

**Java面向对象课程设计结题报告**

|  |  |
| --- | --- |
| **报告题目:** | **教务管理系统** |

|  |  |
| --- | --- |
| **学 院 :** | **信息与电子工程学院** |
| **专 业 :** | **计算机科学与技术** |
| **班 级 :** | **计算机141班、142班、142班** |
| **学 号 :** | **5140399272**  **1140299259、1140299232** |
| **学生姓名 :** | **陈彦琦（组长）、**  **钟壹、王文杰** |
| **指导教师 :** | **许加兵** |

**二○一六年六月**

# 课程设计任务书

## 一、任务目标

1.加深对JAVA语言的理解和掌握，增强对面向对象编程思想的实际运用能力，利用MyEclipse和MySQL制作一个教务管理系统。

2.培养团队协作能力，充分发挥每一个团队成员的优点，锻炼独立思考能力，通过查找文献解决问题，提高沟通表达能力，共同解决一己之力不能解决的疑题难题。

## 二、需求分析

（1）教师客户端功能

* 有密码，可以更改密码
* 可以添加学生和学生的基本信息
* 可以根据学号查询学生基本信息及成绩
* 可以添加新课程、新班级
* 可以控制选课时间
* 可以录入更改公布成绩
* 每个管理员只能管理自己所在学院的信息

（2）学生客户端功能

* 可以查看自己的基本信息
* 可以查看自己的成绩
* 可以选课并查看课表

## 三、系统结构图

教务管理系统

教师客户端

学生客户端

修改密码

添加学生

学生信息

开课设置

课程成绩

增加班级

增加课程

修改密码

基本信息

选择课程

查看课表

成绩查询

## 四、任务分配

* 陈彦琦（组长）：规划跟进总体任务进度，保证产品完成时间和产品质量，需求分析处理，代码结构设计，提供数据库技术支持，编写课程设计报告。负责代码模块为主页面设计，教师客户端的增加班级，增加课程，学生客户端的查看课表，成绩查询。
* 钟壹：查找资料，需求分析，程序测试，部分代码编写注释，添加部分功能，负责代码模块为教师客户端的开课设置，课程成绩，学生客户端的选择课程
* 王文杰：查找资料，需求分析，程序测试，部分代码编写注释，添加部分功能。负责代码模块为教师客户端的修改密码，添加学生，学生信息，学生客户端的修改密码，基本信息

## 五、计划进度

1．5月17日--5月19日：确定项目组成员，搜索资料，查询文献，确定选题。

2．5月20日-- 5月26日：进行需求分析，构建数据库设计图，明确功能模块。

3. 5月日27--5月30日：细分任务，着手界面开发。

4．5月31日-- 6月6日：着手业务逻辑开发。

5．6月7日—6月13日：尽兴教务管理系统的平台调试、测试、修改与完善  
6．6月14日—6月19日：撰写课程设计报告

7．6月20日：课程设计报告答辩

## 六、主要参考文献

[01] 李刚. 疯狂JAVA讲义[M].电子工业出版社.2008

[02] 高宏静. Java从入门到精通[M].化学工业出版社 2009

[03] 朱福喜,傅建明,唐晓军. JAVA项目设计与开发范例 [M]. 电子工业出版社.2005

[04] 张靓,顾慧敏. JAVA案例开发[M]，中国水利水电出版社，2005

[05] 陆正武,张志立. JAVA项目开发实践[M]，中国铁道出版社，2005

[06] 王寅乐. Java项目开发案例导航[M].电子工业出版社.2012年3月

学生（签字）陈彦琦、钟壹、王文杰 2016年5月17日

指导教师 许加兵 2016年5月17日

# 摘 要

随着我国的高等教育事业的蓬勃发展，高校的教务改革也在全面的推行，学分制在各校全面启动，高校信息的自动处理以及网络式的信息交互方式越来越被认可也应用，让计算机来管理学生的信息是现在各个高校够在积极进行的的工作之一，也是高校教务管理工作的重要美容之一。本系统以贵阳学院物电系的教务管理工作的工作内容为实际背景，主要实现用户信息管理、信息的发布和学生成绩的管理等功能。

本文首先介绍了高校教务管理系统的研究背景，简单的介绍常用的数据库技术，主要是SQL Server数据库技术；具体分析了高校教务管理系统的功能需求、角色分析、业务流程和系统目标；本系统分析了高校教务管理信息系统的相关实体，主要包括学生、教师、信息和课程等。

# Abstract

Along with our country's higher education career of booming development, college educational reform in comprehensive, the implementation of credit system in the schools in full swing, and university of automatic processing and information of the information network interactive way more and more recognized also application, let the computer to manage students' information is now various colleges and universities in the positive enough work, and also one of the important work of college educational administration management one of beauty. This system to guiyang college educational administration of the department of electrical things work content for the actual background, mainly realizes user information management, information release and student performance management etc. Function.

This paper first introduced the college educational administration management system the research background, the introduction of the commonly used database technology, mainly is the SQL Server database technology; Detailed analysis of the function of the college educational administration management system requirements, the role of analysis, the business process and system target; This system were analyzed in the teaching management information system related entities, mainly including students, teachers, and information and course, etc.

**目录**

[课程设计任务书 2](#_Toc454365208)

[一、任务目标 2](#_Toc454365209)

[二、需求分析 2](#_Toc454365210)

[三、系统结构图 3](#_Toc454365211)

[四、任务分配 3](#_Toc454365212)

[五、计划进度 3](#_Toc454365213)

[六、主要参考文献 4](#_Toc454365214)

[摘 要 5](#_Toc454365215)

[Abstract 6](#_Toc454365216)

[1. 系统分析 1](#_Toc454365217)

[1.1 问题描述 1](#_Toc454365218)

[1.2 需求分析 1](#_Toc454365219)

[1.3 技术分析 2](#_Toc454365220)

[2. 系统设计 2](#_Toc454365221)

[2.1 系统总体结构设计 2](#_Toc454365222)

[2.2 系统各个功能模块设计 3](#_Toc454365223)

[3． 教务系统设计 5](#_Toc454365224)

[3.1系统开发及运行环境 5](#_Toc454365225)

[3.2 系统主要功能要求 5](#_Toc454365226)

[3.3 教务管理系统总体结构 6](#_Toc454365227)

[4. 数据库设计 6](#_Toc454365228)

[4.1安全性设计 6](#_Toc454365229)

[4.2设计过程 7](#_Toc454365230)

[4.3 E-R图与表 7](#_Toc454365231)

[4.4数据库部分源代码 9](#_Toc454365232)

[5. 系统实现 11](#_Toc454365233)

[5.1.3教师客户端 15](#_Toc454365234)

[5.2 UML图 18](#_Toc454365235)

[功能模型—用例图 18](#_Toc454365236)

[对象模型—类图 19](#_Toc454365237)

[5.3程序设计及调试运行 19](#_Toc454365238)

[5.3.1 连接池 20](#_Toc454365239)

[5.3.2 JFreeChart 23](#_Toc454365240)

[5.2 程序源代码摘要 23](#_Toc454365241)

[公共类 登陆 23](#_Toc454365242)

[学生类 客户端界面 28](#_Toc454365243)

[教师类 管理成绩 33](#_Toc454365244)

[7. 结题感悟 39](#_Toc454365245)

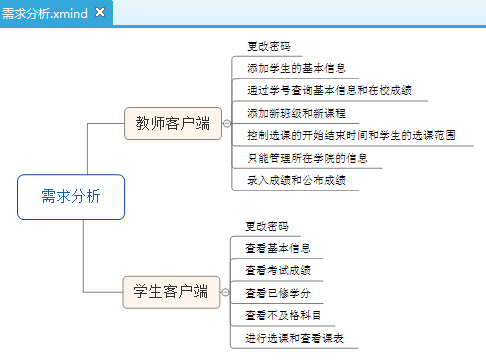
# 1. 系统分析

## 1.1 问题描述

传统的教务管理采用人工记录或者简单的数据管理软件，本身就存在着诸如存储查询修改维护信息不方便等问题，而随着学校规模的扩大和招生人数的增加，解决这些问题刻不容缓，为此，我们选择开发这套教务管理系统软件，在解决传统的教务管理方式存在的问题的同时，更加注重教务管理人员和学生的用户体验。

## 1.2 需求分析

教学管理系统的总目标是：在计算机网络，数据库和先进的开发平台上，利用现 有的软件，配置一定的硬件，开发一个具有开放体系结构的、易扩充的、易维护的、 具有良好人机交互界面的教学管理系统，实现教学管理的自动化的计算机系统，为教 师进行课件、学生作业管理和学生提交作业、下载课件、疑难解答提供方便。 根据可行性研究的结果和客户的要求，分析现有情况及问题，采用 Client/Server 结构，主要有学生、教师、管理员三个处理子系统。



