**学生管理系统**

目录

[**0 java学生管理系统** **2**](#_Toc1462217568_WPSOffice_Level1)

[**该系统主要功能有：** **2**](#_Toc337277851_WPSOffice_Level1)

[**0.1学生基本信息和学生成绩的管理：** **2**](#_Toc1419497324_WPSOffice_Level1)

[**0.2成绩的统计汇总：** **3**](#_Toc1095689945_WPSOffice_Level1)

[**0.3成绩信息查询：** **3**](#_Toc588632590_WPSOffice_Level1)

[**0.4系统登录功能：** **3**](#_Toc1838262048_WPSOffice_Level1)

[**0.5教师管理模块：** **3**](#_Toc1970494994_WPSOffice_Level1)

[**0.6成绩管理模块：** **4**](#_Toc1764043771_WPSOffice_Level1)

[**一、软件配置** **4**](#_Toc124428715_WPSOffice_Level1)

[**二、程序运行环境配置** **4**](#_Toc1771824474_WPSOffice_Level1)

[**三、 系统管理模块** **7**](#_Toc2045685216_WPSOffice_Level1)

[3.1登陆 7](#_Toc337277851_WPSOffice_Level2)

[3.2注册 8](#_Toc1419497324_WPSOffice_Level2)

[3.3修改密码 8](#_Toc1095689945_WPSOffice_Level2)

[3.4退出登陆 9](#_Toc588632590_WPSOffice_Level2)

[**四、 教师管理模块** **10**](#_Toc618236842_WPSOffice_Level1)

[4.1添加学生 10](#_Toc1838262048_WPSOffice_Level2)

[4.2查询学生 11](#_Toc1970494994_WPSOffice_Level2)

[4.3修改学生 15](#_Toc1764043771_WPSOffice_Level2)

[4.4删除学生 17](#_Toc124428715_WPSOffice_Level2)

[**五、 成绩管理模块** **19**](#_Toc1180719308_WPSOffice_Level1)

[5.1成绩添加 20](#_Toc1771824474_WPSOffice_Level2)

[5.2成绩修改 23](#_Toc2045685216_WPSOffice_Level2)

[5.3成绩查询 24](#_Toc618236842_WPSOffice_Level2)

[5.4成绩统计 26](#_Toc1180719308_WPSOffice_Level2)

[**六、 帮助模块** **28**](#_Toc1600511276_WPSOffice_Level1)

[6.1获得可帮助的网页内容 28](#_Toc1600511276_WPSOffice_Level2)

[**七、 创建数据库表格的SQL语句** **28**](#_Toc412853410_WPSOffice_Level1)

[**八、 程序具体代码** **33**](#_Toc307598413_WPSOffice_Level1)

[8.1连接数据库的代码 33](#_Toc412853410_WPSOffice_Level2)

[8.2创建数据库语句的代码 63](#_Toc307598413_WPSOffice_Level2)

[8.3获得当前年的代码 65](#_Toc813388962_WPSOffice_Level2)

[8.4设置窗口大小及位置代码 66](#_Toc1894871179_WPSOffice_Level2)

[8.6 管理员学生信息类和构造方法 67](#_Toc2064904090_WPSOffice_Level2)

[8.6 各类窗口的代码 77](#_Toc1507305110_WPSOffice_Level2)

[8.7 主程序代码 170](#_Toc1559883758_WPSOffice_Level2)

**0 java学生管理系统**

**该系统主要功能有：**

**0.1学生基本信息和学生成绩的管理：**

学生基本信息：姓名、学号、专业、班级、所属院系。

各科成绩：高等数学、英语、数据结构、计算机组成原理、面向对象程序设计

系统可以完成对各类信息的添加、修改、删除等功能。

**0.2成绩的统计汇总：**

可以计算班级平均成绩，某科最高成绩，最低成绩，优、良、中、及格、不及格各个成绩等级的人数，每位同学的总成绩、班级总成绩，总成绩排名等。

**0.3成绩信息查询：**

查询功能是系统的核心功能之一。在系统中既有单条件查询也有多条件查询，可以精确查询和模糊查询。可以按照班级、成绩范围、科目等查询。可以自己根据需要合理设计。

**0.4系统登录功能：**

进入系统，需要登录。可以注册用户，有完整的用户注册、用户登录和密码修改功能。

系统管理模块：因为本系统是用来管理学生成绩的，所以系统运行的正常和安全性非常重要，为了达到此目的本系统需要一个管理系统安全的系统管理模块。其任务是维护系统的正常运行和安全性设置。该模块的主要任务是维护系统的正常运行和安全性设置包括注册、登录、修改密码、退出登录等等。

