.               System.out.println("1. 添加学生");
238.               System.out.println("2. 查询学生");
239.               System.out.println("3. 添加宿舍");
240.               System.out.println("4. 分配宿舍");
241.               System.out.println("5. 退出");
242.               System.out.println("6. 从 Excel 导入学生数据");
243.               System.out.println("7. 将学生数据导出到 Excel");
244.               System.out.println("8. 设置宿舍评分");
245.               System.out.println("9. 查询宿舍评分");
246.               System.out.println("10. 显示宿舍评分排名");
247.               System.out.print("请输入选项（1-10）：");
248.
249.
250.               int choice = scanner.nextInt();
251.               switch (choice) {
252.                   case 1:
253.                       // 添加学生
254.                       System.out.print("输入学生 ID：");
255.                       String studentId = scanner.next();
256.                       System.out.print("输入学生姓名：");
257.                       String studentName = scanner.next();
258.                       System.out.print("输入学生学院：");
259.                       String college = scanner.next();
260.                       System.out.print("输入学生年级：");
261.                       String grade = scanner.next();
262.                       System.out.print("输入学生电话：");
263.                       String phone = scanner.next();
264.                       system.addStudent(new Student(studentId, stude
    ntName, college, grade, phone));
265.                       break;
266.
267.                   case 2:
268.                       // 查询学生
269.                       System.out.print("输入学生 ID：");
```

```
                        String id = scanner.next();
                        Student student = system.queryStudent(id);
                        if (student != null) {
                            String studentInfo = "ID: " + student.getId() + ", 姓名: " + student.getName() +
                                    ", 学院: " + student.getCollege() + ", 年级: " + student.getGrade() +
                                    ", 电话: " + student.getPhone();
                            // 查询学生宿舍信息
                            String dormInfo = system.findDormitoryOfStudent(id);
                            studentInfo += ", 宿舍信息: " + dormInfo;
                            System.out.println(studentInfo);
                        } else {
                            System.out.println("学生不存在！");
                        }
                        break;

                    case 3:
                        // 添加宿舍
                        System.out.print("输入宿舍楼名：");
                        String building = scanner.next();
                        System.out.print("输入房间号：");
                        int room = scanner.nextInt();
                        system.addDormitory(new Dormitory(building, room));
                        break;

                    case 4:
                        // 分配宿舍
                        System.out.print("输入学生 ID：");
                        String sid = scanner.next();
                        System.out.print("输入宿舍楼名：");
                        String bname = scanner.next();
                        System.out.print("输入房间号：");
                        int rnum = scanner.nextInt();
                        system.assignStudentToDormitory(sid, bname, rnum);
                        break;

                    case 5:
                        // 退出程序
                        System.out.println("退出程序。");
                        return;
```

```
309.
310.                     case 6:
311.                         // 从 Excel 导入学生数据
312.                         System.out.print("输入 Excel 文件路径：");
313.                         String importPath = scanner.next();
314.                         try {
315.                             system.loadStudentsFromExcel(importPath);
316.                             System.out.println("学生数据已从 Excel 导入
    ");
317.                         } catch (IOException e) {
318.                             System.out.println("导入 Excel 文件时出
    错：" + e.getMessage());
319.                         }
320.                         break;
321.
322.                     case 7:
323.   // 将学生数据导出到 Excel
324.                         System.out.print("输入导出 Excel 文件的路径：
    ");
325.                         String exportPath = scanner.next();
326.                         try {
327.                             system.saveStudentsToExcel(exportPath);
328.                             System.out.println("学生数据已导出到
    Excel");
329.                         } catch (IOException e) {
330.                             System.out.println("导出到 Excel 文件时出
    错：" + e.getMessage());
331.                         }
332.                         break;
333.                     case 8:
334.                         // 设置宿舍评分
335.                         System.out.print("输入宿舍楼名：");
336.                         String buildingName = scanner.next();
337.                         System.out.print("输入房间号：");
338.                         int roomNumber = scanner.nextInt();
339.                         System.out.print("输入宿舍评分：");
340.                         double rating = scanner.nextDouble();
341.                         system.setDormitoryRating(buildingName, roomNu
    mber, rating);
342.                         break;
343.
344.                     case 9:
345.                         // 查询宿舍评分
```

```
                        System.out.print("输入宿舍楼名：");
                        buildingName = scanner.next();
                        System.out.print("输入房间号：");
                        roomNumber = scanner.nextInt();
                        double dormRating = system.getDormitoryRating(buildingName, roomNumber);
                        if (dormRating != -1) {
                            System.out.println("宿舍 " + buildingName + " 房间号 " + roomNumber + " 的评分是: " + dormRating);
                        }
                        break;
                    case 10:
                        // 显示宿舍评分排名
                        system.rankDormitoriesByRating();
                        break;
                }
            }
        }
    }
```

附录二

```
<!DOCTYPE html>
<html>
<head>
  <title>宿舍评分系统</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
      padding: 0;
      display: flex;
      flex-direction: column;
      align-items: center;
      height: 100vh;
    }
    .container {
      width: 100%;
      max-width: 400px;
      margin-top: 50px;
      display: flex;
      flex-direction: column;
```

```
      align-items: center;
    }
    h2 {
      color: #333;
      text-align: center;
    }
    form {
      background: white;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
      width: 100%;
    }
    label, input {
      display: block;
      width: 100%;
      margin-bottom: 10px;
    }
    input[type="text"] {
      padding: 8px;
      border: 1px solid #ddd;
      border-radius: 4px;
    }
    input[type="submit"] {
      background: #007bff;
      color: white;
      padding: 10px 15px;
      border: none;
      border-radius: 4px;
      cursor: pointer;
      width: 100%;
    }
    input[type="submit"]:hover {
      background: #0056b3;
    }
  </style>
</head>

<body>
<div class="container">
  <h2>宿舍评分系统</h2>
  <form action="/submit" method="post">
    <label for="dormitoryNumber">宿舍号:</label><br>
```

```
    <input type="text" id="dormitoryNumber" name="dormitoryNumber"><br>

    <label for="buildingNumber">楼号:</label><br>
    <input type="text" id="buildingNumber" name="buildingNumber"><br>

    <label for="score">分数:</label><br>
    <input type="text" id="score" name="score"><br>

    <input type="submit" value="添加">
  </form>
</div>
</body>
</html>
```