



# Java编程技术实验报告

**题目：学生信息管理系统**

学 院 计算机学院

专 业 计算机科学与技术

年级班别 2015级3班

学 号 3115004974

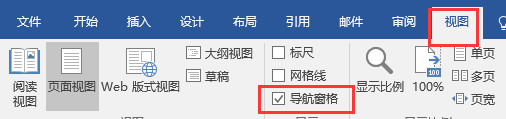
学生姓名 刘国斌

指导教师 李藜

2016年12月

|  |  |
| --- | --- |
| 难度系数 |  |
| 独立完成工作量占总工作量的比例 |  |
| 程序功能完成情况 |  |
| 报告内容与格式 |  |
| 总评成绩 |  |

概述

建议开启导航视图以获得更好的阅读体验！

学生成绩管理系统是广泛使用的一种信息管理系统，在学生成绩管理系统中，需要能处理各班信息、各班下属学生的个人信息及每个学生学习成绩信息。班级信息包括班级编号、班级名称，学生信息包括学号、姓名、性别、所属班级、联系方式等，及各门课程的成绩。与班级相关的操作是查看已有班级、新增班级、修改班级名称、删除班级、录入新生信息等；与学生相关的操作是查看已有学生、修改学生基本信息、删除学生、录入成绩、修改成绩、根据姓名、学号、班级、课程等多条件查询学生成绩，并可以依据多种条件进行排序。

用到的技术

### 平台、语言和工具

* 程序编写语言：Java、SQL
* 开发平台：Eclipse Mars.2、Navicat 11.0.17
* 使用工具：C3P0、Log4j
* 运行环境：jre1.7\_80、MySql 5.7

### 技术细节

**图形**：

1. 本程序的UI使用Java原生的GUI界面，按钮在下排
2. 统一使用JOptionPane类的信息提示窗口。
3. 子窗口弹出时父窗口无法点击。
4. 所有的窗口在关闭时使用dispose();而不是setvisible，保证在窗口关闭时释放该窗口的资源。
5. 去除窗口对象实例化时的引用，利用JVM的垃圾销毁机制节省内存。

**Java编程**：

1. **内部类**：所有的监听器类都用内部类实现，可以使外部无法访问这个内部类，也方便了对其外部类的监听。
2. **后台架构：**程序使用仿MVC架构实现，为view、model（po）、service、util、dao五层。使得变成风格更加清晰、模块化，便于抽离和拓展。
3. **模板方法模式**：所有的dao类继承一个IOToSQL类作为baseDao，baseDao是一个抽象类，里面有增删查改四个基本的未实现方法和一些封装的实现方法，具体类dao继承并实现具体的抽象方法。
4. **反射**：模板方法遍历数据库使用反射原理，传入resaultSet，用newinstance()获取某个po相应的对象，再获取其get、set方法，最后返回需要的po类对象的Map集，再转换成List；
5. **泛型编程**：dao抽象类使用了Java泛型，提高父类整体的抽象程度。
6. **JDBC连接池**：池化数据库连接，提高连接的利用率和程序运行的效率。

程序结构

### 后台架构：

程序使用仿MVC架构实现，整个程序分为view、model（po）、service、util、dao五层

* Dao层：和数据库交互的层次，由于使用未封装的JDBC，所以本程序的dao层也是持久层。
* Service：业务逻辑处理层，负责对dao层异常处理、view层数据校验、业务流程控制、业务逻辑处理。
* Model：实体类层，所有的业务实体类都会存放在这里。比如老师、管理员。
* View：显示层，所有有关GUI的类，以及窗口跳转处理都在这里。
* Util：工具类层，所有使用到的一些工具方法，例如数据库连接的获取、关闭，字符串的正则判断、长度判断、空判断、字符拼接，view层表格处理等都在此封装。由于是工具方法，所以这里的方法都用static字段修饰。