**图1 需求分析图**

## 1.3 技术分析

教务管理系统可用JAVA相关知识完成，理论上没有太大难度，如遇困难可通过网络书籍老师等途径解决，团队成员有三个人，陈彦琦，钟壹，王文杰，其中，陈彦琦在实验室做过类似的项目，钟壹实验课听的十分认真，王文杰在开放性实验课也做过类似的项目，三个人分工合作，尽职尽责，有难同当，共同探讨，一定可以在规定时间高质量的完成所有任务。

# 2. 系统设计

## 2.1 系统总体结构设计

教师客户端

录入成绩和公布成绩

控制宣科的开始结束时间

添加新班级和新课程

通过学号查询学生的基本信息

添加学生的基本信息

更改密码

学生客户端

更改密码

察看基本信息

察看考试成绩

察看已修学分

察看不及格科目

进行选课和察看不及格科目

## 2.2 系统各个功能模块设计

**（1）**

数据库核对旧密码后确认新密码

教师输入旧密码和新密码

教师客户端

更改密码

**（2）**

数据库在学生信息表中添加学生的基本信息

学生将自身基本信息告诉老师，老师将其输入教务管理系统

教师客户端

添加学生的基本信息

**（3）**

数据库通过学号连接不同的表格，在查询页面中生成学生的基本信息和在校成绩

教师在查询页面输入学生学号

教师客户端

通过学号查询基本信息和在校成绩

**（4）**

教师在添加页面上输入新班级信息或者新课程信息

数据库添加新班级或者新课程

教师客户端

添加新班级和新课程

**（5）**

数据库添加可选课程信息并将允许学生选课置数为1

教师输入可选课程信息并允许学生选课

教师客户端

控制选课开始结束时间

**（6）**

数据库添加可学生成绩并将允许学生察看成绩置数为1

教师录入成绩并允许血学生察看成绩

教师客户端

录入成绩和公布成绩

# 教务系统设计

## 3.1系统开发及运行环境

硬件平台：

* CPU：P41.8GHz。
* 内存：256MB以上。

软件平台：

* 操作系统：Windows XP。
* 数据库：MySQL 5.0。
* 开发工具包：JDK Version1.6.2。
* JSP服务器：Tomcat 6.0。
* 浏览器：IE5.0，推荐使用IE6.0。

分辨率：最佳效果1024×768像素。

## 3.2 系统主要功能要求

本系统针对学校中对学生的管理现状，通过网络对学生的信息进行有效管理。本系统主要实现如下功能：

* 系统采用人机对话方式，界面美观友好，信息查询灵活、方便、快捷、准确，数据存储安全可靠。
* 实现校园内一些基础信息的设置及查询。
* 学生信息的添加、删除、修改操作清晰。
* 数据计算自动完成，提高工作效率。
* 老师和学生之间的管理清晰。
* 以图表形式对学生信息进行分析。
* 实现多条件查询。
* 可方便快捷地查询学生信息，并可对学生信息进行调整。
* 操作员可以随时修改学生信息、课程信息、和学生成绩。
* 对用户输入的数据，系统进行严格的数据检验，尽可能排除人为的错误。
* 系统最大限度地实现了易维护性和易操作性。
* 系统运行稳定、安全可靠。

## 3.3 教务管理系统总体结构

学生教务管理系统是基于先进的软件和高速、大容量的硬件基础上的新的管理模式，通过集中式的信息数据库，达到数据共享，提高效率，改进服务等目的。通过第一章对学生管理内容和对学生管理系统的分析，一个标准的学生教务管理系统应该包括以下几大功能，如下图所示。

教务管理系统

教师客户端

学生客户端

修改密码

添加学生

学生信息

开课设置

课程成绩

增加班级

增加课程

修改密码

基本信息

选择课程

查看课表

成绩查询

# 4. 数据库设计

## 4.1安全性设计

数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄露、更改或破坏。系统安全保护措施是否有效是数据库系统的主要指标之一。数据库的安全性和计算机系统的安全性，包括操作系统、网络系统的安全性是紧密联系、相互支持的。

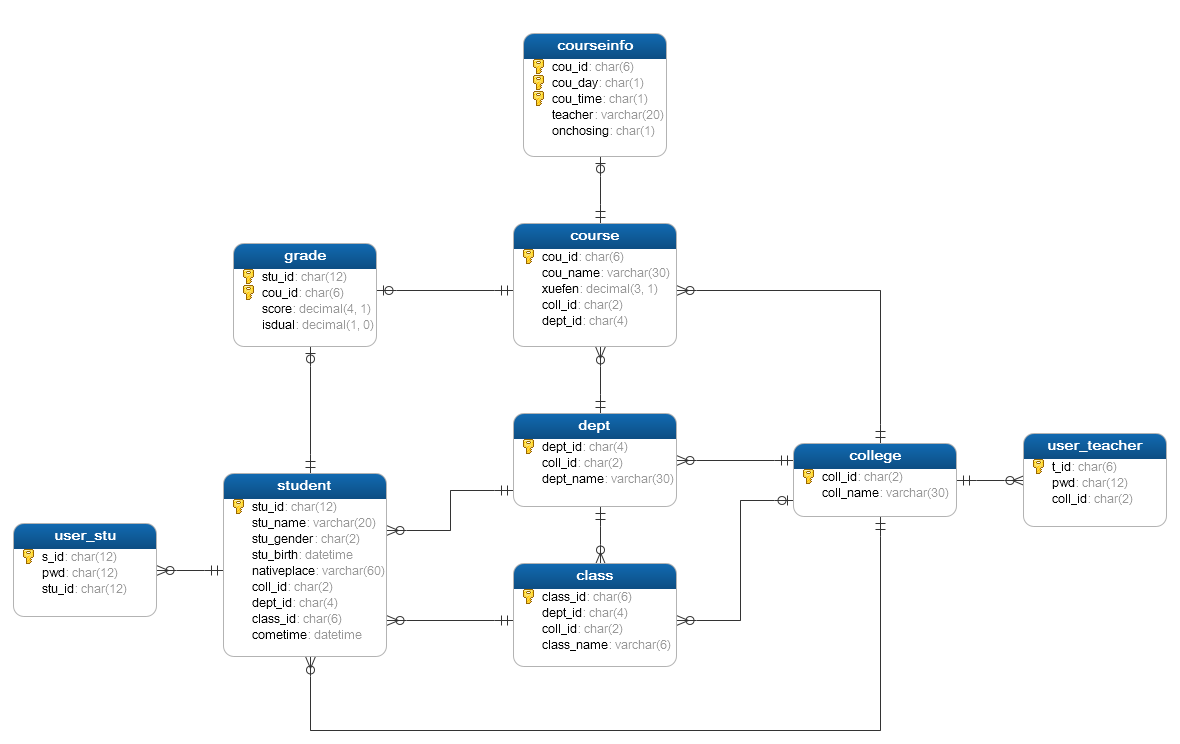
信息系统尽管功能强大，技术先进，但由于受到自身体系结构，设计思路以及运行机制 等限制，也隐含许多不安全因素。常见因素有：数据的输入，输出，存取与备份，源程序以及应用软件，数据库，操作系统等漏洞或缺陷，硬件，通信部分的漏洞，企业内部人员的因素，病毒，“黑客”等因素。

## 4.2设计过程

数据库设计是指对于一个给定的应用环境，构造最优的数据库模式，建立数据库应用系统，使之能有效的存储数据，满足各种用户的应用需求。数据库设计方法是研究数据库及其应用系统的技术，是数据库在应用领域中的主要的研究课题。

为了更好的设计数据库，人们提出了数据库设计规范，把数据库设计分成：需求分析（分析用户要求），概念设计（信息分析和定义），逻辑设计（设计和实现）和物理设计（物理数据库设计）。

## 4.3 E-R图与表



**表1班级信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可否为空 | 说明 |
| class\_id | char(6) | NOT NULL | 班级号 |
| dept\_id | char(4) | NOT NULL | 专业号 |
| coll\_id | char(2) | DEFAULT NULL | 学院号 |
| class\_name | varchar(6) | NOT NULL | 班级名 |

**表2学院信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可否为空 | 说明 |
| coll\_id | char(2) | NOT NULL | 学院号 |
| coll\_name | varchar(30) | NOT NULL | 学院名 |

**表3课程信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可否为空 | 说明 |
| cou\_id | char(6) | NOT NULL | 课程号 |
| cou\_name | varchar(30) | NOT NULL | 课程名 |
| xuefen | decimal(3,1) | NOT NULL | 学分 |
| coll\_id | char(2) | NOT NULL | 学院号 |
| dept\_id | char(4) | NOT NULL | 专业号 |

**表4课程详细信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可否为空 | 说明 |
| cou\_id | char(6) | NOT NULL DEFAULT '' | 课程号 |
| cou\_day | char(1) | NOT NULL DEFAULT '' | 课程日期 |
| cou\_time | char(1) | NOT NULL DEFAULT '' | 课程时间 |
| teacher | varchar(20) | NOT NULL | 老师 |
| onchosing | char(1) | DEFAULT '0' | 是否可选 |

**表5专业信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可否为空 | 说明 |
| dept\_id | char(4) | NOT NULL | 专业号 |
| coll\_id | char(2) | NOT NULL | 学院号 |
| dept\_name | varchar(30) | DEFAULT NULL | 专业名 |

