

软件开发实训报告

题目： 学生信息管理系统

学院（系）： 软件学院

年级专业： 23 级软件工程

小组名称： 添砖 Java

小组成员： 李*田 邢*炜 熊*

企业指导教师： 贺桑迪

企业教师职称： 高级工程师

小组成绩					
项目汇报效果 (10分)	项目展示效果 (45分)		工具使用情况 (10分)	报告质量和答辩效果 (5分)	小组得分
	项目完成质量 (40分)	创新意识 (5分)			

学生信息		组内成绩		小组得分	个人得分
学号	姓名	组内贡献 (25分)	合作意识 (5分)		

《专业综合课程设计》任务书 8

题目	[008] 学生信息管理系统
选题背景	<p>随着社会不断发展，高校学生日常管理涉及的各类数据信息越来越多，随之而来的管理水平也需要得到不断提升。与使用传统的人工手段完成学生日常管理任务和数据记录相比，暴露出来的问题越来越明显：例如工作效率低、容易重复出错、信息检索速度慢，以及教师不能充分掌握班级或学生的平均学习活动情况等，不利于监督学校正常的日常教学管理工作，鉴于此，学生管理系统的研发具有很强的必要性。</p>
功能需求	<p>首要任务是实现学生信息管理、课程安排和成绩管理的基础功能。这些功能是系统的核心。特别地，我们对系统的安全</p>

性给予了高度重视，尤其是针对 SQL 注入等常见安全漏洞。如果项目进度允许，我们将着手开发自动排课功能。

1. 用户登录与注册子模块：

用户登录模块负责验证用户身份，并允许注册用户访问系统的安全功能。通过验证用户提供的凭据，确保只有经过授权的用户可以使用系统。

功能列表：

用户名和密码验证：允许注册用户通过输入用户名和密码进行登录验证。

安全退出：提供安全的退出功能，清除用户会话信息。

错误处理：处理登录过程中可能出现的错误情况，如无效的用户名或密码。

2. 学生信息管理子模块

功能列表：

学生信息（课程 班级 年级 个人信息）增删改查

添加学生信息：系统允许用户输入或上传学生的基本信息，如姓名 班级等。

单个删除：用户可以选中单个学生信息进行删除，确保不会误删重要数据。

修改学生信息：用户可以随时修改学生的基本信息，如班级，年级等。

信息检索：提供多种检索方式，如按姓名、学号、班级等进行快速查找。

3. 课程成绩管理模块

教师管理类 教师界面（包含教师各个功能） 教师有权管理学生课程以及其分数

1) 类 CCourseManage CCourseAdd

课程管理类 选课界面 从数据库 C_Courseid 表中获得课程 id，再通过 id 从表 C_Course 中获得课程具体信息放置于下拉框中，勾选课程将 Course id 添加到学生信息中，传入数据库 C_Stu 表中。

2) 类 CCourseMackAdd

分数添加类 分数设置界面 通过学生 id 获得学生课程 id，再通过课程 id 获得具体课程展示与界面中，为具体课程添加分数将会将数据存于学生信息中，传入数据库 C_Stu 表中。

	<p>4. 操作日记子模块</p> <p>1) . 记录操作事件：系统应能够记录用户的登录、退出、进行数据增加、修改和删除等重要操作事件。每条日志记录应包含时间戳、操作者 ID、操作描述等信息。</p> <p>2) . 日志查看和搜索：系统管理员应能够查看和搜索日志记录。查看功能包括按时间段、操作类型、操作者等条件过滤和排序显示日志。</p> <p>3) . 导出和存储：可能需要支持将日志导出为文本文件，以便于存档、分析需要。</p> <p>4) . 保密性和完整性：日志应当保证其内容的保密性和完整性，不允许未经授权的修改或删除。</p> <p>5) . 实时记录与批量处理：系统应能够实时记录重要操作事件，同时也要支持批量处理，避免过多日志记录对系统性能造成不良影响。</p> <p>6) . 可扩展性和定制化：针对特定的业务需求，可能需要定制化的日志记录格式或内容。系统应具有一定的扩展性，以便根据实际需求进行功能增强和定制开发。</p>
技术要求	<p>1. 遵循 MVC 框架，统一 JDK1.8，使用 Swing 图形化界面，各成员分头行动，最后使用 IDEA 的 code with me 进行代码整合。</p> <p>2. 数据存储：使用数据库存储（MySQL）数据。统一使用 Navicat 数据库图形化界面，方便成员间交流。</p> <p>3. 事件处理：实现 Swing 组件的事件处理，例如按钮点击、菜单选择等操作，与业务逻辑的连接。</p> <p>4. 输入验证：对用户输入进行验证，确保数据的完整性和正确性。例如，姓名不能为空、年龄必须是数字等。</p> <p>5. 界面设计：设计清晰、用户友好的界面布局，使用户能够直观地操作和管理学生信息。</p> <p>6. 异常处理：考虑可能出现的异常情况，例如数据库连接问题等，并进行相应的错误弹窗处理。</p> <p>7. 文档和注释：编写清晰的代码注释，使其易于理解和维护。可以撰写简单的用户文档，说明系统的使用方法和功能。</p> <p>8. 命名严格规范：尤其是类名，方法名必须见名知义，防止团队成员间因为不知道方法名而无法调用。</p> <p>9. 测试和调试：对系统进行充分的测试，尤其是边界值</p>

	<p>和特殊值，例如成绩输入-1. 输入特殊符号\$, 输入 999 等不符合常规的数据的处理。</p> <p>10. 系统安全性：严格保证系统安全性，设立操作日志功能，方便追查使用者；使用 PreparedStament 防止 SQL 注入。</p>
补充说明	对图形化界面可添加 NCHU 的风格化处理。
提交物	软件调研报告，软件需求规格说明书，系统设计，软件测试报告，项目总结，讨论记录，工程文件（源代码、可执行程序、数据库文件）。
指导教师签字	

年 月 日

南昌航空大学实训报告评审意见表

指导教师评语：

成绩：_____

指导教师：_____

年 月 日

答辩小组评语：

成绩：_____

评阅人：_____

年 月 日

课程设计总成绩：

答辩小组成员签字：

年 月 日

摘 要

本项目名称为学生信息管理系统，是一个技术性的系统。本项目在开发过程中使用 MySQL，结合 Java Swing 来实现。此次实训我们制作一个学生信息管理系统。

随着社会不断发展，高校学生日常管理涉及的各类数据信息越来越多，随之而来的管理水平也需要得到不断提升。与使用传统的人工手段完成学生日常管理任务和数据记录相比，暴露出来的问题越来越明显：例如工作效率低、容易重复出错、信息检索速度慢，以及教师不能充分掌握班级或学生的平均学习活动情况等，不利于监督学校正常的日常教学管理工作，鉴于此，学生管理系统的研究与设计具有很强的必要性。

关键词：MySQL；技术性；Java Swing ；信息管理

目 录

摘 要.....	i
1 绪 论.....	1
1.1 课程目标.....	1
1.2 课程内容	1
1.3 课程教学模式	2
1.4 课程的使用对象	2
1.5 项目背景及目的	2
1.5.1 项目背景.....	2
1.5.2 目的和意义.....	2
2 需求分析.....	3
2.1 需求概述	3
2.2 可行性分析	3
2.3 功能模块描述	4
2.4 可行性分析	4
2.4.1 技术可行性分析.....	4
2.4.2 组织可行性分析.....	5
2.4.3 时间可行性分析.....	5
2.5 运行环境分析	5
2.5.1 硬件环境.....	5
2.5.2 软件环境.....	5
3 学生教师信息管理模块设计	14
3.1 功能描述	14
3.2 相关函数	14
3.3 主函数入口	19
3.4 关键代码	20
3.5 界面设计	25
3.5.1 学生信息管理主界面.....	25
4 年级·班级管理模块模块.....	28

4.1 功能描述	28
4.2 相关函数	29
4.3 主函数入口	36
4.4 关键代码	37
4.5 界面设计	43
5 选课·课程管理模块	45
5.1 功能描述	45
5.2 相关函数	45
5.3 主函数入口	48
5.4 关键代码	48
5.5 界面设计	52
6 用户登录注册模块	55
6.1 功能描述	55
6.2 相关函数	55
6.3 主函数入口	56
6.4 关键代码	56
6.5 界面设计	60
6.5.1 登录首页	60
7 成绩管理模块	63
7.1 功能描述	63
7.2 相关函数	63
7.3 主函数入口	64
7.4 关键代码	64
7.5 界面设计	66
8 测试	69
8.1 集成测试	69
总 结	70
参考文献资料	71

1 绪 论

1.1 课程目标

系统支持教师进行成绩录入、修改、删除和查询操作，确保数据的准确性和及时性。同时，学生可以通过系统方便地查询自己的考试成绩，帮助他们更好地了解 and 评估自己的学习表现。建立一个 C / S 系统，能够根据不同角色的权限，分别为学生、教师和教务处管理员提供功能，包括学生信息查询、发表留言、学生成绩录入、考试成绩查询以及学籍调整等操作。

1.2 课程内容

时间	课程设计内容
第 1 周第 1 次 (7.10)	国内外相关项目调研 系统技术 功能分析 学生信息管理系统相似系统分析
第 1 周第 2 次 (7.13)	学生管理系统功能模块分析 系统架构分析
第 1 周第 3 次 (7.15)	学生管理系统系统设计
第 1 周第 4 次 (7.20)	学生管理系统系统测试
第 1 周第 5 次 (7.24)	学生管理系统项目答辩

1.3 课程教学模式

项目驱动模式，强调问题的解决，以问题作为出发点，如何运用所学的知识能够解决问题，同时培养参训人员对技术点和技能点的综合应用能力。在实训过程中通过整体目标、分解目标及阶段里程碑的确定，提升参训人员的效率、提升参训人员的抗压力，不断提升参训人员灵活运用所学的各种专业知识及技术技能解决复杂综合实际问题的能力。

1.4 课程的使用对象

23 级软件工程学生

1.5 项目背景及目的

1.5.1 项目背景

开发背景：随着社会不断发展，高校学生日常管理涉及的各类数据信息越来越多，随之而来的管理水平也需要得到不断提升。与使用传统的人工手段完成学生日常管理任务和数据记录相比，暴露出来的问题越来越明显：例如工作效率低、容易重复出错、信息检索速度慢，以及教师不能充分掌握班级或学生的平均学习活动情况等，不利于监督学校正常的日常教学管理工作，鉴于此，学生管理系统的研究与设计具有很强的必要性。