其中，dao为数据层，util/po/service为业务逻辑层，view为显示层。

图形层通过业务逻辑层获取数据层的数据，降低了耦合，实现模块化编程

整体架构如下图所示：

业 务 层

Service/po/util

数 据 层

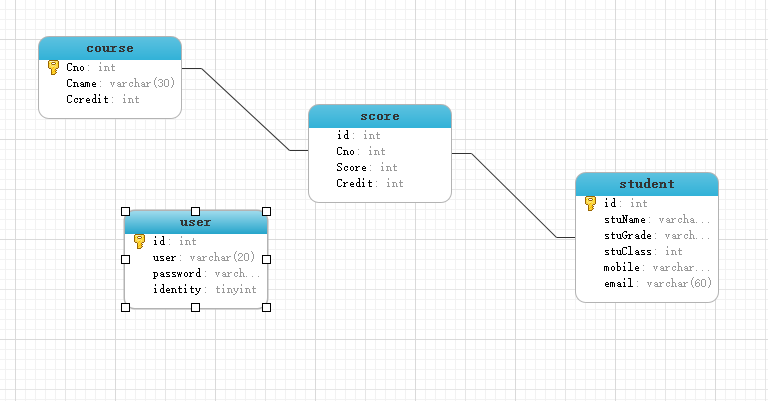
Dao

显 示 层

View

### 数据库表结构

**数据库ER图表**



* Course：课程表，有课程id(主键)、课程名、学分三个字段
* Score：分数表，有分数id（student表外键）、课程表主键id的关联外键、分数、学分。
* Student：学生表，学生学号id（主键）、学生名、学生年级、学生班级、手机号、邮箱。
* user：用户id（主键）、用户名、密码、身份（管理员还是老师）

**数据库表的外键**

**多对多**：Student和course有多对多的关系，使用中间表score作为关联实现这种多对多的关系。

**一对多**：student和score、course和score都是一对多的关系，由于这个程序的特殊性，可以将多对多的映射表作为score表实现。

**所有的外键都是级联删除属性**，删除一条主数据时，有关联外键的数据也会一同删除。比如删除一个学生时，会将其所有的分数信息删除。

程序功能展示

### 整体结构

按姓名查找

按学号查找

按班级查找

按课程查找

权限升级

删除用户

更改权限

录入新生

删除学生

学生成绩管理系统

权限管理

学生管理

成绩管理

修改学生

查找学生

录入成绩

修改成绩

查找成绩

用户登录

学分规则

学分管理

修改学分

课程管理

添加课程

删除课程

查找课程

修改课程

按课程查找

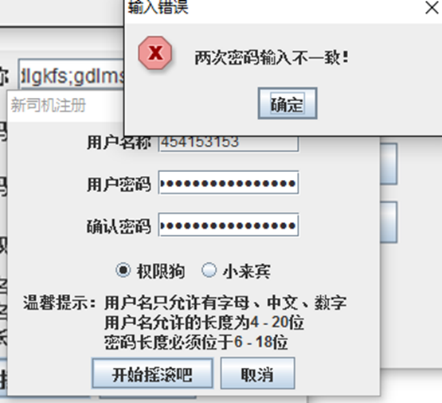
按课程号查找

按学分查找

### 支持两种身份的用户注册、登录

登录界面如图（图片已经更换）

对所有的输入都有进行正则校验，用户名有2-6位中文和6-18位英文或数字限制，且不能输入特殊符号；密码有6-18位英文或数字限制。如下图

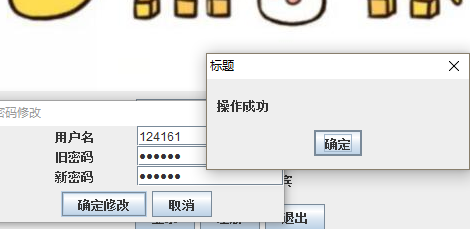
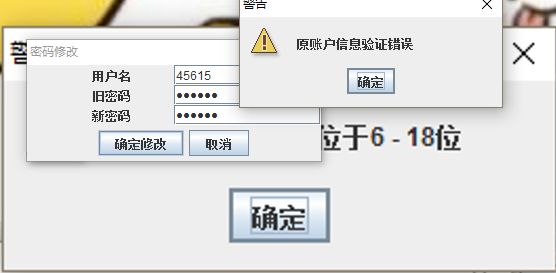