**表6成绩信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可否为空 | 说明 |
| stu\_id | char(12) | NOT NULL DEFAULT '' | 学生学号 |
| cou\_id | char(6) | NOT NULL DEFAULT '' | 课程号 |
| score | decimal(4,1) | DEFAULT '0.0' | 分数 |
| isdual | decimal(1,0) | DEFAULT '0' | 是否可查看 |

**表7学生信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可否为空 | 说明 |
| stu\_id | char(12) | NOT NULL | 学生学号 |
| stu\_name | varchar(20) | NOT NULL | 学生姓名 |
| stu\_gender | char(2) | DEFAULT NULL | 学生性别 |
| stu\_birth | datetime | DEFAULT NULL | 学生生日 |
| nativeplace | varchar(60) | DEFAULT NULL | 故乡 |
| coll\_id | char(2) | NOT NULL | 学院号 |
| dept\_id | char(4) | NOT NULL | 专业号 |
| class\_id | char(6) | NOT NULL | 班级号 |
| cometime | datetime | NOT NULL | 入学时间 |

**表8学生用户信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可否为空 | 说明 |
| s\_id | char(12) | NOT NULL | 学生用户账号 |
| pwd | char(12) | NOT NULL | 密码 |
| stu\_id | char(12) | NOT NULL | 学生学号 |

**表9教师用户信息**

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 可否为空 | 说明 |
| t\_id | char(6) | NOT NULL | 教师用户账号 |
| pwd | char(12) | NOT NULL | 密码 |
| coll\_id | char(2) | NOT NULL | 学院号 |

## 4.4数据库部分源代码

DROP DATABASE IF EXISTS `test`;

CREATE DATABASE `test` /\*!40100 DEFAULT CHARACTER SET latin1 \*/;

USE `test`;

DROP TABLE IF EXISTS `class`;

CREATE TABLE `class` (

`class\_id` char(6) NOT NULL,

`dept\_id` char(4) NOT NULL,

`coll\_id` char(2) DEFAULT NULL,

`class\_name` varchar(6) NOT NULL,

PRIMARY KEY (`class\_id`),

KEY `class\_fk1` (`dept\_id`),

KEY `class\_fk2` (`coll\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

LOCK TABLES `class` WRITE;

INSERT INTO `class` VALUES ('010101','0101','01','计算机一班');

INSERT INTO `class` VALUES ('020201','0201','02','数学二班');

UNLOCK TABLES;

DROP TABLE IF EXISTS `college`;

CREATE TABLE `college` (

`coll\_id` char(2) NOT NULL,

`coll\_name` varchar(30) NOT NULL,

PRIMARY KEY (`coll\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

LOCK TABLES `college` WRITE;

INSERT INTO `college` VALUES ('01','信息学院');

INSERT INTO `college` VALUES ('02','理学院');

UNLOCK TABLES;

DROP TABLE IF EXISTS `course`;

CREATE TABLE `course` (

`cou\_id` char(6) NOT NULL,

`cou\_name` varchar(30) NOT NULL,

`xuefen` decimal(3,1) NOT NULL,

`coll\_id` char(2) NOT NULL,

`dept\_id` char(4) NOT NULL,

PRIMARY KEY (`cou\_id`),

KEY `cou\_fk1` (`dept\_id`),

KEY `cou\_fk2` (`coll\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

LOCK TABLES `course` WRITE;

INSERT INTO `course` VALUES ('010001','JAVA程序设计',4.0,'01','0101');

INSERT INTO `course` VALUES ('010002','计算机组成原理',4.0,'01','0101');

INSERT INTO `course` VALUES ('010003','计算机系统原理',4.0,'01','0101');

INSERT INTO `course` VALUES ('010004','C语言程序设计',4.0,'01','0101');

INSERT INTO `course` VALUES ('020001','线性代数',3,'02','0201');

UNLOCK TABLES;

DROP TABLE IF EXISTS `courseinfo`;

CREATE TABLE `courseinfo` (

`cou\_id` char(6) NOT NULL DEFAULT '',

`cou\_day` char(1) NOT NULL DEFAULT '',

`cou\_time` char(1) NOT NULL DEFAULT '',

`teacher` varchar(20) NOT NULL,

`onchosing` char(1) DEFAULT '0',

PRIMARY KEY (`cou\_id`,`cou\_day`,`cou\_time`)

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

LOCK TABLES `courseinfo` WRITE;

INSERT INTO `courseinfo` VALUES ('010001','1','2','许加兵','0');

INSERT INTO `courseinfo` VALUES ('010002','1','1','金国英','0');

INSERT INTO `courseinfo` VALUES ('010003','5','3','王玉巧','1');

INSERT INTO `courseinfo` VALUES ('010004','3','5','周碧涛','1');

UNLOCK TABLES;

DROP TABLE IF EXISTS `dept`;

CREATE TABLE `dept` (

`dept\_id` char(4) NOT NULL,

`coll\_id` char(2) NOT NULL,

`dept\_name` varchar(30) DEFAULT NULL,

PRIMARY KEY (`dept\_id`),

KEY `dept\_fk` (`coll\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

LOCK TABLES `dept` WRITE;

INSERT INTO `dept` VALUES ('0101','01','计算机专业');

INSERT INTO `dept` VALUES ('0201','02','数学专业');

/\*!40000 ALTER TABLE `dept` ENABLE KEYS \*/;

UNLOCK TABLES;

DROP TABLE IF EXISTS `grade`;

CREATE TABLE `grade` (

`stu\_id` char(12) NOT NULL DEFAULT '',

`cou\_id` char(6) NOT NULL DEFAULT '',

`score` decimal(4,1) DEFAULT '0.0',

`isdual` decimal(1,0) DEFAULT '0',

PRIMARY KEY (`stu\_id`,`cou\_id`),

KEY `grade\_fk1` (`cou\_id`)

) ENGINE=InnoDB DEFAULT CHARSET=gbk;

LOCK TABLES `grade` WRITE;

INSERT INTO `grade` VALUES ('514039927200','010001','99','0');

INSERT INTO `grade` VALUES ('514039927200','020001','96','0');

INSERT INTO `grade` VALUES ('514039927200','010002','58','0');

INSERT INTO `grade` VALUES ('514039927200','010003','96','1');

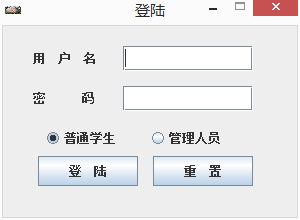
INSERT INTO `grade` VALUES ('514039927200','010004','59','1');

UNLOCK TABLES;