1.5.2 目的和意义

开发目标：系统支持教师进行成绩录入、修改、删除和查询操作，确保数据的准确性和及时性。同时，学生可以通过系统方便地查询自己的考试成绩，帮助他们更好地了解和评估自己的学习表现。建立一个 C / S 系统，能够根据不同角色的权限，分别为学生、教师和教务处管理员提供功能，包括学生信息查询、发表留言、学生成绩录入、考试成绩查询以及学籍调整等操作。

2 需求分析

2.1 需求概述

学生管理系统旨在提供全面的学生信息管理，包括学生基本资料、课程选修、成绩记录、考勤管理及学术表现跟踪等功能，以便学校和教育机构能够有效管理和监控学生的学习进展和行为表现。

2.2 可行性分析

1. 效率提升

学生管理系统能够大大提升学校管理工作的效率。通过电子化管理学生信息、课程安排、成绩录入等，取代传统的手工记录和处理方式，节省大量时间和人力成本。

2. 数据集成与共享

学生管理系统能够实现数据的集成与共享。各个部门（如教务处、学生处、财务处）可以在同一个系统平台上实时获取和共享学生相关信息，避免信息孤岛和数据不一致性问题。

3. 精确的数据分析与报告

系统能够提供丰富的数据分析功能，帮助学校管理层和教师快速分析学生的学习情况、成绩趋势等。生成精确的统计报告和分析结果，支持学校决策和教学改进。

4. 提升服务质量

学生管理系统通过提供便捷的在线查询、信息反馈和服务支持，能够提升学生和家长的满意度。例如，学生可以方便地查看课程安排、成绩单等信息，家长能够及时了解学生的学习情况。

5. 安全性与隐私保护

系统可以设立不同层次的权限管理机制，确保敏感信息的安全性和隐私保护。只有授权人员才能访问和修改特定信息，有效防止信息泄露和非法访问。

6. 技术支持与可维护性

学生管理系统采用现代化的技术架构，能够提供稳定的运行环境和可靠的技术支持。系统的可维护性高，能够根据需求进行扩展和升级，保持系统的长期可用性。

7. 成本效益分析

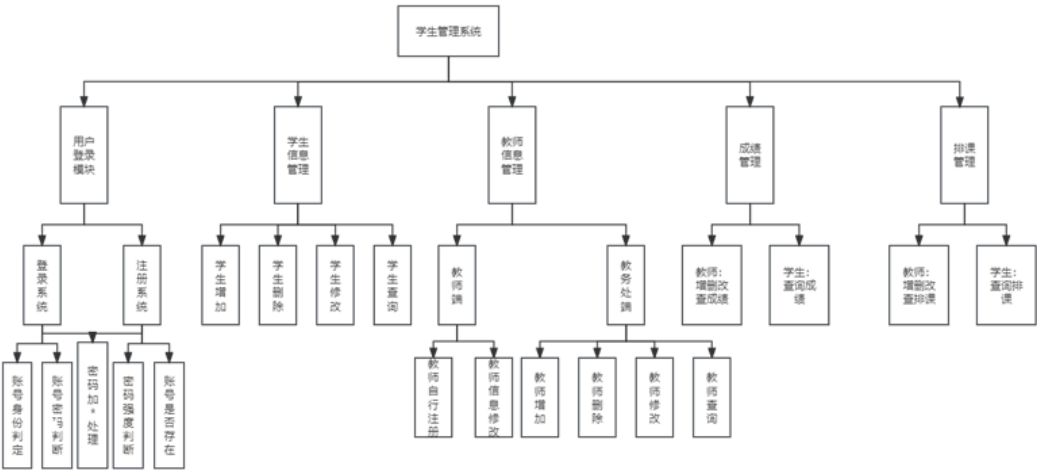
虽然学生管理系统的建设和维护可能需要一定的投入，但通过提高效率和服务质量，可以降低长期运营成本。系统的长期效益显著，可以视为长远投资。

8. 用户培训与接受度

为了确保系统的有效使用，需要进行相关用户培训和技术支持。系统设计应符合用户习惯和操作流程，提高用户接受度和使用效率。

2.3 功能模块描述

功能模块可大体分为 5 个子模块，分别是用户登录模块 学生信息管理 教师信息管理 成绩管理 排课管理。



2.4 可行性分析

2.4.1 技术可行性分析

1. 技术架构与设计

学生管理系统的技术架构应该能够支持系统的所有功能需求，如学生信息管理、课程安排、成绩录入与管理、通知发布等。良好的技术架构能够提供稳定的系统性能和扩展性，以应对未来可能的需求变化和功能扩展。

2. 数据管理与安全

系统需要有效地管理大量的学生数据和敏感信息，包括个人信息、成绩、课程安排等。因此，数据安全是至关重要的考虑因素。系统应该采用适当的数据加密、权限控制、备份和恢复机制，以确保数据的机密性、完整性和可用性。

3. 用户界面与用户体验

良好的用户界面设计能够提升系统的易用性和用户接受度。系统应该考虑用户的操作习惯和界面设计原则，使用户能够轻松地进行操作、查询信息和执行任务。

4. 平台兼容性与可扩展性

学生管理系统可能需要在多种平台上运行，包括不同操作系统和设备。因此，系统的开发应考虑跨平台兼容性，并且具备良好的可扩展性，以便未来能够集成新的技术和

功能模块。

5. 故障排除与技术支持

在系统实施和运行过程中，可能会面临技术问题和故障。因此，系统需要配备有效的故障排除机制和及时的技术支持服务，以确保系统的稳定性和可用性。

6. 集成与互操作性

学生管理系统通常需要与其他教育管理系统（如财务管理系统、图书管理系统）进行集成，以实现数据的共享和交换。因此，系统的设计应考虑到接口标准和互操作性，以便与现有系统和未来的系统进行无缝集成。

2.4.2 组织可行性分析

我们团队共 3 人，通过分析讨论，项目组长制定了合理的实施进度计划，合理进行任务分工，团队成员通过相互配合，发挥自己擅长的技能，能确保项目的顺利完成。

2.4.3 时间可行性分析

经过成员探讨，以及项目组长的合理分工，确保能够在要求的时间内实现所需要的功能，具有时间可行性。

2.5 运行环境分析

本系统兼容 windows 11 及当前主流操作系统，能够在低配置的环境下进行，可以让使用者减少开支。

2.5.1 硬件环境

服务器	最低配置	推荐配置
应用和数据库服务器	CPU: P4 2.0G	CPU: P4 2.8G
	Mem:512M	Mem:2G
	HD:40G	HD:120G

2.5.2 软件环境

	名称	版本	语种
操作系统	Windows	11	简体中文
数据库平台	MySQL	8.0	English

2.6.1 界面设计

学生信息管理系统

NCHU All Rights Reserved

用户名：

密 码：

登录类型

登录

关闭

用户名 输入框
varchar (12)

密码输入框
varchar (20)
用*代替

登录类型下拉框

可选择 管理员登录，
学生登录和老师登录，
对应不同的权限。

登录和关闭按钮。
点击登录，检测用户名和密码是否为
空，其次在与数据库连接，检验账户和
密码是否正确。

背景使用南昌航
空大学照片，并
降低其透明度，
防止遮挡页面

账户管理	信息管理	选课管理	成绩管理	关于软件
<div>欢迎文字 展示登录用户的相关信息</div>		<div>一个动态时钟 展示时间信息信息</div> <div>NCHU LOGO</div>		

账户管理 下设
重新登录
修改密码
用户管理
退出

信息管理 下设
学生信息管理
教师信息管理
课程信息管理
年级信息管理
班级信息管理

选课管理 下设
选课录入
选课总览

成绩管理 下设
成绩录入
成绩总览

页面同样以低透明度图
片作为背景

关于软件 下设
帮助
操作日记
开发人员

2.6.2 软件图标

该模块功能如图所示：

软件图标：我们选择了当前一位非常热门的明星作为软件图标。



2.6.3 信息管理模块设计

该模块功能如图所示：

各个信息的增删查改

学生信息管理

学生管理 支持批量删除

操作一体化

添加

修改

查询

删除

刷新

记录数:10

学号	姓名	性别	班级ID	班别名称
10004	陈罗志	男	1	核弹技术07-1
10015	陈皮	男	1	核弹技术07-1
10016	张三	女	2	核弹技术07-2
10017	李四	女	2	核弹技术07-2
10019	昭君	男	3	软件工程
10020	王军	男	4	信息技术
10021	马云	男	3	软件工程
10022	赵雅琴	男	5	计算机技术
15849	使得	男	1	核弹技术07-1

教师信息管理

教师管理

操作一体化

添加

修改

删除

刷新

记录数:4

授课教师ID	授课教师姓名	教师性别	教师学历
1	李老师	女	博士
3	吴老师	男	硕士
4	张老师	男	学士
6	薛仁贵	女	学士

教师快速添加

教师名称:

学历:

学士

教师性别:

☒ 男 ☐ 女

确定

取消

班级信息管理

 班级管理 ×

操作一体化

添加

修改

删除

刷新

记录数:5

班级ID	班级名称	所属年级	最大人数
1	核弹技术07-1	一年级	50
2	核弹技术07-2	一年级	50
3	软件工程	二年级	25
4	信息技术	二年级	55
5	计算机技术	三年级	50

班级快速添加

班级名称:

所属年级:

一年级

▼

班级最大人数:

确定

取消

年级信息管理

年级管理

×

操作一体化

添加

修改

删除

刷新

记录数:7

年级ID	年级名称
1	一年级
2	二年级
3	三年级
4	四年级
5	五年级
6	六年级
7	七年级

年级快速添加

年级名称:

确定

取消

课程信息管理

课程管理

×

操作一体化

添加

修改

查询

删除

刷新

记录数:9

课程ID	课程名称	课程学分	授课教师	开课时间
1	计算机技术	4	李老師	2011-05-28 12:38:58
2	核弹技术	3	李老師	2011-05-28 12:39:33
4	高等数学	8	吳老師	2011-05-28 12:40:30
8	高分子	5	李老師	2008-08-08 00:00:00
9	航空航天	6	吳老師	2009-09-06 00:00:00
11	软件工程导论	3	吳老師	2018-03-05 00:00:00
12	人工智能	4	張老師	2018-03-07 00:00:00
13	计算机网络	3	薛仁貴	2018-03-07 00:00:00
14	离散数学	4	吳老師	2018-03-07 00:00:00

课程快速开设

课程名称: 授课老师: 李老師

▼

确定

课程学分: 开课时间:

取消

课程介绍:

2.6.4 选课管理设计

该模块功能如图所示：

选课

根据年级班级选定学生进行选课

选课录入

年级选择: 一年级 班级选择: 核弹技术07-1

选课学生: 陈罗志 选修课程: 计算机技术

10004 号学生已选课程包括:

课程编号: 1 课程名称: 计算机技术
课程编号: 2 课程名称: 核弹技术

添加 取消

温馨提示: 所选各门课程成绩互相独立。

班级排课

通过班级给整个班级排课

批量排课

✕

年级选择：

一年级

▼

班级选择：

核弹技术07-1

▼

选修课程：

高等数学

▼

当前学生：

陈罗志

陈皮

使得

确认添加选课

选课总览

查看整个选课情况

3 课件结构设计

选课管理

×

选课信息一览表

按课程名称列举：

计算机技术

☐

学生选课门数：

还没选课学生

记录数：4

查询方式：

按ID

☐ 是否全表模糊

查询

课程ID	课程名称	学生ID	学生名称	教师ID	教师名称	课程成绩
1	计算机技术	10004	陈罗志	1	李老师	87
1	计算机技术	10017	李四	1	李老师	88
1	计算机技术	10019	昭君	1	李老师	0
1	计算机技术	10021	马云	1	李老师	98

操作面板

学生姓名：

所选课程：

计算机技术

打开选课管理面板

修改选课

删除选课

2.6.5 成绩管理设计

该模块功能如图所示：

通过年级班级选中学生及其课程进行打分

已选课程成绩录入

×

年级选择：

一年级

班级选择：

核弹技术07-1

选课学生：

陈罗志

已修课程：

计算机技术

☐ 开启过滤

87

确定录入

注：“开启过滤”只显示学生成绩还没被录入的课程。

成绩统计

统计课程学生整体得分情况



3 学生教师信息管理模块设计

3.1 功能描述

用户进入主界面之后，用户（管理员）可以查看学生教师信息，可以进行学生教师信息的增删查改，用户（教师）可以查看学生教师信息，修改学生信息，用户（学生）仅能看自身信息。

3.2 相关函数

```
private void initialize() {  
    this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);  
    this.setSize(658, 368);  
    this.setTitle("学生管理 支持批量删除");  
    this.setLocationRelativeTo(null);  
    //this.setModal(true);  
  
    JButton_student_add = new JButton();  
    JButton_student_add.setText("添加");  
    JButton_student_modify = new JButton();  
  
    JButton_student_modify.setText("修改");  
    JButton_student_query = new JButton();
```



```
jButton_student_query.setText("查询");
jButton_student_delete = new JButton();

jButton_student_delete.setText("删除");
jButton_student_flash = new JButton();

jButton_student_flash.setText("刷新");
jTable = new JTable();
jTable.setSelectionBackground(new Color(0, 255, 204));
jTable.setSelectionForeground(new Color(153, 0, 0));
jTable.setCursor(new Cursor(Cursor.HAND_CURSOR));

jLabel_Row = new JLabel();
jLabel_Row.setSize(25, 30);

jJToolBarBar = new JToolBar();
jJToolBarBar.add(jButton_student_add);
jJToolBarBar.add(jButton_student_modify);
jJToolBarBar.add(jButton_student_query);
jJToolBarBar.add(jButton_student_delete);
jJToolBarBar.add(jButton_student_flash);
jJToolBarBar.add(jLabel_Row);

jScrollPane = new JScrollPane();
jScrollPane.setViewportView(jTable);
jContentPane = new JPanel();
jContentPane.setLayout(new BorderLayout());
jContentPane.setBorder(BorderFactory.createTitledBorder("操作一体化"));
jContentPane.add(jScrollPane, BorderLayout.CENTER);
jContentPane.add(jJToolBarBar, BorderLayout.NORTH);
this.setContentPane(jContentPane);

btnListener btl = new btnListener();
jButton_student_add.addActionListener(btl);
jButton_student_modify.addActionListener(btl);
```

```
jButton_student_query.addActionListener(btl);
jButton_student_delete.addActionListener(btl);
jButton_student_flash.addActionListener(btl);
}

private void initialize() {
    jLabel_teach_sex = new JLabel();
    jLabel_teach_sex.setText("教师性别: ");
    jLabel_teach_sex.setBounds(new Rectangle(23, 71, 88, 26));
    jLabel_teach_name = new JLabel();
    jLabel_teach_name.setText("教师名称: ");
    jLabel_teach_name.setBounds(new Rectangle(23, 25, 88, 28));
    jLabel_teach_educatioon = new JLabel();
    jLabel_teach_educatioon.setText("学历: ");
    jLabel_teach_educatioon.setBounds(new Rectangle(280, 20, 73, 30));
    this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    this.setSize(550, 500);
    this.setTitle("教师管理");
    //this.setModal(true);
    this.setLocationRelativeTo(null);

    jButton_course_add = new JButton();
    jButton_course_add.setText("添加");
    jButton_course_query = new JButton();
    jButton_course_query.setText("修改");
    jButton_course_delete = new JButton();
    jButton_course_delete.setText("删除");
    jButton_course_flash = new JButton();
    jButton_course_flash.setText("刷新");

    jRadioButton_boy = new JRadioButton("男", true);
    jRadioButton_boy.setBounds(new Rectangle(119, 69, 59, 28));

    jRadioButton_girl = new JRadioButton("女", false);
    jRadioButton_girl.setBounds(new Rectangle(179, 69, 56, 30));
```

```
jRadioButton_boy.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        teach_sex = "男";
    }
});
jRadioButton_girl.addActionListener(new
java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent e) {
        teach_sex = "女";
    }
});
```

```
gettsButtonGroup(); // 加入按钮组
```

```
jButton_ok = new JButton();
jButton_ok.setBounds(new Rectangle(242, 65, 94, 29));
jButton_ok.setText("确定");

jButton_cancel=new JButton();
jButton_cancel.setBounds(new Rectangle(347, 66, 88, 27));
jButton_cancel.setText("取消");
```

```
jLabel_course_counts = new JLabel();
jLabel_course_counts.setSize(20, 20);
jJToolBarBar = new JToolBar();
jJToolBarBar.setBounds(new Rectangle(5, 21, 714, 34));
jJToolBarBar.add(jButton_course_add);
jJToolBarBar.add(jButton_course_query);
jJToolBarBar.add(jButton_course_delete);
jJToolBarBar.add(jButton_course_flash);
jJToolBarBar.add(jLabel_course_counts);
```

```
jTable = new JTable();
jScrollPane = new JScrollPane();
jScrollPane.setBounds(new Rectangle(20, 71, 450, 250));
jScrollPane.setViewportView(jTable);

jTextField_teach_name = new JTextField();
jTextField_teach_name.setBounds(new Rectangle(116, 26, 122,
26));

jComboBox_teach_education = new JComboBox();
jComboBox_teach_education.setBounds(new Rectangle(326, 18, 122,
30));

jPanel = new JPanel();
jPanel.setLayout(null);
jPanel.setBounds(new Rectangle(22, 332, 500, 106));
jPanel.setBorder(BorderFactory.createTitledBorder("教师快速添加"));

jPanel.add(jLabel_teach_name, null);
jPanel.add(jLabel_teach_sex, null);
jPanel.add(jTextField_teach_name, null);
jPanel.add(jComboBox_teach_education, null);
jPanel.add(jLabel_teach_educatioon, null);
jPanel.add(jButton_ok, null);
jPanel.add(jButton_cancel, null);
jPanel.add(jRadioButton_boy, null);
jPanel.add(jRadioButton_boy, null);
jPanel.add(jRadioButton_boy, null);

jPanel.add(jRadioButton_girl, null);

jContentPane = new JPanel();
jContentPane.setLayout(null);
jContentPane.setBorder(BorderFactory.createTitledBorder("操作一体化"));
jContentPane.add(jScrollPane, null);
```

```
jContentPane.add(jToolBarBar, null);
jContentPane.add(jPanel, null);
this.setContentPane(jContentPane);
jContentPane.add(jScrollPane, null);

this.setContentPane(jContentPane);

}
```

3.3 主函数入口

```
public StudentManage() {
    super();
    initialize();
    initData();
}

public TeacherManage() {
    super();
    initialize();
    initData();
    btnListener btn = new btnListener();
    jButton_course_add.addActionListener(btn);
    jButton_course_query.addActionListener(btn);
    jButton_course_delete.addActionListener(btn);
    jButton_course_flash.addActionListener(btn);
    jButton_ok.addActionListener(btn);
    jButton_cancel.addActionListener(btn);

    jTable.getSelectionModel().addListSelectionListener(new
tableListener());
}
```

3.4 关键代码

增删查改的实现

```
private class btnListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        StudentDao sd = new StudentDao();
        if (e.getSource() == jButton_student_add) {
            //学生添加
            StudentAdd sa = new StudentAdd(false, -1);
            sa.setVisible(true);
            flashData();
        }
        else if (e.getSource() == jButton_student_modify) {
            //学生修改
            if (jTable.getSelectedRow() != -1) { // 选择行数如果非空
                int stu_id =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
0).toString());
                StudentAdd sa = new StudentAdd(true, stu_id);
                sa.setVisible(true);
                flashData();
            } else {
                JOptionPane.showMessageDialog(null, "请选择要修改的行!");
            }
        }
        else if (e.getSource() == jButton_student_delete) { // 支持多
行删除
            if (jTable.getSelectedRow() != -1) {
                int stu_id =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
0).toString());
                sd.deleteListByStuId(stu_id);

                JOptionPane.showMessageDialog(null, "删除成功!");

                model.removeRow(jTable.getSelectedRow());
            }
        }
    }
}
```