### 修改密码

对原有密码进行更改，对密码同样有正则判断。



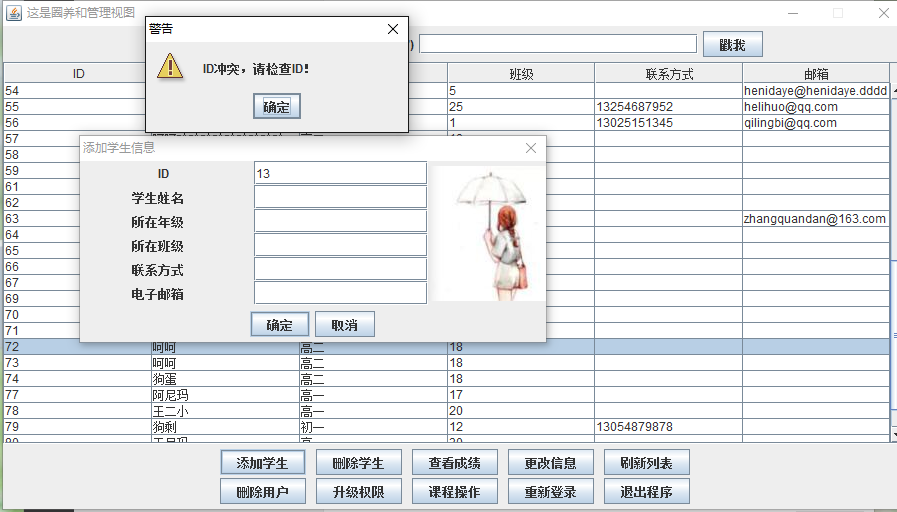


### 对学生的增删查改

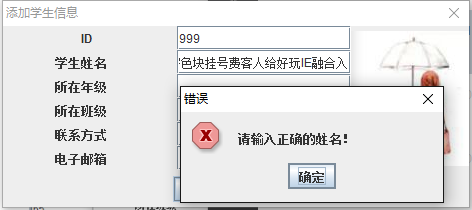
可以对学生的信息进行修改、学生查询、增加删除。所有输入操作均有防暴力输入和正则判断。