# 系统实现

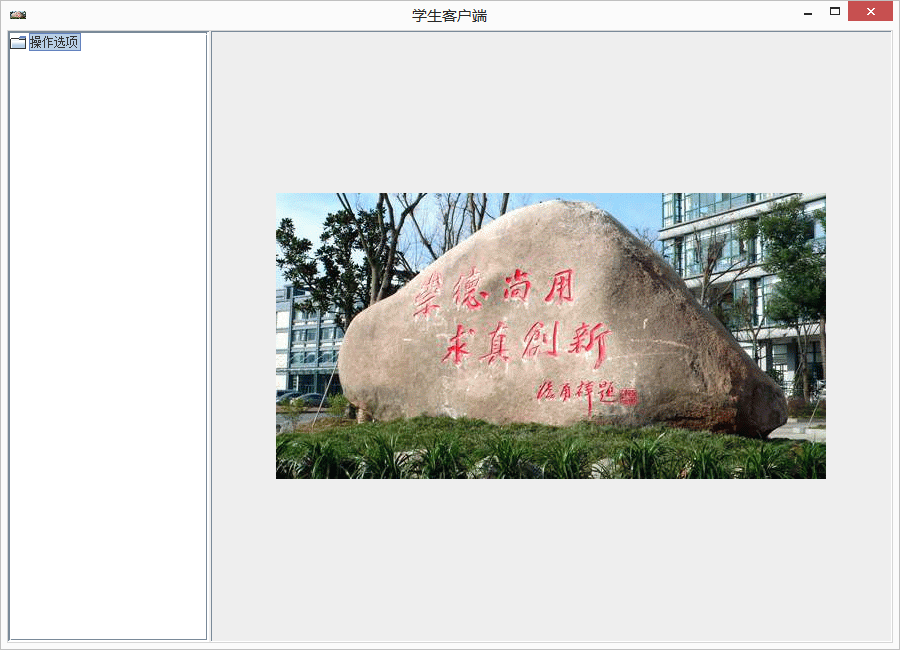
**5.1 界面设计**

**5.1.1 登陆窗口**

****

**图1 登陆界面**

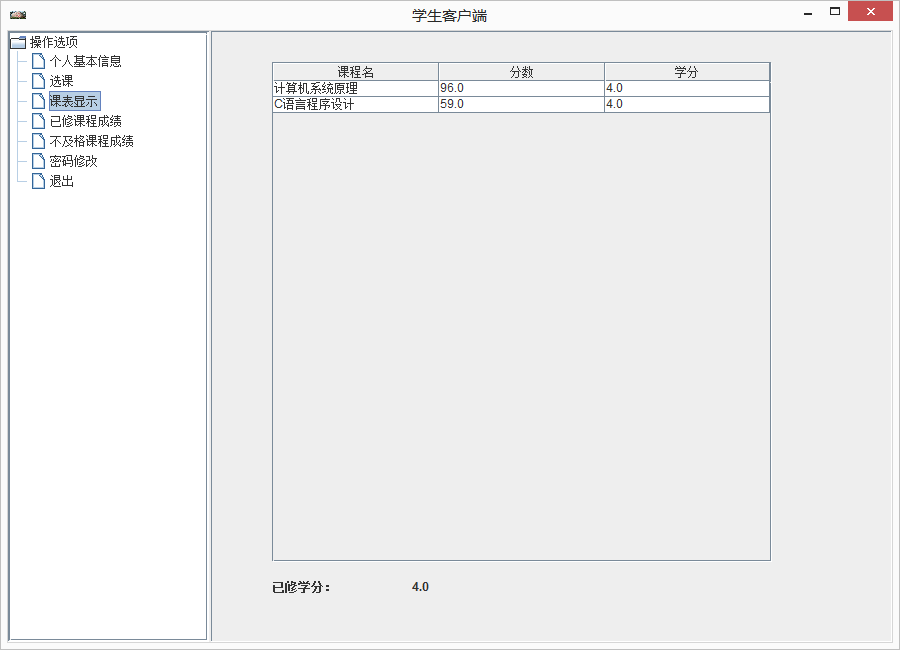
**5.1.2 学生客户端**

****

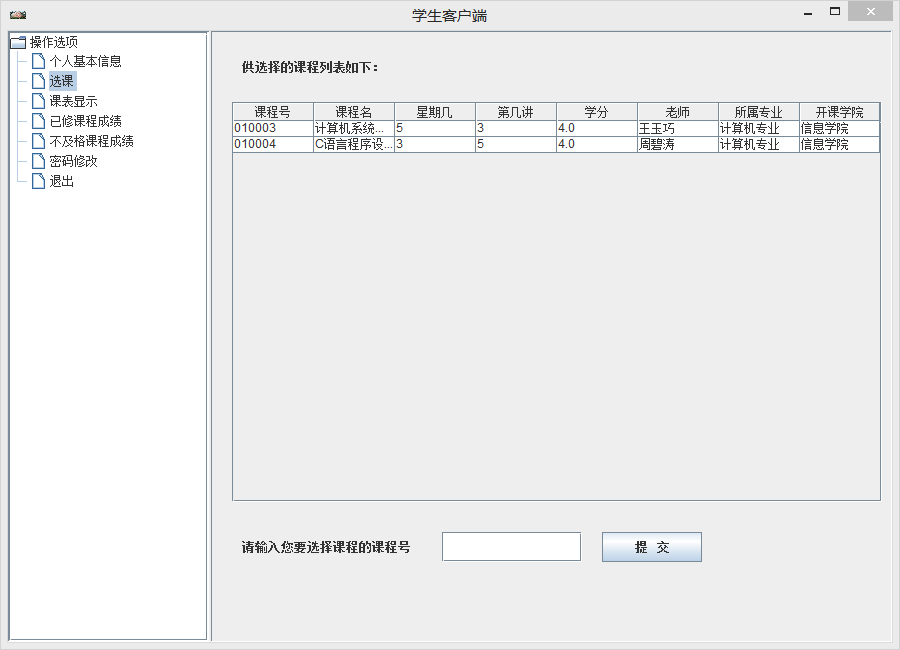
**图2 学生客户端主界面**

****

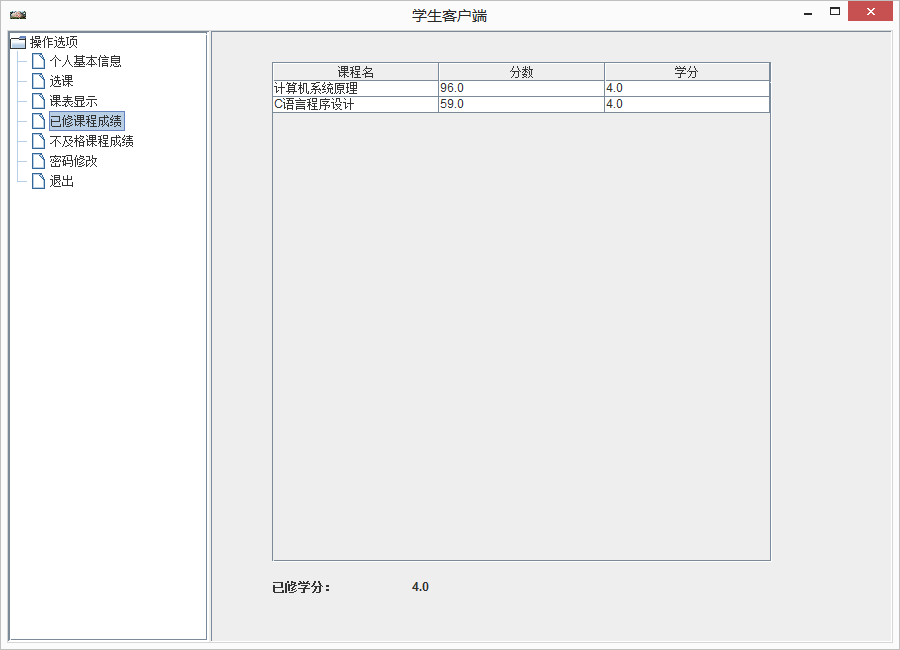
**图3 学生个人基本信息**



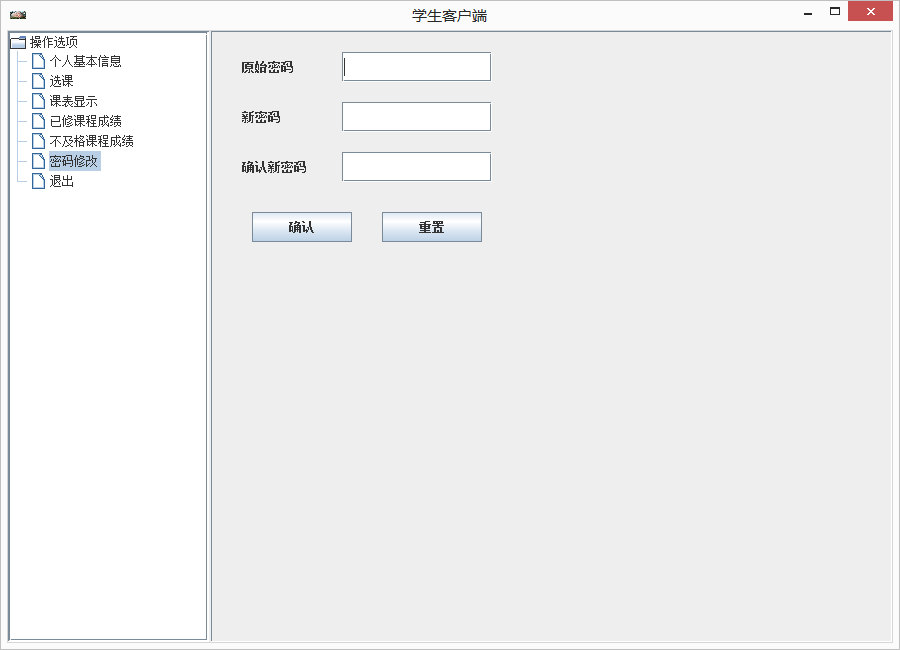
**图4 学生课表管理**



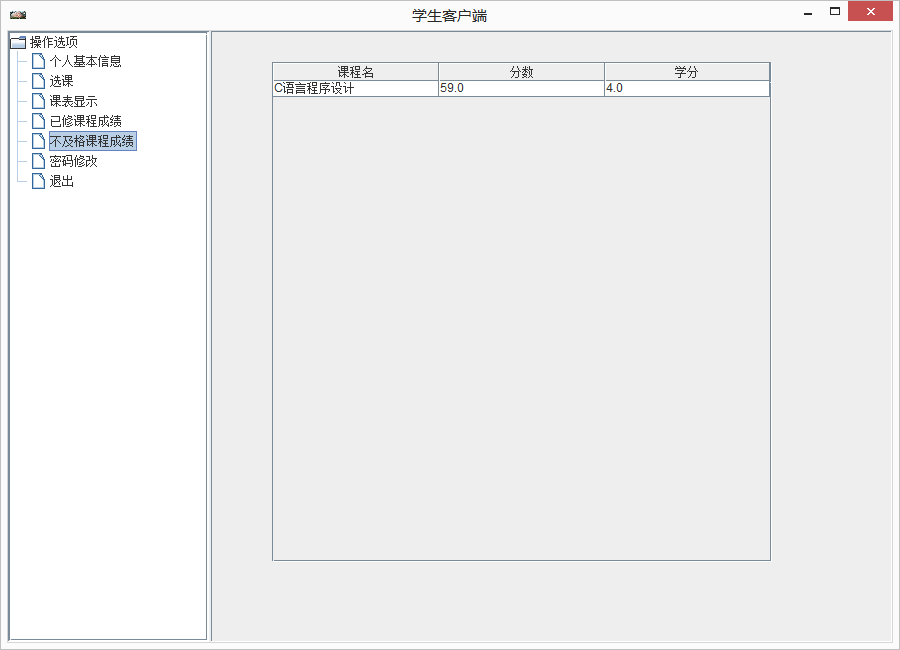
**图5 学生选课端口**



**图6 已选课程成绩**

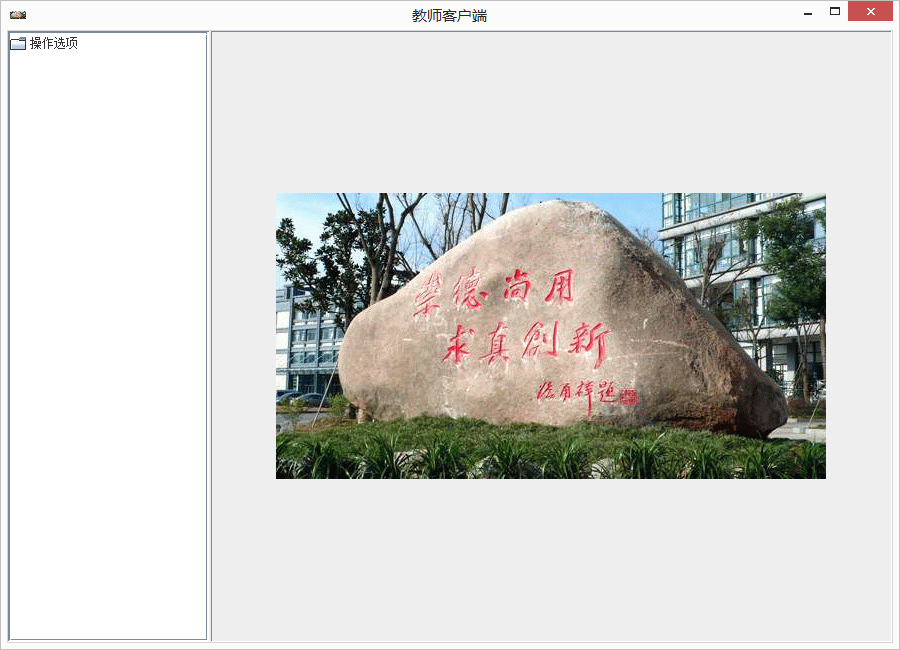


**图7 密码修改**



**图8 不及格课程成绩**

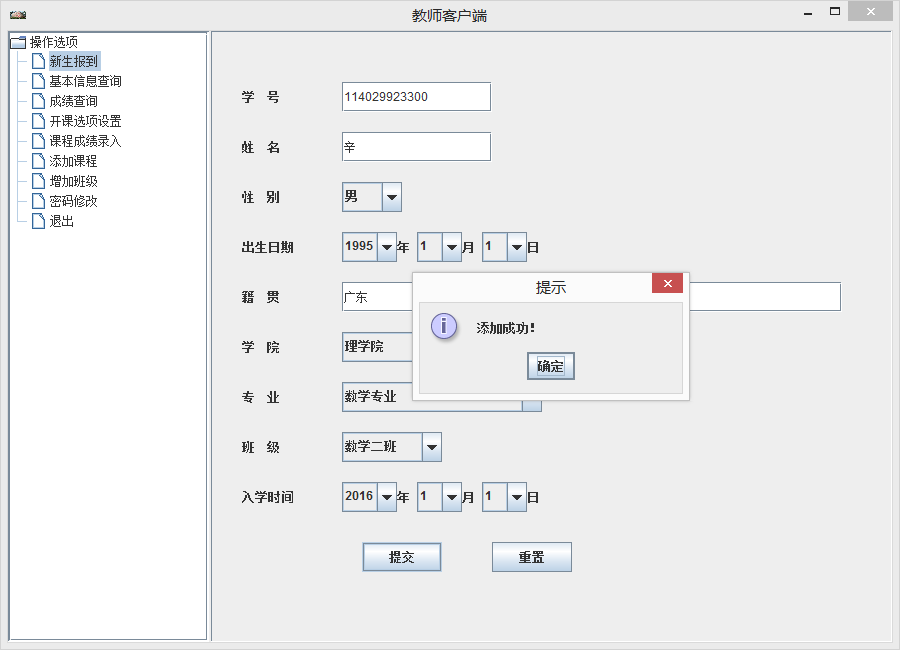
## 5.1.3教师客户端



**图1 教师客户端主页面**



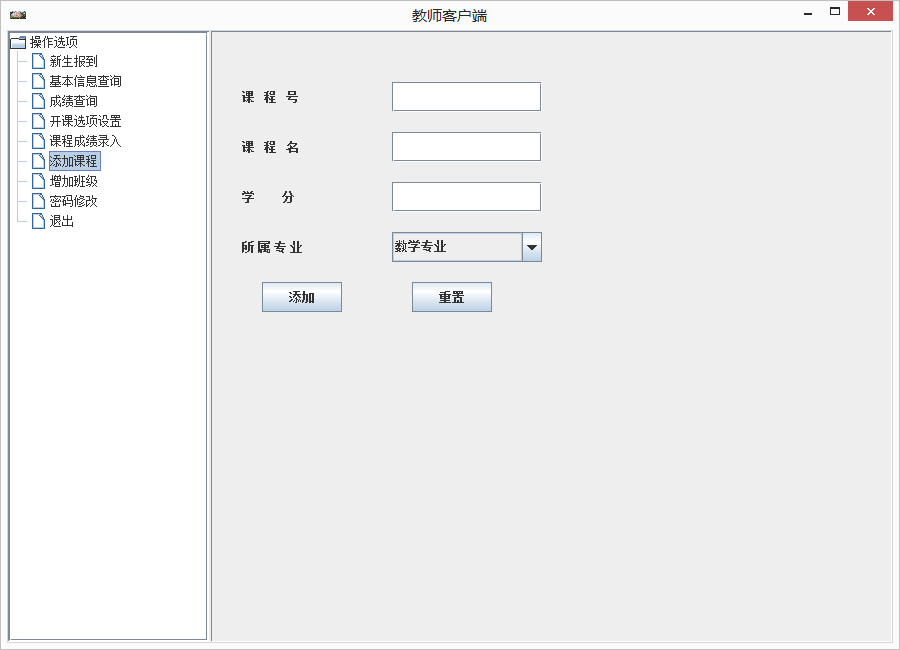
**图2 教师端学生基本信息查询**



**图3 添加学生**



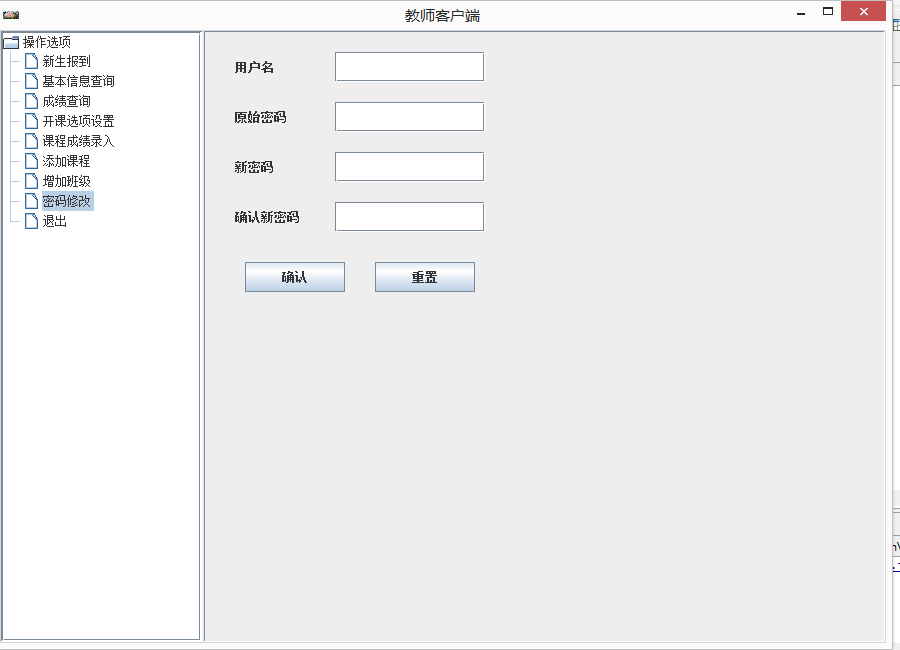
**图4 课程成绩录入**



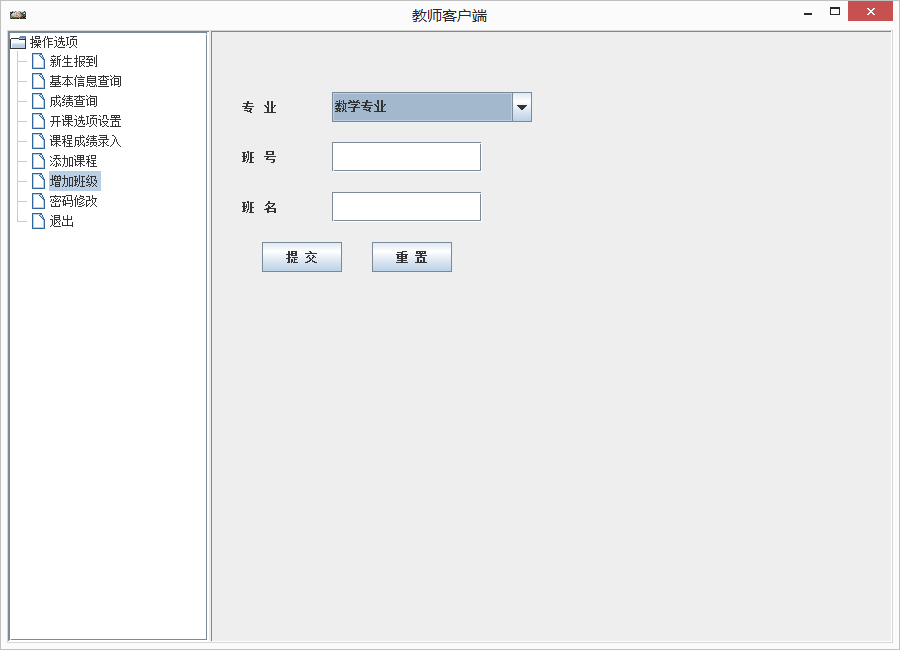
**图5 添加课程**



**图6 开课选项设置**



**图7 密码修改**



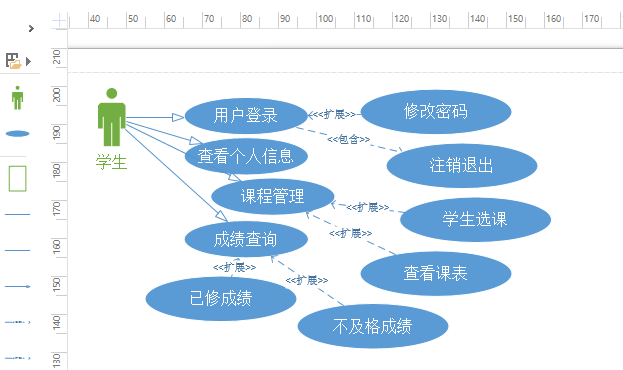
**图8添加班级**

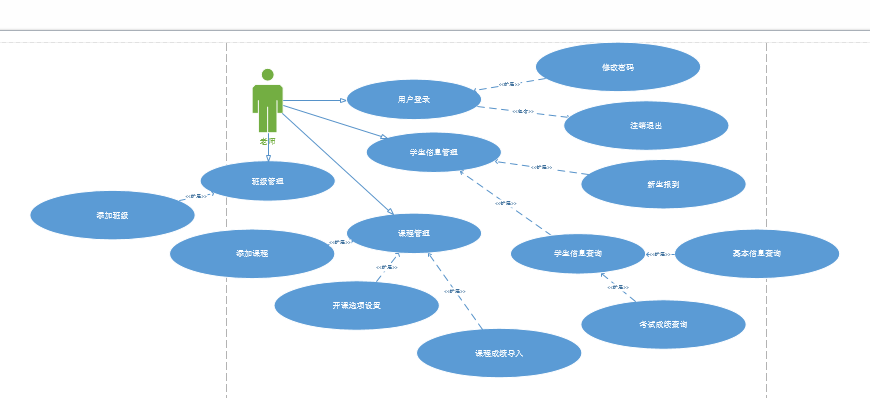


**图9 成绩查询**

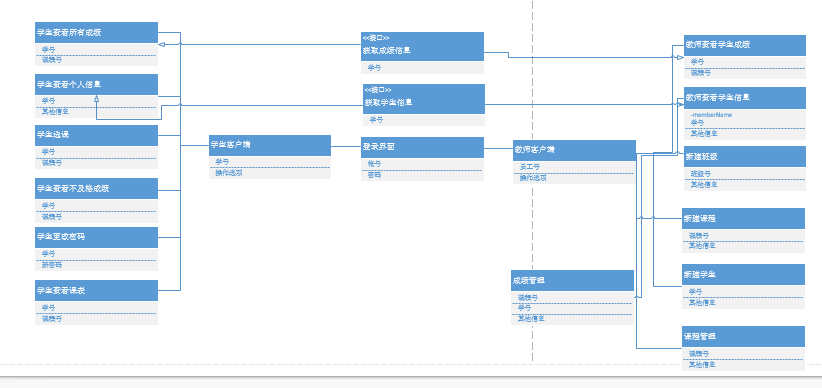
## 5.2 UML图

### 功能模型—用例图





### 对象模型—类图



## 5.3程序设计及调试运行

按功能模块结构设计菜单,布置各明细模块工作界面，编写各个模块的程序代码，进行编译连接运行，经过反复调试修改，以达到系统功能要求。这个过程具有大量的工作量，应仔细对每个程序细节进行分析思考、编写程序并调试，做到程序简洁清楚，尽量避免可能潜在的程序错误，并在必要的地方加以注释，以便于程序阅读和修改。