3 课件结构设计

```
        //flashData();
    } else {
        JOptionPane.showMessageDialog(null, "请选择要删除的行!");
    }
} else if (e.getSource() == jButton_student_query) {

    int stu_id;
    try {
        stu_id = Integer.parseInt(JOptionPane.showInputDialog("按
学号查询, 请输入要查询的学号: "));
    } catch (Exception e2) {
        // TODO: handle exception
        return;
    }
    student_lists = sd.getLists(true, stu_id);
    flashData();

} else if (e.getSource() == jButton_student_flash) {
    initData();
}

}

}

public class btnListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        TeacherDao td = new TeacherDao();
        if (e.getSource() == jButton_course_add) {

            SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
            String course_begin_time = sdf.format(new Date());
            jTextField_teach_name.setEditable(true);
            jTextField_teach_name.setEditable(true);
            jButton_ok.setEnabled(true);

        }
    }
}
```

```

else if (e.getSource() == jButton_course_delete) {
    if (jTable.getSelectedRow() != -1) {
        int teach_id =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
0).toString());
        td.deleteListByTeachId(teach_id);
        JOptionPane.showMessageDialog(null, "删除成功!");
        //pojo.removeRow(jTable.getSelectedRow());
        //数据量不是很多, 感觉重新刷新一次好点
        counts = td.getLists(false, -1).size();
        initData();
    } else {
        JOptionPane.showMessageDialog(null, "请选择要删除的行!");
    }
}

else if (e.getSource() == jButton_course_query) {
    //int teach_id =
Integer.parseInt(JOptionPane.showInputDialog("按学号查询, 请输入要查询的
学号: "));
    //teacher_lists = td.getLists(true, teach_id);
    //flashData();
    if (jTable.getSelectedRow() != -1) {
        String teachName =
jTable.getValueAt(jTable.getSelectedRow(), 1).toString();
        String teachSex =
jTable.getValueAt(jTable.getSelectedRow(), 2).toString();
        String teachEducation =
jTable.getValueAt(jTable.getSelectedRow(), 3).toString();
        if(null == teachName || "".equals(teachName)){
            JOptionPane.showMessageDialog(null, "不存在该数据, 请刷新
面板!");
            return;
        }
        jTextField_teach_name.setText(teachName);
        jTextField_teach_name.setEditable(true);
        for(int i=0; i <
jComboBox_teach_education.getItemCount(); i++){
            String ted = (String)

```

```

jComboBox_teach_education.getItemAt(i);
    if(ted.equals(teachEducation)){
        jComboBox_teach_education.setSelectedIndex(i);
        break;
    }
}

if ("男".equals(teachSex)) {
    jRadioButton_boy.setSelected(true);
} else {
    jRadioButton_girl.setSelected(true);
}

jButton_ok.setText("确定修改");
jButton_ok.setEnabled(true);
}else {
    JOptionPane.showMessageDialog(null, "请选择要修改的行!");
}

}

else if (e.getSource() == jButton_course_flash) {
    initData();
}

else if (e.getSource() == jButton_ok) {
    String teach_name = jTextField_teach_name.getText();
    if ("确定修改".equals(jButton_ok.getText())) {
        int teacher_id =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
0).toString());
        String teach_sex = jRadioButton_boy.isSelected() ? "男" :
"女";
        String teach_education =
jComboBox_teach_education.getSelectedItem().toString();
        if (teach_education != null) {
            if (td.modifyTeacher(teacher_id, teach_name, teach_sex,
teach_education))
            {
                JOptionPane.showMessageDialog(null, "修改成功");
                initData();
            }else{

```

```
JOptionPane.showMessageDialog(null, "修改失败");
    }
}
return;
}
//String teach_education =
teacher_lists.get(jComboBox_teach_education.getSelectedIndex()).get
Education();
String teach_education =
jComboBox_teach_education.getSelectedItem().toString();
if(td.addTeacher(teach_name, teach_sex, teach_education))
{JOptionPane.showMessageDialog(null, "添加成功");
}else
    JOptionPane.showMessageDialog(null, "添加失败");
teacher_lists = td.getLists(false, -1);
flashData();
}
else if(e.getSource()==jButton_cancel){
    initData();
}
}
}
```

3.5 界面设计

3.5.1 学生信息管理主界面

学生管理 支持批量删除

操作一体化

添加

修改

查询

删除

刷新

记录数:9

学号	姓名	性别	班级ID	班别名称
10004	陈罗志	男	1	核弹技术07-1
10015	陈皮	男	1	核弹技术07-1
10016	张三	女	2	核弹技术07-2
10017	李四	女	2	核弹技术07-2
10019	昭君	男	3	软件工程
10020	王军	男	4	信息技术
10021	马云	男	3	软件工程
10022	赵雅琴	男	5	计算机技术
15849	使得	男	1	核弹技术07-1

3.5.2 教师信息管理主界面

 教师管理 ×

操作一体化

添加

修改

删除

刷新

记录数:4

授课教师ID	授课教师姓名	教师性别	教师学历
1	李老师	女	博士
3	吴老师	男	硕士
4	张老师	男	学士
6	薛仁贵	女	学士

教师快速添加

教师名称：

学历：

学士

教师性别：☒ 男 ☐ 女

确定

取消

3.5.3 功能界面弹窗

 添加学生 ×

学号:

姓名:

性别: ☒ 男 ☐ 女


班级: ▼

教师快速添加

教师名称: 学历: ▼

教师性别: ☒ 男 ☐ 女

输入 ×

 按学号查询, 请输入要查询的学号:

4 年级·班级管理模块模块

4.1 功能描述

用户进入主界面之后，用户可以查看年级与班级信息。

年级管理

×

操作一体化

添加

修改

删除

刷新

记录数:7

年级ID	年级名称
1	一年级
2	二年级
3	三年级
4	四年级
5	五年级
6	六年级
7	七年级

年级快速添加

年级名称:

确定

取消

班级管理

×

操作一体化

添加

修改

删除

刷新

记录数:5

班级ID	班级名称	所属年级	最大人数
1	核弹技术07-1	一年级	50
2	核弹技术07-2	一年级	50
3	软件工程	二年级	25
4	信息技术	二年级	55
5	计算机技术	三年级	50

班级快速添加

班级名称:

所属年级:

一年级 ▼

班级最大人数:

确定

取消

4.2 相关函数

```
private void initialize() {
//    jLabel_teach_sex = new JLabel();
//    jLabel_teach_sex.setText("教师性别: ");
//    jLabel_teach_sex.setBounds(new Rectangle(23, 71, 88, 26));
jLabel_teach_name = new JLabel();
jLabel_teach_name.setText("年级名称: ");
jLabel_teach_name.setBounds(new Rectangle(23, 25, 88, 28));
//    jLabel_teach_educatioon = new JLabel();
//    jLabel_teach_educatioon.setText("教师学历: ");
}
```

```
//      jLabel_teach_educatioon.setBounds(new Rectangle(245, 20, 73,
30));

this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
this.setSize(497, 479);
this.setTitle("年级管理");
//this.setModal(true);
this.setLocationRelativeTo(null);
    JButton_course_add = new JButton();
    JButton_course_add.setText("添加");
    JButton_course_query = new JButton();
    JButton_course_query.setText("修改");
    JButton_course_delete = new JButton();
    JButton_course_delete.setText("删除");
    JButton_course_flash = new JButton();
    JButton_course_flash.setText("刷新");

//      jRadioButton_boy = new JRadioButton("男",true);
//      jRadioButton_boy.setBounds(new Rectangle(119, 69, 59, 28));
//
//
//      jRadioButton_girl = new JRadioButton("女",false);
//      jRadioButton_girl.setBounds(new Rectangle(179, 69, 56, 30));
//
//      jRadioButton_boy.addActionListener(new
java.awt.event.ActionListener() {
//          public void actionPerformed(java.awt.event.ActionEvent e)
//          {
//              teach_sex = "男";
//          }
//      });
//      jRadioButton_girl.addActionListener(new
java.awt.event.ActionListener() {
//          public void actionPerformed(java.awt.event.ActionEvent e)
//          {
//              teach_sex = "女";
//          }
//      });
```



```
//gettsButtonGroup(); // 加入按钮组
```

```
jButton_ok = new JButton();  
jButton_ok.setBounds(new Rectangle(100, 65, 94, 29));  
jButton_ok.setText("确定");  
  
jButton_cancel=new JButton();  
jButton_cancel.setBounds(new Rectangle(247, 65, 94, 29));  
jButton_cancel.setText("取消");
```

```
jLabel_course_counts = new JLabel();  
jLabel_course_counts.setSize(20, 20);  
jJToolBarBar = new JToolBar();  
jJToolBarBar.setBounds(new Rectangle(5, 21, 714, 34));  
jJToolBarBar.add(jButton_course_add);  
jJToolBarBar.add(jButton_course_query);  
jJToolBarBar.add(jButton_course_delete);  
jJToolBarBar.add(jButton_course_flash);  
jJToolBarBar.add(jLabel_course_counts);
```

```
jTable = new JTable();  
jScrollPane = new JScrollPane();  
jScrollPane.setBounds(new Rectangle(20, 71, 450, 250));  
jScrollPane.setViewportView(jTable);
```

```
jTextField_teach_name = new JTextField();  
jTextField_teach_name.setBounds(new Rectangle(116, 26, 122,
```

```
26));
```

```
jComboBox_teach_education = new JComboBox();  
jComboBox_teach_education.setBounds(new Rectangle(326, 18,
```

```
122, 30));
```

```
        jPanel = new JPanel();
        jPanel.setLayout(null);
        jPanel.setBounds(new Rectangle(22, 332, 456, 106));
        jPanel.setBorder(BorderFactory.createTitledBorder("年
级快速添加"));
        jPanel.add(jLabel_teach_name, null);
//        jPanel.add(jLabel_teach_sex, null);
        jPanel.add(jTextField_teach_name, null);
//        jPanel.add(jComboBox_teach_education, null);
//        jPanel.add(jLabel_teach_educatioon, null);
        jPanel.add(jButton_ok, null);
        jPanel.add(jButton_cancel, null);
//        jPanel.add(jRadioButton_boy, null);
//        jPanel.add(jRadioButton_boy, null);
//        jPanel.add(jRadioButton_boy, null);
//
//        jPanel.add(jRadioButton_girl, null);

        jContentPane = new JPanel();
        jContentPane.setLayout(null);

        jContentPane.setBorder(BorderFactory.createTitledBorder("操作一体化
"));
        jContentPane.add(jScrollPane, null);
        jContentPane.add(jJToolBarBar, null);
        jContentPane.add(jPanel, null);
        this.setContentPane(jContentPane);
        jContentPane.add(jScrollPane, null);

        this.setContentPane(jContentPane);

    }
}
```

```
private void initialize() {
    jLabel_teach_sex = new JLabel();
    jLabel_teach_sex.setText("班级最大人数: ");
    jLabel_teach_sex.setBounds(new Rectangle(17, 71, 110, 26));
    jLabel_teach_name = new JLabel();
    jLabel_teach_name.setText("班级名称: ");
    jLabel_teach_name.setBounds(new Rectangle(17, 25, 88, 28));
    jLabel_teach_educatioon = new JLabel();
    jLabel_teach_educatioon.setText("所属年级: ");
    jLabel_teach_educatioon.setBounds(new Rectangle(245, 25, 90,
30));
    this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    this.setSize(497, 479);
    this.setTitle("班级管理");
    //this.setModal(true);
    this.setLocationRelativeTo(null);
    jButton_course_add = new JButton();
    jButton_course_add.setText("添加");
    jButton_course_query = new JButton();
    jButton_course_query.setText("修改");
    jButton_course_delete = new JButton();
    jButton_course_delete.setText("删除");
    jButton_course_flash = new JButton();
    jButton_course_flash.setText("刷新");

    jButton_ok = new JButton();
    jButton_ok.setBounds(new Rectangle(250, 65, 90, 29));
    jButton_ok.setText("确定");

    jButton_cancel=new JButton();
    jButton_cancel.setBounds(new Rectangle(347, 65, 70, 29));
```

```
jButton_cancel.setText("取消");

jLabel_course_counts = new JLabel();
jLabel_course_counts.setSize(20, 20);
jJToolBarBar = new JToolBar();
jJToolBarBar.setBounds(new Rectangle(5, 21, 714, 34));
jJToolBarBar.add(jButton_course_add);
jJToolBarBar.add(jButton_course_query);
jJToolBarBar.add(jButton_course_delete);
jJToolBarBar.add(jButton_course_flash);
jJToolBarBar.add(jLabel_course_counts);

jTable = new JTable();
jScrollPane = new JScrollPane();
jScrollPane.setBounds(new Rectangle(20, 71, 450, 250));
jScrollPane.setViewportViewView(jTable);

jTextField_teach_name = new JTextField();
jTextField_teach_name.setBounds(new Rectangle(116, 26, 122,
30));

jTextField_max_num = new JTextField();
jTextField_max_num.setBounds(new Rectangle(119, 69, 70, 28));

jComboBox_teach_education = new JComboBox();
jComboBox_teach_education.setBounds(new Rectangle(326, 26,
122, 30));

jPanel = new JPanel();
jPanel.setLayout(null);
jPanel.setBounds(new Rectangle(22, 332, 456, 106));
jPanel.setBorder(BorderFactory.createTitledBorder("班级快速添加"));
jPanel.add(jLabel_teach_name, null);
```

```
jPanel.add(jLabel_teach_sex, null);
jPanel.add(jTextField_teach_name, null);
jPanel.add(jComboBox_teach_education, null);
jPanel.add(jLabel_teach_educatioon, null);
jPanel.add(jTextField_max_num, null);
jPanel.add(jButton_ok, null);
jPanel.add(jButton_cancel, null);
//      jPanel.add(jRadioButton_boy, null);
//      jPanel.add(jRadioButton_boy, null);
//      jPanel.add(jRadioButton_boy, null);
//
//      jPanel.add(jRadioButton_girl, null);

jContentPane = new JPanel();
jContentPane.setLayout(null);

jContentPane.setBorder(BorderFactory.createTitledBorder("操作一体化
"));

jContentPane.add(jScrollPane, null);
jContentPane.add(jJToolBarBar, null);
jContentPane.add(jPanel, null);
this.setContentPane(jContentPane);
jContentPane.add(jScrollPane, null);

this.setContentPane(jContentPane);

}
```