* **增加学生**

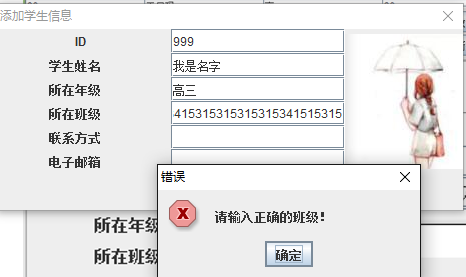
**已存在的学号不能重复添加**



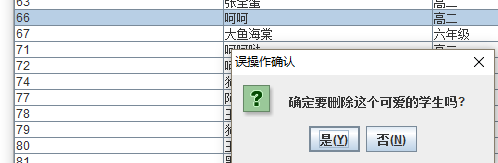
**姓名输入过长、有特殊符号、非纯中英文、含有数字**



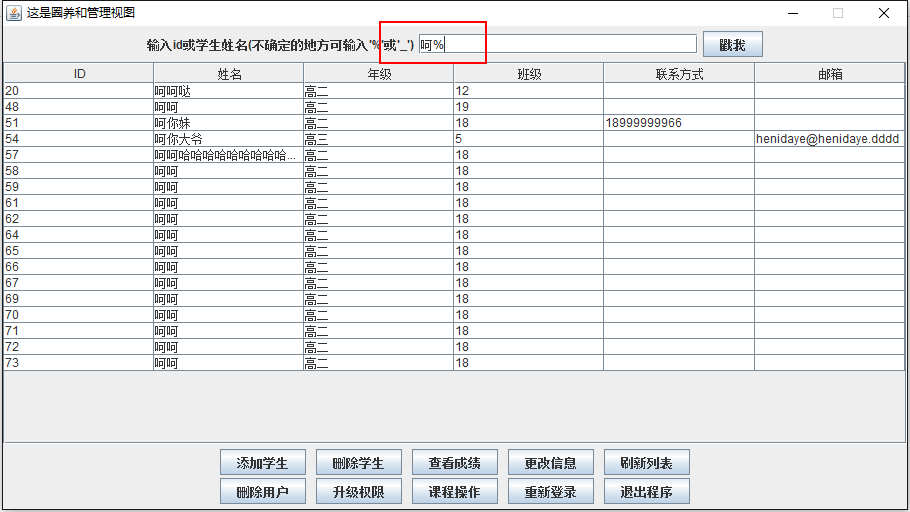
**班级数字过大、非数字**



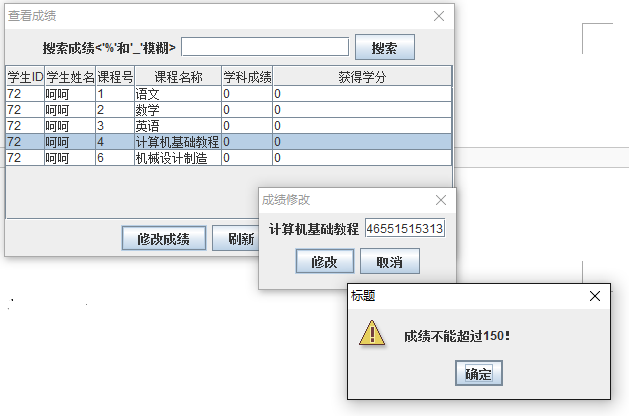
* + **可以对学生删除操作**

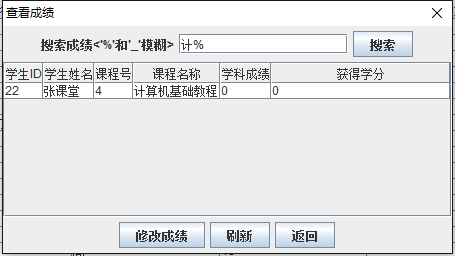


* + **模糊查询学生信息**



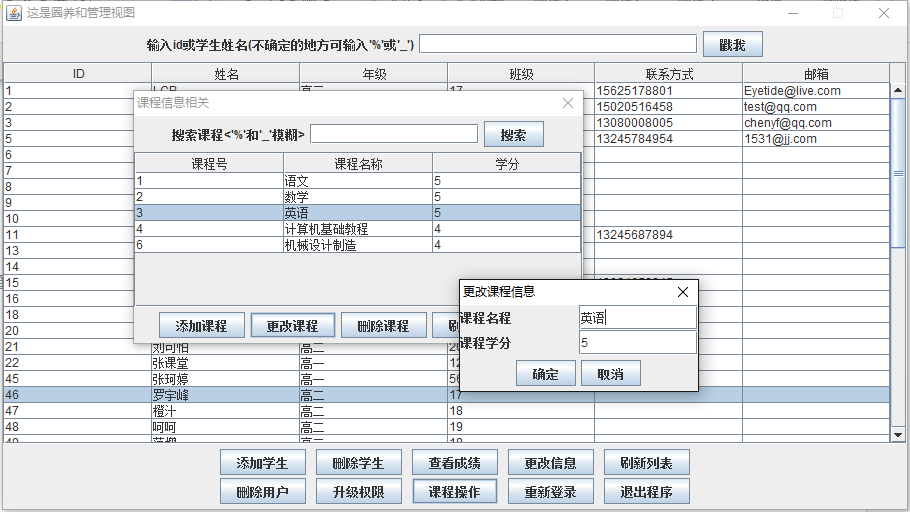
* **成绩管理和查询**



**查询成绩**



课程更改**（可以更改学分，但获得该课程学分的学生学分也会随之更改）**



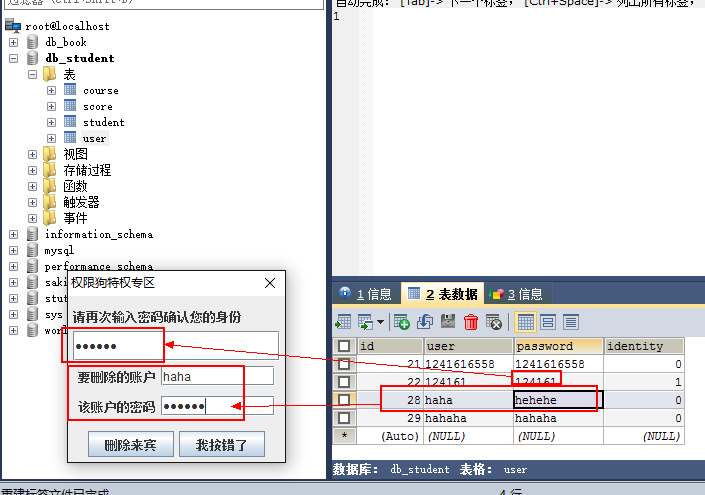
### 权限管理

老师可以增删学生、查看修改成绩、升级成管理员账户；管理员可以修改学生课程、学分和老师账户的管理.