### 5.3.1 连接池

通常情况下，在每次访问数据库之前都要先建立与数据库的连接，这将消耗一定的资源，并延长了访问数据库的时间，如果访问量较高，将严重营销那个系统的性能。为了解决这一问题，引入了连接池的概念，所谓连接池，及时预先建立好一定数量的数据库连接，模拟存放在一个连接池中，由连接池负责对这些数据库连接进行管理，这样，当需要访问数据库时，就可以通过已经建立好的连接访问数据库了，从而免去了每次在访问数据库之前建立数据库连接的开销。

package com.zhangwu.dbconn;

import java.sql.\*;

import javax.naming.\*;

import javax.sql.DataSource;

public class DBResult {

public static Connection conn = null;

static{

try {

Context ctx = new InitialContext();

ctx = (Context) ctx.lookup("java:comp/env");

DataSource ds = (DataSource) ctx.lookup("jdbc/zhangwu"); // 获取连接池对象

try {

conn = ds.getConnection();

} catch (SQLException e) {

e.printStackTrace();

}

} catch (NamingException e) {

e.printStackTrace();

}

}

/\*\*

\* 用于获得执行SQL语句的ResultSet对象

\*/

public ResultSet getResult(String sql) {

try {

Statement stmt = conn.createStatement();

ResultSet rs = stmt.executeQuery(sql);

return rs;

} catch (Exception e) {

}

return null;

}

/\*\*

\* 用于执行SQL语句没有返回值

\*/

public void doExecute(String sql) {

try {

Statement stmt = conn.createStatement();

stmt.executeQuery(sql);

} catch (Exception e) {

}

}

/\*\*

\* 用于获得执行SQL语句的PreparedStatement(预处理)对象

\*/

public PreparedStatement getPreparedStatement(String sql) {

try {

PreparedStatement pstmt = conn.prepareStatement(sql);

return pstmt;

} catch (Exception e) {

}

return null;