4.3 主函数入口

```
public GradeManage() {
    super();
    initialize();
    initData();
    btnListener btn = new btnListener();
    jButton_course_add.addActionListener(btn);
    jButton_course_query.addActionListener(btn);
    jButton_course_delete.addActionListener(btn);
    jButton_course_flash.addActionListener(btn);
    jButton_ok.addActionListener(btn);
    jButton_cancel.addActionListener(btn);

    jTable.getSelectionModel().addListSelectionListener(new
tableListener());
}

public ClassManage() {
    super();
    initialize();
    initData();
    btnListener btn = new btnListener();
    jButton_course_add.addActionListener(btn);
    jButton_course_query.addActionListener(btn);
    jButton_course_delete.addActionListener(btn);
    jButton_course_flash.addActionListener(btn);
    jButton_ok.addActionListener(btn);
    jButton_cancel.addActionListener(btn);

    jTable.getSelectionModel().addListSelectionListener(new
tableListener());
}
```

4.4 关键代码

```
public class btnListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        GradeDao gd = new GradeDao();
        if (e.getSource() == jButton_course_add) {

            //SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
            //String course_begin_time = sdf.format(new Date());
            //jTextField_teach_name.setEditable(true);
            jTextField_teach_name.setEditable(true);
            jButton_ok.setEnabled(true);
            jTextField_max_num.setEditable(true);
        }
        else if (e.getSource() == jButton_course_delete) {
            if (jTable.getSelectedRow() != -1) {
                int class_id =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
0).toString());
                if(JOptionPane.showConfirmDialog(null, "班级下可能有多个学
生, 请谨慎删除, 否则可能导致程序出错, 是否确定删除? ") ==
JOptionPane.OK_OPTION) {
                    ClassDao cd = new ClassDao();
                    if(cd.deleteListByClassId(class_id)){
                        JOptionPane.showMessageDialog(null, "删除成功!");
                        initData();
                    }else{
                        JOptionPane.showMessageDialog(null, "删除失败, 该班级下
有学生, 请先删除班级!");
                    }
                }
            }
        } else {
            JOptionPane.showMessageDialog(null, "请选择要删除的行!");
        }
    }
    else if (e.getSource() == jButton_course_query) {
        if (jTable.getSelectedRow() != -1) {
```

```
ClassDao cd = new ClassDao();
int class_id =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
0).toString());
String className =
jTable.getValueAt(jTable.getSelectedRow(), 1).toString();
String gradeName =
jTable.getValueAt(jTable.getSelectedRow(), 2).toString();
Integer maxNumInteger =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
3).toString());
if(null == className || "".equals(className)){
    JOptionPane.showMessageDialog(null, "不存在该数据，请刷新
面板!");
    return;
}
jTextField_teach_name.setText(className);
jTextField_teach_name.setEditable(true);
jTextField_max_num.setText(maxNumInteger.toString());
jTextField_max_num.setEditable(true);
for(int i=0;i <
jComboBox_teach_education.getItemCount();i++){
    GradeModel gModel = (GradeModel)
jComboBox_teach_education.getItemAt(i);
    if(gModel.getGrade_name().equals(gradeName)){
        jComboBox_teach_education.setSelectedIndex(i);
        break;
    }
}
jButton_ok.setText("确定修改");
jButton_ok.setEnabled(true);

} else {
    JOptionPane.showMessageDialog(null, "请选择要修改的行!");
}

//grade_lists = td.getLists(true, teach_id);
flashData();
```



```

    }
    else if (e.getSource() == jButton_course_flash) {
        initData();
    }
    else if (e.getSource() == jButton_ok) {
        if ("确定修改".equals(jButton_ok.getText())) {
            int class_id =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
0).toString());
            String class_name = jTextField_teach_name.getText();
            Integer max_num ;
            try {
                max_num =
Integer.parseInt(jTextField_max_num.getText());
            } catch (Exception e2) {
                JOptionPane.showMessageDialog(null, "最大人数请输入数
字!");
                return;
            }
            GradeModel gradeModelTmp = (GradeModel)
jComboBox_teach_education.getSelectedItem();
            if (gradeModelTmp != null) {
                ClassDao cd = new ClassDao();
                if (cd.modifyClass(class_id, class_name,
gradeModelTmp.getGrade_id(), max_num))
                {
                    JOptionPane.showMessageDialog(null, "修改成功");
                    initData();
                } else {
                    JOptionPane.showMessageDialog(null, "修改失败");
                }
            }
            return;
        }
        String class_name = jTextField_teach_name.getText();
        if ("".equals(class_name)) {
            JOptionPane.showMessageDialog(null, "请输入名称!");

```

```

        return;
    }
    Integer maxNum ;
    try {
        maxNum =
Integer.parseInt(jTextField_max_num.getText().toString());
    } catch (Exception e2) {
        JOptionPane.showMessageDialog(null, "最大人数请输入数字!");
        return;
    }
    GradeModel gradeModelTmp = (GradeModel)
jComboBox_teach_education.getSelectedItem();
    if(gradeModelTmp != null){
        ClassDao cd = new ClassDao();
        if(cd.addClass(class_name, gradeModelTmp.getGrade_id(),
maxNum))
        {
            JOptionPane.showMessageDialog(null, "添加成功");
            initData();
        }else{
            JOptionPane.showMessageDialog(null, "添加失败");
        }
    }

    //grade_lists = td.getLists(false, -1);
}
else if(e.getSource()==jButton_cancel){
    initData();
}
}
}

public class btnListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        GradeDao gd = new GradeDao();
        if (e.getSource() == jButton_course_add) {

            //SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");

```

```

        //String course_begin_time = sdf.format(new Date());
        //jTextField_teach_name.setEditable(true);
        jTextField_teach_name.setEditable(true);
        jButton_ok.setEnabled(true);

    }

    else if (e.getSource() == jButton_course_delete) {
        if (jTable.getSelectedRow() != -1) {
            int grade_id =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
0).toString());

            if(JOptionPane.showConfirmDialog(null, "年级下可能有多个班
级, 请谨慎删除, 否则可能导致程序出错, 是否确定删除? ") ==
JOptionPane.OK_OPTION) {
                if(gd.deleteListByGradeId(grade_id)){
                    JOptionPane.showMessageDialog(null, "删除成功!");
                    initData();
                }else{
                    JOptionPane.showMessageDialog(null, "删除失败, 该年级下
有班级, 请先删除班级!");
                }
            }
        } else {
            JOptionPane.showMessageDialog(null, "请选择要删除的行!");
        }
    }

    else if (e.getSource() == jButton_course_query) {
        if (jTable.getSelectedRow() != -1) {
            int grade_id =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
0).toString());

            GradeModel gradeModelTmp = gd.getGradeModel(grade_id);
            if(null == gradeModelTmp){
                JOptionPane.showMessageDialog(null, "不存在该数据, 请刷新
面板! "+grade_id);
                return;
            }
        }
    }
}

```

```

        Object input_objObject =
JOptionPane.showInputDialog(null, "请输入名称: \n", "修改年级名称
", JOptionPane.PLAIN_MESSAGE, null, null, gradeModelTmp.getGrade_name()
);

        if(input_objObject != null){
            String grade_name = input_objObject.toString();
            if("").equals(grade_name)){
                return;
            }

if(grade_name.equals(gradeModelTmp.getGrade_name()))return;
            if(gd.modifyGrade(grade_id, grade_name)){
                JOptionPane.showMessageDialog(null, "修改成功!");
                initData();
                return;
            }
        }
        JOptionPane.showMessageDialog(null, "修改失败!");
        //pojo.removeRow(jTable.getSelectedRow());
        //数据量不是很多, 感觉重新刷新一次好点
        //counts = td.getLists(false, -1).size();
    } else {
        JOptionPane.showMessageDialog(null, "请选择要修改的行!");
    }

    //grade_lists = td.getLists(true, teach_id);
    flashData();
}
else if (e.getSource() == jButton_course_flash) {
    initData();
}
else if (e.getSource() == jButton_ok) {
    String grade_name = jTextField_teach_name.getText();
    if("").equals(grade_name){
        JOptionPane.showMessageDialog(null, "请输入名称!");
        return;
    }
}
//String teach_education =