**0.5教师管理模块：**

该模块的主要任务是维护系统的正常运行和安全性设置，实现对学生的个人信息的管理工作，对学生信息进行维护更新，从而方便学校管理部门对学校的基本情况的快速查询和了解。包括添加学生信息、查询和修改学生信息、删除学生信息等等。

**0.6成绩管理模块：**

因为本系统需要对学生成绩进行统计、分析。这也是本系统的一个重要的作用，所以有必要专门构建一个管理学生成绩的模块。方便教师等快速把握学生、班级、学校的学习状况。学校的成绩管理工作是检验学生学习情况的一个主要手段，本模块包括成绩添加、成绩修改、成绩查询、成绩的统计等功能模块。

**一、软件配置**

1、系统:Mac os系统 ，windows 系统，linux系统都适用

2、数据库版本：MYSQL 8.0.20

3、数据库桥接jar包版本信息: mysql-connector-java-8.0.20.jar

4、连接数据库的可视化工具:Navicat Premium

5、编辑数据库语句软件：sublime Text

6、编写java代码程序：eclipse

7、编码均为UTF-8编码

**二、程序运行环境配置**

1、数据库修改（仅MySQL5.7.5以上版本数据库需要更改）

MySQL 5.7.5及以上功能依赖检测功能。如果启用了ONLY\_FULL\_GROUP\_BY SQL模式（默认情况下），MySQL将拒绝选择列表，HAVING条件或ORDER BY列表的查询引用在GROUP BY子句中既未命名的非集合列，也不在功能上依赖于它们。（5.7.5之前，MySQL没有检测到功能依赖关系，默认情况下不启用ONLY\_FULL\_GROUP\_BY。有关5.7.5之前的行为的说明，请参见“MySQL 5.6参考手册”。

修改方式：在已经连接的Navicat查询语句或者是终端的MySQL中输入

SET GLOBAL sql\_mode=(SELECT REPLACE(@@sql\_mode,‘ONLY\_FULL\_GROUP\_BY’,’’));

即可解决

2、创建数据库，以及创建所需要的表根据数据库语句

3、导入项目，更改相应连接数据库的信息

新版MYSQL桥接方式中，比旧版需多加cj

旧版com.mysql.jdbc.Driver

新版con.mysql.cj.jdbc.Driver

URL后StudentSystemDao改为对应的数据库名称

USERNAME依旧用root用户

PASSWORD即root用户的密码

// 新版mysql要加cj连接方式

String ***DRIVER*** = "com.mysql.cj.jdbc.Driver";

// url格式

String ***URL*** = "jdbc:mysql://localhost:3306/StudentSystemDao";

// 用户名称

String ***USERNAME*** = "root";

// 用户密码

String ***PASSWORD*** = "12345678";