}

}

在通过连接池技术访问数据库时，首先需要在Tomcat下配置数据库连接池：

1. 将MySql 数据库的3 个JDBC驱动包 mysql-connector-java-3.0.15-ga-bin.jar、commons-dbcp-1.2.1.jar放到Tomcat安装路径下的lib文件夹中。
2. 配置数据源。

### 5.3.2 JFreeChart

JFreeChart是一组功能强大、灵活易用的Java绘图API，使用它可以生成多种通用性的报表，包括柱状图、饼图、曲线图、甘特图等。它能够用在Swing和Web等中制作自定义的图表或报表，并且得到广泛的应用。本文将通过引领读者学习在JFreeChart中饼图、柱状图和曲线图的进阶应用，来达到熟练使用JFreeChart的目的。

## 5.2 程序源代码摘要

### 公共类 登陆

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.event.\*;

import java.util.\*;

import java.net.\*;

import java.io.\*;

import java.sql.\*;

import javax.sql.\*;

public class F\_Login extends JFrame implements ActionListener

{

private Connection conn;

private Statement stmt;

private ResultSet rs;

public String host="127.0.0.1:3306";

//创建用来存放空间的容器

private JPanel jp=new JPanel();

private JLabel jl1=new JLabel("用 户 名");

private JLabel jl2=new JLabel("密 码");

//正在登陆提示标签

private JLabel jl3=new JLabel("");

private JTextField jtf=new JTextField();

private JPasswordField jpwf=new JPasswordField();

private JRadioButton[] jrbArray=

{

new JRadioButton("普通学生",true),

new JRadioButton("管理人员")

};

private ButtonGroup bg=new ButtonGroup();

private JButton jb1=new JButton("登 陆");

private JButton jb2=new JButton("重 置");

public F\_Login()