```

```

teacher_lists.get(jComboBox_teach_education.getSelectedIndex()).get
Education();
        //String teach_education =
jComboBox_teach_education.getSelectedItem().toString();
        if(gd.addGrade(grade_name))
        {
            JOptionPane.showMessageDialog(null, "添加成功");
        }else{
            JOptionPane.showMessageDialog(null, "添加失败");
        }
        //grade_lists = td.getLists(false, -1);
        initData();
    }
    else if(e.getSource()==jButton_cancel){
        initData();
    }
}
}
}

```

4.5 界面设计

班级快速添加

班级名称: 所属年级:

班级最大人数:

年级快速添加

年级名称:

5 选课 · 课程管理模块

5.1 功能描述

用户可以选课 · 课程管理模块中对学生进行选课，打分，查看全体学生选课详情。

5.2 相关函数

```
private void initialize() {
    jLabel_teach_name = new JLabel();
    jLabel_teach_name.setText("授课老师: ");
    jLabel_teach_name.setBounds(new Rectangle(253, 18, 75, 29));
    jLabel_course_credit = new JLabel();
    jLabel_course_credit.setText("课程学分: ");
    jLabel_course_credit.setBounds(new Rectangle(23, 71, 88, 26));
    jLabel_course_name = new JLabel();
    jLabel_course_name.setText("课程名称: ");
    jLabel_course_name.setBounds(new Rectangle(23, 25, 88, 28));
    jLabel11 = new JLabel();
    jLabel11.setBounds(new Rectangle(725, 71, 96, 31));
    jLabel11.setText("课程介绍: ");
    jLabel = new JLabel();
    jLabel.setText("开课时间: ");
    jLabel.setBounds(new Rectangle(250, 63, 80, 30));
    this.setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    this.setSize(950, 500);
    this.setTitle("课程管理");
    //this.setModal(true);
    this.setLocationRelativeTo(null);

    jButton_course_add = new JButton();
    jButton_course_add.setText("添加");
    jButton_course_edit = new JButton();
    jButton_course_edit.setText("修改");
    jButton_course_query = new JButton();
    jButton_course_query.setText("查询");
    jButton_course_delete = new JButton();
```

```
jButton_course_delete.setText("删除");
jButton_course_flash = new JButton();
jButton_course_flash.setText("刷新");

jButton_ok = new JButton();
jButton_ok.setBounds(new Rectangle(496, 17, 114, 29));
jButton_ok.setText("确定");

jButton_cancel=new JButton();
jButton_cancel.setBounds(new Rectangle(496, 63, 115, 27));
jButton_cancel.setText("取消");

jLabel_course_counts = new JLabel();
jLabel_course_counts.setSize(20, 20);
jJToolBarBar = new JToolBar();
jJToolBarBar.setBounds(new Rectangle(5, 21, 714, 34));
jJToolBarBar.add(jButton_course_add);
jJToolBarBar.add(jButton_course_edit);
jJToolBarBar.add(jButton_course_query);
jJToolBarBar.add(jButton_course_delete);
jJToolBarBar.add(jButton_course_flash);
jJToolBarBar.add(jLabel_course_counts);

jTable = new JTable();
jScrollPane = new JScrollPane();
jScrollPane.setBounds(new Rectangle(20, 71, 700, 250));
jScrollPane.setViewportView(jTable);

jTextField_course_info = new JTextArea();
jTextField_course_info.setBounds(new Rectangle(725, 108, 167,
214));

jTextField_course_name = new JTextField();
```


3 课件结构设计

```
jTextField_course_name.setBounds(new Rectangle(116, 26, 122,
26));

jTextField_course_credit = new JTextField();
jTextField_course_credit.setBounds(new Rectangle(117, 71, 123,
26));

jComboBox_teach_name = new JComboBox();
jComboBox_teach_name.setBounds(new Rectangle(326, 18, 160, 30));

jTextField_course_begin_time = new JTextField();
jTextField_course_begin_time.setBounds(new Rectangle(330, 64,
160, 28));

jPanel = new JPanel();
jPanel.setLayout(null);
jPanel.setBounds(new Rectangle(22, 332, 700, 106));
jPanel.setBorder(BorderFactory.createTitledBorder("课程快
速开设"));

jPanel.add(jLabel_course_name, null);
jPanel.add(jLabel_course_credit, null);
jPanel.add(jTextField_course_name, null);
jPanel.add(jTextField_course_credit, null);
jPanel.add(jLabel_teach_name, null);
jPanel.add(jComboBox_teach_name, null);
jPanel.add(jLabel, null);
jPanel.add(jTextField_course_begin_time, null);
jPanel.add(jButton_ok, null);
jPanel.add(jButton_cancel, null);

jContentPane = new JPanel();
jContentPane.setLayout(null);
jContentPane.setBorder(BorderFactory.createTitledBorder("操
作一体化"));

jContentPane.add(jScrollPane, null);
jContentPane.add(jJToolBarBar, null);
```

```
jContentPane.add(jLabel1, null);
jContentPane.add(jTextField_course_info, null);
jContentPane.add(jPanel, null);
this.setContentPane(jContentPane);
jContentPane.add(jScrollPane, null);

this.setContentPane(jContentPane);

}
```

5.3 主函数入口

```
public CourseManage() {
    super();
    initialize();
    initData();
    btnListener btn = new btnListener();
    jButton_course_add.addActionListener(btn);
    jButton_course_edit.addActionListener(btn);
    jButton_course_query.addActionListener(btn);
    jButton_course_delete.addActionListener(btn);
    jButton_course_flash.addActionListener(btn);
    jButton_ok.addActionListener(btn);
    jButton_cancel.addActionListener(btn);

    jTable.getSelectionModel().addListSelectionListener(new
tableListener());
}
```

5.4 关键代码

```
public class btnListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        CourseDao cd = new CourseDao();

        if (e.getSource() == jButton_course_add) {
```

```
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
String course_begin_time = sdf.format(new Date());
jTextField_course_name.setEditable(true);
jTextField_course_begin_time.setEditable(true);
jTextField_course_begin_time.setText(course_begin_time);
jTextField_course_credit.setEditable(true);
jTextField_course_info.setEditable(true);
jButton_ok.setEnabled(true);
jButton_ok.setText("确定");
}else if(e.getSource() == jButton_course_edit){
    if (jTable.getSelectedRow() != -1) {
        String course_name =
jTable.getValueAt(jTable.getSelectedRow(), 1).toString();
        int course_credit =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
2).toString());
        String teach_name =
jTable.getValueAt(jTable.getSelectedRow(), 3).toString();
        String course_begin_time =
jTable.getValueAt(jTable.getSelectedRow(), 4).toString();
        jTextField_course_name.setText(course_name);
        jTextField_course_name.setEditable(true);
        jTextField_course_credit.setText(course_credit+"");
        jTextField_course_credit.setEditable(true);
        jTextField_course_begin_time.setText(course_begin_time);
        jTextField_course_begin_time.setEditable(true);
        jButton_ok.setText("确认修改");
        jButton_ok.setEnabled(true);
        jTextField_course_info.setEditable(true);
        for(int i = 0; i < jComboBox_teach_name.getItemCount(); i++){
            if(teach_name.equals(jComboBox_teach_name.getItemAt(i).toString()))
            {
                jComboBox_teach_name.setSelectedIndex(i);
                break;
            }
        }
    }
}
```

```

    }
    }else{
        JOptionPane.showMessageDialog(null, "请选择要修改的行!");
    }
}

else if (e.getSource() == jButton_course_delete) {
    if (jTable.getSelectedRow() != -1) {
        int course_id =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
0).toString());
        cd.deleteListByCourseId(course_id);
        JOptionPane.showMessageDialog(null, "删除成功!");
        model.removeRow(jTable.getSelectedRow());
    } else {
        JOptionPane.showMessageDialog(null, "请选择要删除的行!");
    }
}

else if (e.getSource() == jButton_course_query) {
    int course_id;
    try {
        course_id =
Integer.parseInt(JOptionPane.showInputDialog("按学号查询, 请输入要查询的
学号: "));
    } catch (Exception e2) {
        // TODO: handle exception
        return;
    }
    course_lists = cd.getLists(true, course_id);
    flashData();
}

else if (e.getSource() == jButton_course_flash) {
    initData();
}

else if (e.getSource() == jButton_ok) {
    if ("确认修改".equals(jButton_ok.getText())) {
        int course_id =
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(),
0).toString());

```

```

        int course_credit =
Integer.parseInt(jTextField_course_credit.getText().toString());
        int teach_id =
teacher_lists.get(jComboBox_teach_name.getSelectedIndex()).getTeach
_id();

        String course_name = jTextField_course_name.getText();
        String course_begin_time =
jTextField_course_begin_time.getText();
        String course_info = jTextField_course_info.getText();
        if(cd.modifyCourse(course_id, course_name, course_credit,
teach_id, course_info, course_begin_time)){
            JOptionPane.showMessageDialog(null, "修改成功!");
            initData();
        }else{
            JOptionPane.showMessageDialog(null, "修改失败!");
        }
        return;
    }

    String course_name = jTextField_course_name.getText();
    int course_credit =
Integer.parseInt(jTextField_course_credit.getText());
    String course_begin_time =
jTextField_course_begin_time.getText();
    String course_info = jTextField_course_info.getText();
    int teach_id =
teacher_lists.get(jComboBox_teach_name.getSelectedIndex()).getTeach
_id();

    if(cd.addCourse(course_name, course_credit, course_info,
course_begin_time, teach_id))
    {JOptionPane.showMessageDialog(null, "添加成功");
    }else
    {JOptionPane.showMessageDialog(null, "添加失败");

    course_lists = cd.getLists(false, -1);
    flashData();
    }
    else if(e.getSource()==jButton_cancel){

```

```
initData();  
  
}  
  
}  
}
```

5.5 界面设计

选课录入

×

年级选择：

一年级

班级选择：

核弹技术07-1

选课学生：

陈罗志

选修课程：

计算机技术

10004

号学生已选课程包括：

课程编号：1 课程名称：计算机技术
课程编号：2 课程名称：核弹技术

添加

取消

温馨提示：所选各门课程成绩互相独立。

批量排课

×

年级选择:

一年级

▼

班级选择:

▼

选修课程:

▼

确认添加选课

南昌航空大学课程设计（论文）

选课管理

×

选课信息一览表

按课程名称列举：

计算机技术

学生选课门数：

还没选课学生

记录数：4

查询方式：

按ID

☐

是否全表模糊

查询

操作面板

学生姓名：

所選课程：

计算机技术

打开选课管理面板

修改选课

删除选课

课程管理

×

操作一体化

添加

修改

查询

删除

刷新

记录数:9

课程ID	课程名称	课程学分	授课教师	开课时间
1	计算机技术	4	李老师	2011-05-28 12:38:58
2	核弹技术	3	李老师	2011-05-28 12:39:33
4	高等数学	8	吴老师	2011-05-28 12:40:30
8	高分子	5	李老师	2008-08-08 00:00:00
9	航空航天	6	吴老师	2009-09-06 00:00:00
11	软件工程导论	3	吴老师	2018-03-05 00:00:00
12	人工智能	4	张老师	2018-03-07 00:00:00
13	计算机网络	3	薛仁贵	2018-03-07 00:00:00
14	离散数学	4	吴老师	2018-03-07 00:00:00

课程介绍：

课程快速开设

课程名称：

授课老师：

李老师

确定

课程学分：

开课时间：

取消

6 用户登录注册模块

6.1 功能描述

用户登录模块负责验证用户身份，并允许注册用户访问系统的安全功能。通过验证用户提供的凭据，确保只有经过授权的用户可以使用系统。

6.2 相关函数

相关属性

```
19     private static final long serialVersionUID = 1L;
20     public static String storeUserName = null; // 登录用户名
21     public static String storeUserPassword = null; // 登录密码
22     static int storeUserId; // 登录用户名
23     static boolean RELOAD = true; // 重新登陆标记
24     static int login_user_type; // 0表示管理员, 1表示老师, 2表示学生
25     private JPanel jContentPane = null;
26     private JButton jButton21 = null;
27     private JButton jButton22 = null;
28     private JTextField jTextField = null;
29     private JPasswordField jPasswordField = null;
30     private JLabel jLabel_userName = null;
31     private JLabel jLabel_password = null;
32     private JLabel jLabel_privilege = null;
33     private JLabel jLabel_NCHU = null;
34     private BtnListener btl = null;
35     private JComboBox jComboBox = null;
36     private JLabel jLabel_tips = null;
37     private JLabel jLabel_title = null;
```

相

关

方

法

```

38
39 > public Login() {...}
43
44 > public Login(boolean reload) {...}
48
    形参: args
    主函数
53 > public static void main(String[] args) {...}
57
58 //载入图片
59 @> private static void loadImageAndDisplay(JLabel label, String imagePath) {...}
73
74 > private void initialize() {...}
181
    监听类
185 > public class BtnListener implements ActionListener {...}

```

6.3 主函数入口

```

53 > public static void main(String[] args) {
54     Login login = new Login(RELOAD);
55     login.setVisible(true);
56 }
57

```

6.4 关键代码

A. 登录界面

1. 实现多个组件加间距居中对齐

```

double widthUsernameText =
jLabel_userName.getPreferredSize().getWidth();
double widthUserinput =
jTextField.getPreferredSize().getWidth();
double gap1 = 50;
double x1 = (windowWidth - (widthUsernameText +
widthUserinput + gap1)) / 2.0;
double x2 = x1 + gap1 + widthUsernameText;
jLabel_userName.setBounds(new Rectangle((int) x1, 150,
150, 50));
jTextField.setBounds(new Rectangle((int) x2, 150, 250,
40));

```

2. 点击登录后，后端与数据库进行交互

```

UserDao ud = new UserDao();
String user = jTextField.getText().trim();
String password = new

```

```
String(jPasswordField.getPassword()).trim();//char
to String
```

```
storeUserName = user;
storeUserPassword = password;
login_user_type = jComboBox.getSelectedIndex();
```

3. 输入的账户名和密码为空的判断

```
if ("".equals(user)) {
    JOptionPane.showMessageDialog(null, "用户名不能为空");
    return;
}
if ("".equals(password)) {
    JOptionPane.showMessageDialog(null, "密码不能为空");
    return;
}
```

4. 根据输入的账号类型，连接对应的数据库进行账户名密码的判断。
仅展示管理员登录代码。

```
if (login_user_type == 0) {
    if (ud.userLogin(login_user_type, storeUserName,
storeUserPassword)) {
.....
}
```

5. 登录成功后，唤起主框架界面，并且进行操作日记录入。

```
dispose();
MainFrame mf = new MainFrame();
mf.setTypeOfUser(0);
mf.setUserName(storeUserName);
mf.setVisible(true);
JDBCUtil.update("insert into
c_log(login_user,log_operate,log_time) values('" +
storeUserName + "','"+ log_operate + "','"+ dt + "')");
```

B. 用户管理

1. 初始化表格的表头

```
String heads[] = { "ID", "姓名", "登录时间", "登录 ip", "用户权限" };
```

```
model = new DefaultTableModel(null, heads);  
UserDao ud= new UserDao();  
user_lists = ud.getLists(false, -1);  
flashData();
```

2. 进行数据更改后，刷新表格数据

```
counts = user_lists.size();  
  
model.setRowCount(user_lists.size()); // 设置行数  
for(int i=0;i<counts;i++){  
  
model.setValueAt(user_lists.get(i).getUser_id(), i,  
0);  
  
model.setValueAt(user_lists.get(i).getUser_name(), i,  
1);  
  
model.setValueAt(user_lists.get(i).getUser_login_time(), i, 2);  
  
model.setValueAt(user_lists.get(i).getUser_login_ip(), i, 3);  
  
model.setValueAt(user_lists.get(i).getUser_privilege(), i, 4);  
}  
jTable.setModel(model);  
jTable.getColumnModel().getColumn(0).setPreferredWidth(30);  
jTable.getColumnModel().getColumn(1).setPreferredWidth(35);  
jTable.getColumnModel().getColumn(2).setPreferredWidth(60);  
jTable.getColumnModel().getColumn(3).setPreferredWidth
```

```
dth(60);  
jTable.getColumnModel().getColumn(4).setPreferredWidth(40);
```

3. 自己不能删除自己账号的验证

```
int id =  
Integer.parseInt(jTable.getValueAt(jTable.getSelectedRow(), 0).toString());  
if (Login.storeUserId == id) {  
    JOptionPane.showMessageDialog(null, "自己不能删除自己!");  
}
```

C. 修改密码界面

1. 修改完密码后，储存在类的属性中，并且更改数据库的数据，关闭主界面窗口，重新登录。

```
Login.storeUserPassword = new1; //把改好的密码写回去。
```

```
dispose(); //关闭修改密码窗口
```

```
mainFrame.dispose(); //关闭主界面窗口
```

```
Login loginDialog = new Login();
```

```
loginDialog.setVisible(true); //改好了重新登录
```

6.5 界面设计

6.5.1 登录首页



6.5.2 主框架界面

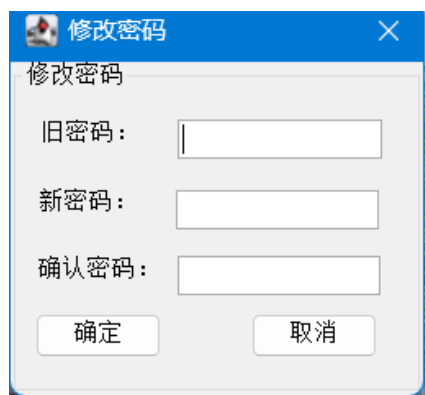
3 课件结构设计



6.5.3 用户管理



6.5.4 修改密码界面



修改密码

修改密码

旧密码:

新密码:

确认密码:

确定 取消

7 成绩管理模块

（完成人：23201837-李茂田）

7.1 功能描述

学生成绩录入:管理员或教师能够录入学生的各科目的成绩数据。每条成绩记录通常包括学生 ID、科目、成绩、考试日期等信息。

成绩查询与浏览:用户（教师、或管理员）能够根据学生 ID、课程、时间段等条件查询和浏览学生的成绩信息。查询结果应支持排序和筛选功能。

成绩统计与分析:系统能够根据不同的维度（如学生、科目、班级等）对成绩数据进行统计和分析，包括平均成绩、最高分、最低分、成绩分布等统计指标。

数据安全与保护:确保数据的机密性和完整性，防止未经授权的访问、篡改或泄露。

7.2 相关函数

```

46 > public GradeManage() {...}
60
61 > private void initialize() {...}
173
174 > private void gettsButtonGroup() {...}
181

    初始化界面
185 > public void initData() {...}
200 > public void flashData(){...}

    内部类监听器模块
214 > public class btnListener implements ActionListener {...}
298
299 > public class tableListener implements ListSelectionListener {...}
307 > public static void main(String args[]){...}
311 }
312

```

7.3 主函数入口

```

46     public GradeManage() {
47         super();
48         initialize();
49         initData();
50         btnListener btn = new btnListener();
51         jButton_course_add.addActionListener(btn);
52         jButton_course_query.addActionListener(btn);
53         jButton_course_delete.addActionListener(btn);
54         jButton_course_flash.addActionListener(btn);
55         jButton_ok.addActionListener(btn);
56         jButton_cancel.addActionListener(btn);
57         jTable.getSelectionModel().addListSelectionListener(new tableListener());
58     }

```

7.4 关键代码

A. 已选课程成绩录入功能

1. 下拉框数据更改时，刷新下一个下拉框数据

```

jComboBox_course.removeAllItems();
CCourseDao cd1 = new CCourseDao();
int stu_id =
student_lists.get(jComboBox_stu.getSelectedIndex())
.getStu_id();
ccourse_lists = cd1.getListsByStuId(stu_id, false);
for (int i = 0; i < ccourse_lists.size(); i++) {

jComboBox_course.addItem(ccourse_lists.get(i).getCo
urse_name());
}
int course_index =
jComboBox_course.getSelectedIndex();
//int course_id =
ccourse_lists.get(course_index).getCourse_id();
if(course_index != -1)
jTextField_course_mark.setText(Integer.toString(cco
urse_lists.get(course_index).getCcourse_mark()));

```

2. 点击开始过滤单选框，课程选择下拉框刷新，只会出现未打分的课程

```

if(jComboBox_stu.getSelectedIndex() != -1 && jComboBox_
course.getSelectedIndex() != -1) {
    int stu_id =
student_lists.get(jComboBox_stu.getSelectedIndex())