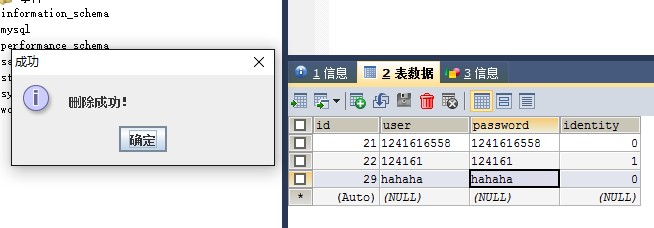


### 删除普通老师用户

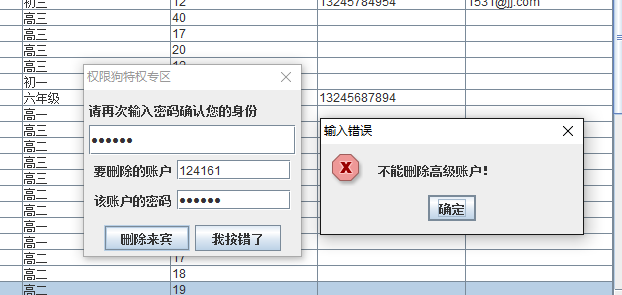
**高级用户模式**



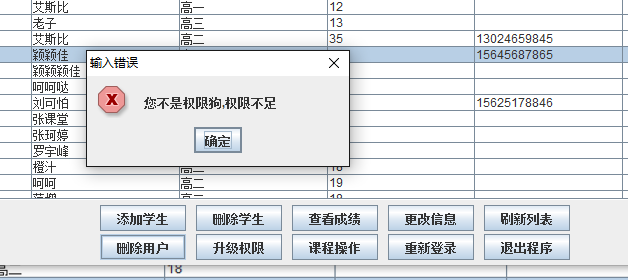
**删除确认，此时数据库还有这个用户**

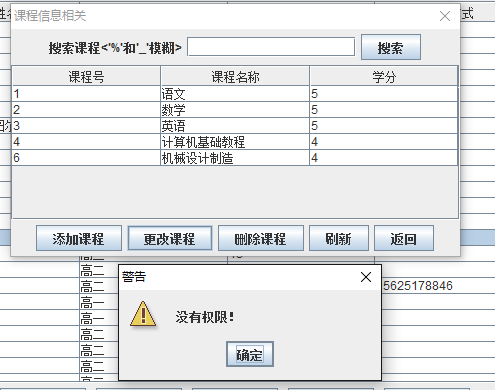


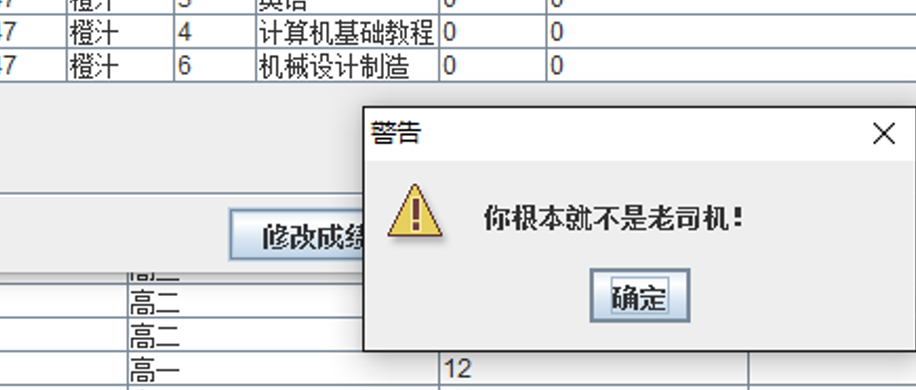
**删除成功后，用户消失**



**老师账户登录，没有高级权限**

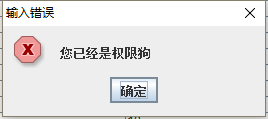




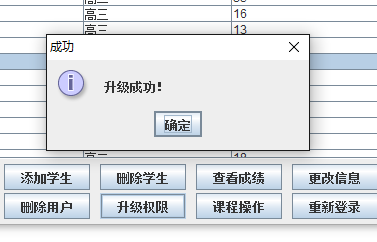


### 权限升级

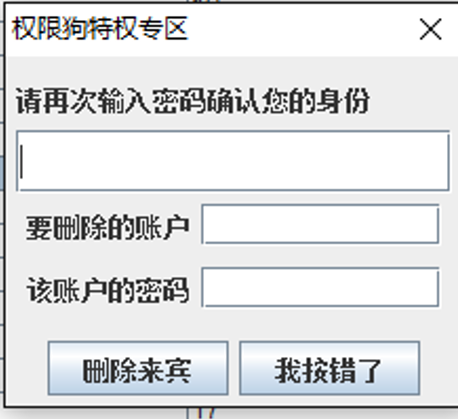
**对于高级用户**



**对于低级用户**



**升级权限后可以正常使用高级功能**



业务流程

退出系统

录入

修改

删除

进入系统

数据操作

成绩查询

查询

学分

查询

课程

按姓名

、

学号

查询

按

班级

、

课程查询

用户登录

权限管理

升级权限

删除用户

文件结构