{

this.addListener();

initialFrame();

}

public void addListener(){

this.jb1.addActionListener(this);//为登陆按钮注册监听器

this.jb2.addActionListener(this);//为重置按钮注册监听器

this.jtf.addActionListener(this);//为用户名文本框注册监听器

this.jpwf.addActionListener(this);//为用户名密码框注册监听器

}

public void initialFrame()

{

jp.setLayout(null);

this.jl1.setBounds(30,20,110,25);

this.jp.add(jl1);

this.jtf.setBounds(120,20,130,25);

this.jp.add(jtf);

this.jl2.setBounds(30,60,110,25);

this.jp.add(jl2);

this.jpwf.setBounds(120,60,130,25);

this.jpwf.setEchoChar('\*');

this.jp.add(jpwf);

this.bg.add(jrbArray[0]);

this.bg.add(jrbArray[1]);

this.jrbArray[0].setBounds(40,100,100,25);

this.jp.add(jrbArray[0]);

this.jrbArray[1].setBounds(145,100,100,25);

this.jp.add(jrbArray[1]);

this.jb1.setBounds(35,130,100,30);

this.jp.add(jb1);

this.jb2.setBounds(150,130,100,30);

this.jp.add(jb2);

this.jl3.setBounds(40,170,150,25);

this.jp.add(jl3);

this.add(jp);

this.setTitle("登陆");

Image image=new ImageIcon("ico.gif").getImage();

this.setIconImage(image);

this.setResizable(false);

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

int centerX=screenSize.width/2;

int centerY=screenSize.height/2;

int w=300;//本窗体宽度

int h=220;//本窗体高度

this.setBounds(centerX-w/2,centerY-h/2-100,w,h);//设置窗体出现在屏幕中央

this.setVisible(true);

this.jtf.requestFocus(true);

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==this.jb1)

{

String name=this.jtf.getText().trim();

if(name.equals("")){

JOptionPane.showMessageDialog(this,"请输入用户名","错误",

JOptionPane.ERROR\_MESSAGE);

jl3.setText("");return;

}

String pwd=this.jpwf.getText().trim();

if(pwd.equals("")){

JOptionPane.showMessageDialog(this,"请输入密码","错误",

JOptionPane.ERROR\_MESSAGE);

jl3.setText("");return;

}

int type=this.jrbArray[0].isSelected()?0:1;//获取登陆类型

try{

this.initialConnection();

if(type==0){//普通学生登陆

String sql="select \* from user\_stu where "+

"s\_id='"+name+"' and pwd='"+pwd+"'";

rs=stmt.executeQuery(sql);

if(rs.next()){

new S\_StuClient(name,host);//创建学生客户短窗口

this.dispose();//关闭登陆窗口并释放资源

}

else{

JOptionPane.showMessageDialog(this,"用户名或密码错误","错误",

JOptionPane.ERROR\_MESSAGE);

jl3.setText("");

}

this.closeConn();

}

else{//教师登陆

String sql="select coll\_id from user\_teacher where "+

"t\_id='"+name+"' and pwd='"+pwd+"'";

rs=stmt.executeQuery(sql);

if(rs.next()){

String coll\_id=rs.getString(1);

new T\_TeacherClient(coll\_id,host);

this.dispose();

}

else{

JOptionPane.showMessageDialog(this,"用户名或密码错误","错误",

JOptionPane.ERROR\_MESSAGE);

jl3.setText("");

}

this.closeConn();

}

}

catch(SQLException ea){ea.printStackTrace();}

}

else if(e.getSource()==this.jb2){//按下重置按钮,清空输入信息

this.jtf.setText("");

this.jpwf.setText("");

}

else if(e.getSource()==jtf){//当输入用户名并回车时

this.jpwf.requestFocus(true);

}

else if(e.getSource()==jpwf){//当输入密码并回车时

this.jb1.requestFocus(true);

}

}

public void initialConnection()

{

try

{

Class.forName("org.gjt.mm.mysql.Driver");

conn=DriverManager.getConnection("jdbc:mysql://"+host+"/test","root","dfsa4556");

stmt=conn.createStatement();

}

catch(SQLException e)

{

JOptionPane.showMessageDialog(this,"连接失败，请检查主机地址是否正确","错误",JOptionPane.ERROR\_MESSAGE);

e.printStackTrace();

}

catch(ClassNotFoundException e)

{

e.printStackTrace();

}

}

public void closeConn()

{

try

{

if(rs!=null)

{

rs.close();

}

if(stmt!=null)

{

stmt.close();

}

if(conn!=null)

{

conn.close();

}

}

catch(SQLException e)

{

e.printStackTrace();

}

}

public static void main(String args[])

{

F\_Login login=new F\_Login();

}

}

### 学生类 客户端界面

import java.util.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.event.\*;

import java.net.\*;

import java.io.\*;

import javax.swing.tree.\*;

public class S\_StuClient extends JFrame

{

private String host;

private String stu\_id;

private DefaultMutableTreeNode dmtnRoot=new DefaultMutableTreeNode(new MyNode("操作选项","1"));

private DefaultMutableTreeNode dmtn11=new DefaultMutableTreeNode(new MyNode("个人基本信息","11"));

private DefaultMutableTreeNode dmtn12=new DefaultMutableTreeNode(new MyNode("选课","12"));

private DefaultMutableTreeNode dmtn13=new DefaultMutableTreeNode(new MyNode("课表显示","13"));

private DefaultMutableTreeNode dmtn14=new DefaultMutableTreeNode(new MyNode("已修课程成绩","14"));

private DefaultMutableTreeNode dmtn15=new DefaultMutableTreeNode(new MyNode("不及格课程成绩","15"));

private DefaultMutableTreeNode dmtn19=new DefaultMutableTreeNode(new MyNode("密码修改","19"));

private DefaultMutableTreeNode dmtn20=new DefaultMutableTreeNode(new MyNode("退出","20"));

private DefaultTreeModel dtm=new DefaultTreeModel(dmtnRoot);

private JTree jtree=new JTree(dtm);

private JScrollPane jspz=new JScrollPane(jtree);

private JPanel jpy=new JPanel();

private JSplitPane jsp1=new JSplitPane(JSplitPane.HORIZONTAL\_SPLIT,jspz,jpy);

private CardLayout cl;

private F\_Welcome welcome;

private S\_ChoseCourse chosecourse;

private S\_CourseTable coursetable;

private S\_StuGrade stugrade;

private S\_StuFailGrade stufailgrade;

private S\_StuInfo stuinfo;

private S\_ChangePwd changepwd;

public S\_StuClient(String stu\_id,String host)

{

this.host=host;

this.stu\_id=stu\_id;

this.initialTree();

this.initialPane();

this.initialJpy();

this.addListener();

this.initialFrame();

}

public void initialJpy()

{

jpy.setLayout(new CardLayout());

cl=(CardLayout)jpy.getLayout();

jpy.add(welcome,"welcome");

welcome.setBackground(Color.red);

jpy.add(welcome,"welcome");

jpy.add(chosecourse,"chosecourse");

jpy.add(coursetable,"coursetable");

jpy.add(stugrade,"stugrade");

jpy.add(stufailgrade,"stufailgrade");

jpy.add(stuinfo,"stuinfo");

jpy.add(changepwd,"changepwd");

}

public void initialPane()

{

welcome=new F\_Welcome("学生成绩管理系统");

chosecourse=new S\_ChoseCourse(stu\_id,host);

coursetable=new S\_CourseTable(stu\_id,host);

stugrade=new S\_StuGrade(stu\_id,host);

stufailgrade=new S\_StuFailGrade(stu\_id,host);

stuinfo=new S\_StuInfo(stu\_id,host);

changepwd=new S\_ChangePwd(stu\_id,host);

}

public void initialTree()

{

dmtnRoot.add(dmtn11);

dmtnRoot.add(dmtn12);

dmtnRoot.add(dmtn13);

dmtnRoot.add(dmtn14);

dmtnRoot.add(dmtn15);

dmtnRoot.add(dmtn19);

dmtnRoot.add(dmtn20);

jtree.setToggleClickCount(1);

}

public void addListener()

{

jtree.addMouseListener(

new MouseAdapter()

{

public void mouseClicked(MouseEvent e)

{

DefaultMutableTreeNode dmtntemp=(DefaultMutableTreeNode)jtree.getLastSelectedPathComponent();

MyNode mynode=(MyNode)dmtntemp.getUserObject();

String id=mynode.getId();

if(id.equals("1"))

{

cl.show(jpy,"welcome");

}

else if(id.equals("20"))

{

int i=JOptionPane.showConfirmDialog(jpy,"您确认要退出出系统吗？","询问",JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

if(i==0)

{

System.exit(0);

}

}

else if(id.equals("19"))

{

cl.show(jpy,"changepwd");

changepwd.setFocus();

}

else if(id.equals("11"))

{

cl.show(jpy,"stuinfo");

}

else if(id.equals("12"))

{

cl.show(jpy,"chosecourse");

}

else if(id.equals("13"))