```

```

.getStu_id();

CCourseDao ccd = new CCourseDao();
ccourse_lists = ccd.getListsByStuId(stu_id, true);
jComboBox_course.removeAllItems();

for(int i =0;i<ccourse_lists.size();i++){

jComboBox_course.addItem(ccourse_lists.get(i).getCourse_name());
}
int course_index =
jComboBox_course.getSelectedIndex();
if(course_index != -1)

jTextField_course_mark.setText(Integer.toString(ccourse_lists.get(course_index).getCcourse_mark()));
}

```

B.成绩区间分布图功能

1.找到数据中的最大值，缩放柱状图，绘制柱子

```

int x = i * barWidth;
int barHeight = (int) ((double) data[i] / maxValue * height); // 缩放数据以适应面板高度

int y = height - barHeight; // 计算柱子顶部位置

// 绘制柱子

g.fillRect(x, y, barWidth - 1, barHeight);

```

2.绘制数据，动态调整，防止超出面板边界而无法展示

```

y+=fm.stringWidth(pencentage);
if (y-50<=0){
}else {
    y-=50;
}
g.setColor(new Color(241,115,172));

```

```
g.drawString(pencentage, x + (barWidth - pWidth) / 2, y);
```

C.快捷查询成绩

按格式显示数据

```
jTextArea_ccourse = new JTextArea();
CCourseDao cCourseDao = new CCourseDao();
List<StuCourseMarkModel> lists =
cCourseDao.getCCourseMarkCourseNameByStuId(studentID);
for(int i=0;i<lists.size();i++){
    String courseName = lists.get(i).getCourseName();
    int mark=lists.get(i).getCourseMark();

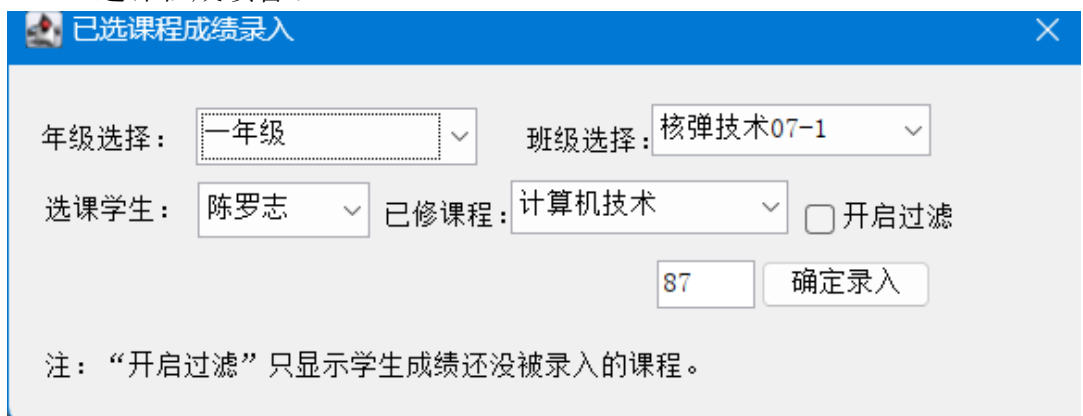
    jTextArea_ccourse.append("课程名: " + courseName +
    "+"课程分数: "+mark);

    jTextArea_ccourse.append("\n");
}

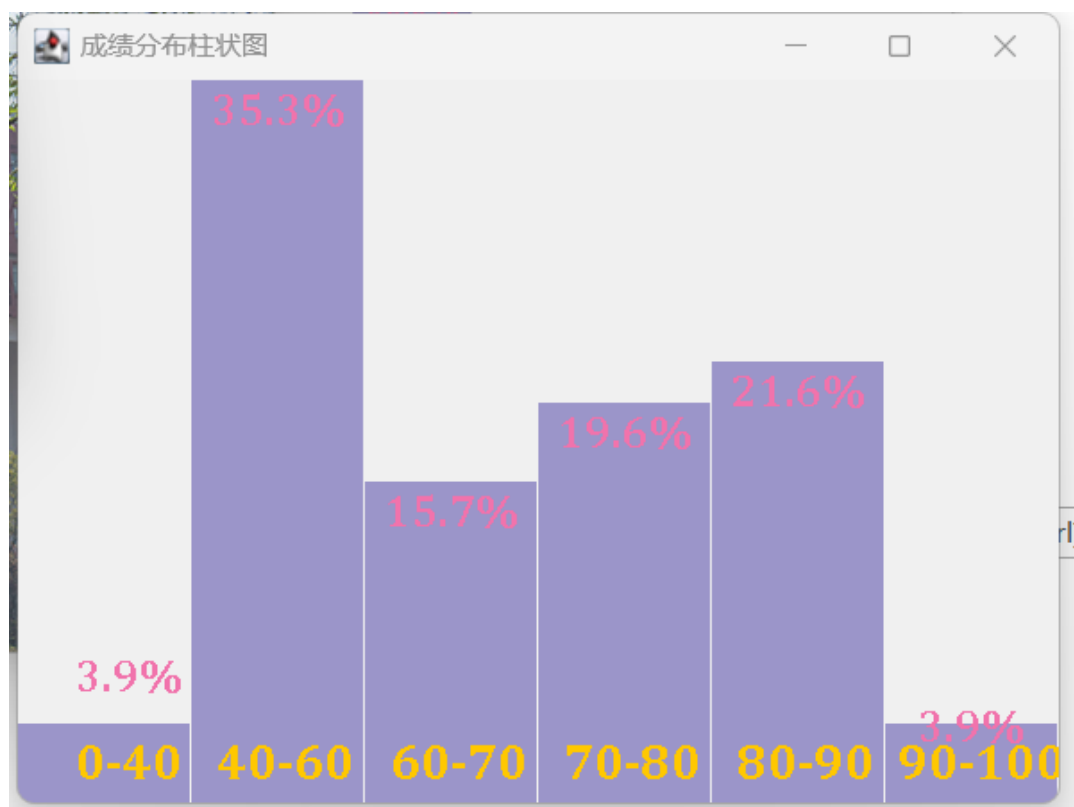
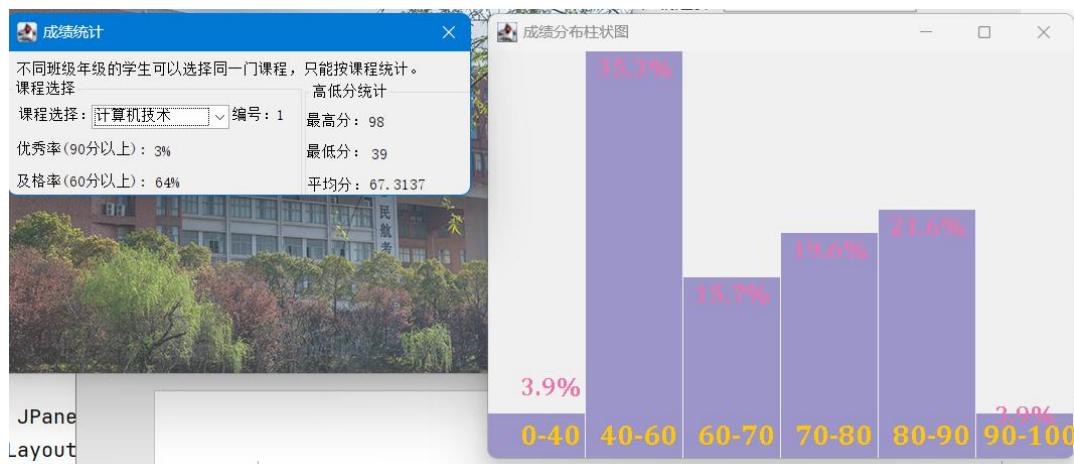
jTextArea_ccourse.setFont(new Font("楷体", Font.PLAIN,
30));
jTextArea_ccourse.setEditable(false);
```

7.5 界面设计

1. 已选课程成绩管理



2. 成绩统计与区间分布图



3. 快捷查询我的成绩

查看我的成绩



我的成绩

课程名：计算机技术	课程分数：87
课程名：核弹技术	课程分数：65
课程名：高分子	课程分数：57
课程名：航空航天	课程分数：76
课程名：潜艇技术	课程分数：68
课程名：人工智能	课程分数：86
课程名：离散数学	课程分数：75

8 测试

8.1 集成测试

学生管理系统的测试目的主要包括以下几个方面：

1. 验证功能的完整性和正确性：确保学生管理系统能够按照设计要求正常运行，包括学生信息管理、课程安排、成绩录入与查询等核心功能的正确性和完整性。
2. 评估系统的性能和稳定性：通过性能测试，评估系统在不同负载和并发访问条件下的响应速度、资源消耗情况，以及系统在高压力期间的表现，确保系统能够保持稳定的性能。
3. 保障系统的安全性和数据保护：进行安全性测试，验证系统在面对潜在的网络攻击和数据泄露风险时，能够有效保护用户的个人信息和敏感数据，包括访问控制、数据加密、安全认证等方面的测试和优化。
4. 确保系统的兼容性和可用性：通过兼容性测试，确保学生管理系统能够在不同的操作系统、设备和浏览器上正常运行，提升用户的使用体验和操作效率。
5. 验证系统的错误处理能力：错误处理测试确保系统在面对用户输入错误、异常操作或系统故障时能够正确识别、报告和处理，确保系统的稳定性和可靠性。
6. 确认系统修改和更新的影响：回归测试确保系统在经过修改或更新后，新功能的添加或现有功能的改动不会破坏系统的其他部分，保持系统的整体一致性和稳定性。

总 结

在这个项目中，我们组由三人组成，每个人分工合作，尽管最初选择的架构设计不同导致了我们要实现的效果有些困难。在这过程中，我们的成员曾经争论过一些问题，但是我们全力寻找可能的解决方案和函数方法，并与同学进行了深入讨论，收获颇丰。

这次暑期实训让我们深入了解了项目的构建、实施和完成的整个过程，也让我们接触到了许多之前未曾涉及的算法、语法知识以及环境配置。从开始的迷茫到测试的成功实现，都令我们收获颇丰。一开始压力让我们感到挑战但我们从中汲取了力量，启动了项目的构思并最终实现了我们的想法。

我清楚记得那些日日夜夜，我们埋头查阅相关资料，不断完善我们的项目。这次实训让我深刻认识到了团队合作的重要性和默契配合的价值。在这段实践中，我收获颇丰，也深信如果有下一次机会，我们能够做得更出色。

参考文献资料

- [1]田宇 基于 WebGIS 的三维地质灾害信息管理系统设计与实现
- [2]吴泷彬 谢宇飞 刘森 李彪 段晓辉 以学生为中心的弹性开放实验室管理系统
- [3]张彬 基于 Java 的员工信息数据库管理系统的开发与设计