{

cl.show(jpy,"coursetable");

coursetable.initialData();

coursetable.updataview();

}

else if(id.equals("14"))

{

cl.show(jpy,"stugrade");

}

else if(id.equals("15"))

{

cl.show(jpy,"stufailgrade");

}

}

}

);

}

public void initialFrame()

{

this.add(jsp1);

jsp1.setDividerLocation(200);

jsp1.setDividerSize(4);

this.setTitle("学生客户端");

Image image=new ImageIcon("ico.gif").getImage();

this.setIconImage(image);

Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();

int centerX=screenSize.width/2;

int centerY=screenSize.height/2;

int w=900;

int h=650;

this.setBounds(centerX-w/2,centerY-h/2-30,w,h);

this.setVisible(true);

}

class MyNode

{

private String values;

private String id;

public MyNode(String values,String id)

{

this.values=values;

this.id=id;

}

public String toString()

{

return this.values;

}

public String getId()

{

return this.id;

}

}

}

### 教师类 管理成绩

import java.util.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import javax.swing.event.\*;

import java.net.\*;

import java.io.\*;

import javax.swing.tree.\*;

import java.sql.\*;

import javax.sql.\*;

import javax.swing.table.\*;

public class T\_GradeManage extends JPanel implements ActionListener

{

private String host;

private Connection conn;

private Statement stmt;

private ResultSet rs;

//存放当前教师所在的学院号

private String coll\_id;

private Map map\_dept=new HashMap();

private Vector v\_dept=new Vector();

private JComboBox jcb=new JComboBox(v\_dept);

private JLabel jl=new JLabel("请选择您要操作的课程");

//创建存放表头及表格数据的Vector对象

private Vector v\_head=new Vector();

private Vector v\_data=new Vector();

private JTable jt;//声名表格引用

private JScrollPane jsp;

//创建动作按钮，只有公布后的成绩学生才可以看到

private JButton jb=new JButton("公布该科成绩单");

public T\_GradeManage(String coll\_id,String host)

{

this.host=host;

this.coll\_id=coll\_id;

this.initialData();

this.initialFrame();

this.initialListener();

}

public void initialFrame()

{

this.setLayout(null);

jl.setBounds(30,20,150,30);this.add(jl);

jcb.setBounds(180,20,100,30);this.add(jcb);

jb.setBounds(350,20,150,30);this.add(jb);

jt=new JTable(new DefaultTableModel(v\_data,v\_head));

jsp=new JScrollPane(jt);

jsp.setBounds(30,70,500,500);this.add(jsp);

}

public void initialListener()

{

jcb.addActionListener(this);jb.addActionListener(this);

TableChangeListener tl=new TableChangeListener(stmt);

jt.getSelectionModel().addListSelectionListener(tl);

jt.getColumnModel().addColumnModelListener(tl);

jt.getModel().addTableModelListener(tl);

}

public void initialData()

{

//初始化表头

v\_head.add("课程号");v\_head.add("学号");

v\_head.add("姓名");v\_head.add("成绩(分)");

String sql="select distinct cou\_name,course.cou\_id from course,grade where"+

" course.coll\_id='"+coll\_id+"' and course.cou\_id=grade.cou\_id and isdual=0";

try

{//查询数据库，将课程名及课程号存入map\_dept与v\_dept中

this.initialConnection();

rs=stmt.executeQuery(sql);

while(rs.next()){

String cou\_name=new String(rs.getString(1).getBytes("gb2312"));

String cou\_id=rs.getString(2);

map\_dept.put(cou\_name,cou\_id);

v\_dept.add(cou\_name);

}

rs.close();

}

catch(SQLException e){e.printStackTrace();}

catch(UnsupportedEncodingException e){e.printStackTrace();}

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==jcb)

{//当下拉列表框中的选择内容发生变化时的处理代码

v\_data.removeAllElements();//将v\_data清空

//获得下拉列表选中的课程名

String cur\_cou\_name=(String)jcb.getSelectedItem();

//根据课程名获得课程号

String cur\_cou\_id=(String)map\_dept.get(cur\_cou\_name);

String sql="select grade.cou\_id,grade.stu\_id,student.stu\_name,score "+

"from grade,student where grade.stu\_id=student.stu\_id and "+

"isdual=0 and grade.cou\_id='"+cur\_cou\_id+"'";

try{

rs=stmt.executeQuery(sql);

while(rs.next()){//将与该课程号相关的未处理的信息存入v\_data

Vector v=new Vector();

String cou\_id=rs.getString(1);

String stu\_id=rs.getString(2);

String stu\_name=new String(rs.getString(3).getBytes("gb2312"));

String score=rs.getDouble(4)+"";

v.add(cou\_id);v.add(stu\_id);v.add(stu\_name);v.add(score);

v\_data.add(v);

}

rs.close();

DefaultTableModel temp1=(DefaultTableModel)jt.getModel();//更新表格模型，

temp1.setDataVector(v\_data,v\_head);

temp1.fireTableStructureChanged();//更新显示信息

}

catch(Exception ea){ea.printStackTrace();}

}

else if(e.getSource()==jb){//当按下公布成绩的按钮时

try{//获得要公布成绩的课程名

String cur\_cou\_name=(String)jcb.getSelectedItem();

if(cur\_cou\_name!=null){

String cur\_cou\_id=(String)map\_dept.get(cur\_cou\_name);

String sql="update grade set isdual=1 where "+

"cou\_id='"+cur\_cou\_id+"' and isdual=0";

int i=stmt.executeUpdate(sql);

}

else{

JOptionPane.showMessageDialog(this,"请先选择课程名称","错误",

JOptionPane.ERROR\_MESSAGE);

}

}

catch(Exception ea){ea.printStackTrace();}

}

}

public void initialConnection()

{

try

{

Class.forName("org.gjt.mm.mysql.Driver");

conn=DriverManager.getConnection("jdbc:mysql://"+host+"/test","root","dfsa4556");

stmt=conn.createStatement();

}

catch(SQLException e)

{

e.printStackTrace();

}

catch(ClassNotFoundException e)

{

e.printStackTrace();

}

}

public void closeConn()

{

try

{

if(rs!=null)

{

rs.close();

}

if(stmt!=null)

{

stmt.close();

}

if(conn!=null)

{

conn.close();

}

}

catch(SQLException e)

{

e.printStackTrace();

}

}

class TableChangeListener implements ListSelectionListener,

TableModelListener,TableColumnModelListener

{

int rowNum,colNum;

Statement statement;

public TableChangeListener(Statement statement)

{this.statement=statement;}

public void valueChanged(ListSelectionEvent e){//更新行的值

rowNum=jt.getSelectedRow();

}

public void columnSelectionChanged(ListSelectionEvent e){//更新列的值

colNum=jt.getSelectedColumn();

}

public void tableChanged(TableModelEvent e)

{

//当更改的是第三列(分数列)时的处理代码

if(colNum==3)

{ //获得输入的数据

String str=(String)jt.getValueAt(rowNum,colNum);

//获得该数据所对应的课程号

String cou\_id=(String)jt.getValueAt(rowNum,0);

//获得该数据所对应学生的学号

String stu\_id=(String)jt.getValueAt(rowNum,1);

try{//将str转化为Double

Double d=Double.parseDouble(str);

if(d<0||d>100)

{

jt.setValueAt("0",rowNum,colNum);

}

}

catch(Exception ea){//不是数字，将其设为0

jt.setValueAt("0",rowNum,colNum);

}

//将更改的数据同步到数据库

String sql="update grade set score="+str+" where "+

"cou\_id='"+cou\_id+"' and stu\_id='"+stu\_id+"'";

try{

int i=statement.executeUpdate(sql);

}

catch(Exception ea){ea.printStackTrace();}

}

}

//实现接口中的其他方法

public void columnMoved(TableColumnModelEvent e){}

public void columnRemoved(TableColumnModelEvent e){}

public void columnMarginChanged(ChangeEvent e){}

public void columnAdded(TableColumnModelEvent e){}

}

public static void main(String args[])

{

T\_GradeManage gi=new T\_GradeManage("01","127.0.0.1:3306");

JFrame jf=new JFrame();

jf.setBounds(10,10,700,650);

jf.add(gi);

jf.setVisible(true);

}

}

# 7. 结题感悟

在这个项目中，我们遇到了很多困难，也学到了许多知识。

* 用Xmind画思维导图做需求分析。
* 用Axure做原形设计（由于界面简单，并未在此花太多时间）。
* 体会到原形设计的重要性。
* 用Viso画UML图。
* 用navicat for mySQL画ER图。
* 增强了对数据库和数据表的理解。
* 增强了对JDBC编程的理解。
* 增强了对正则表达式的理解。
* 增强了对异常处理的理解。
* 增强了对图形用户界面的理解。
* 增强了对监听器的理解。
* 学会了利用java创建树表。
* 学会了利用java创建表格。
* 学会了利用java制作各类选项按钮。
* 明白了绝对路径和相对路径的差异。
* 学到了一个系统的项目的制作过程。
* 增加了团队合作能力。