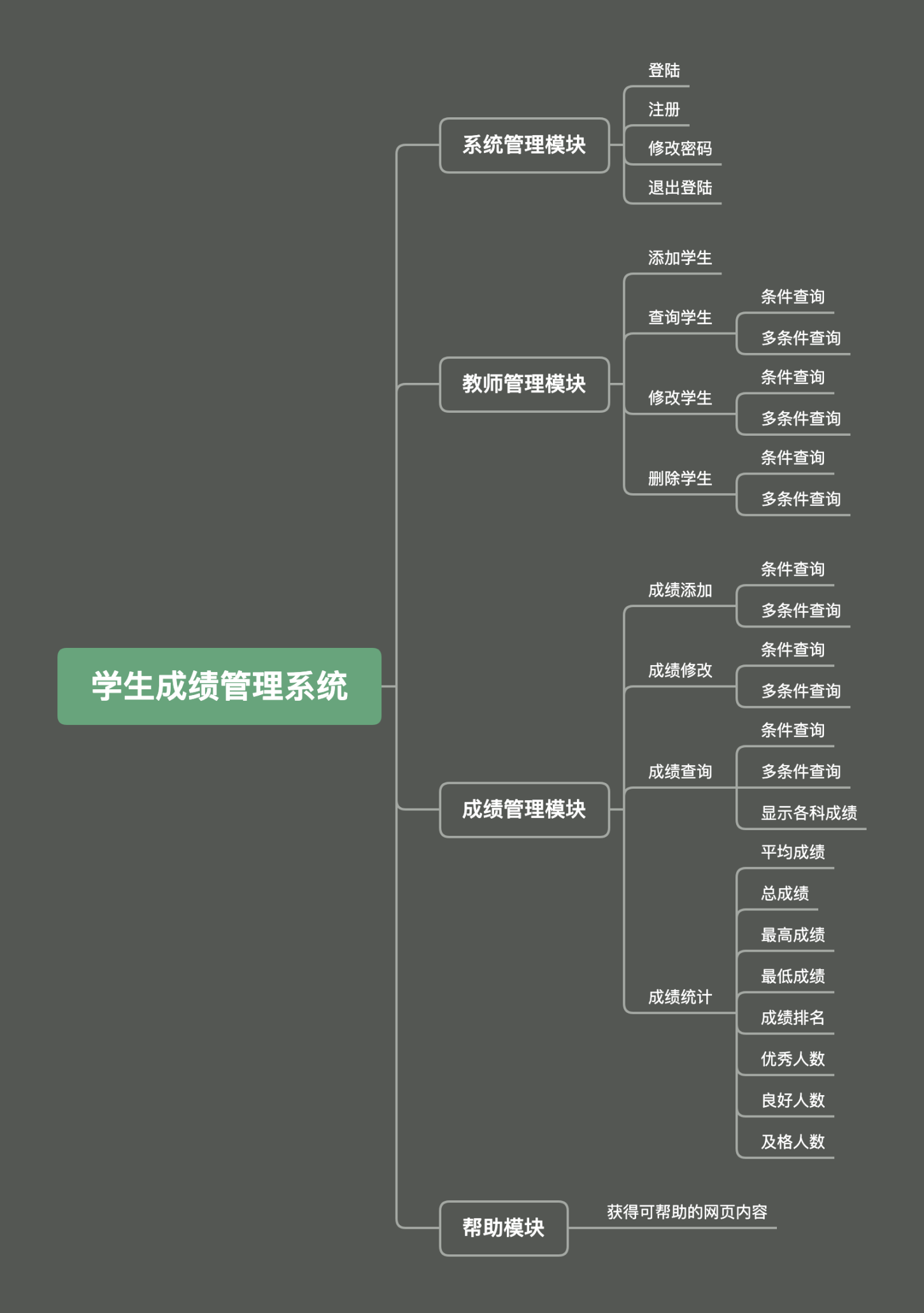
4、导入与数据库版本对应的桥接jar包，我的项目是用的MYSQL 8.0.20所以jar包为mysql-connector-java-8.0.20.jar

配置好以上就能运行程序了。

（windows系统编码可能会出错可参照我的博客进行修改）

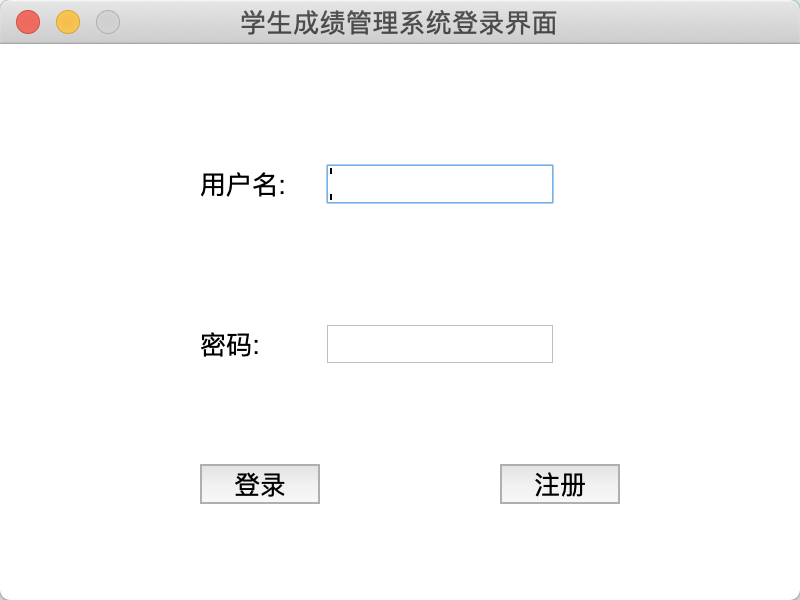
URL更改jdbc:mysql://localhost:3306/数据库名?**characterEncoding=utf-8**

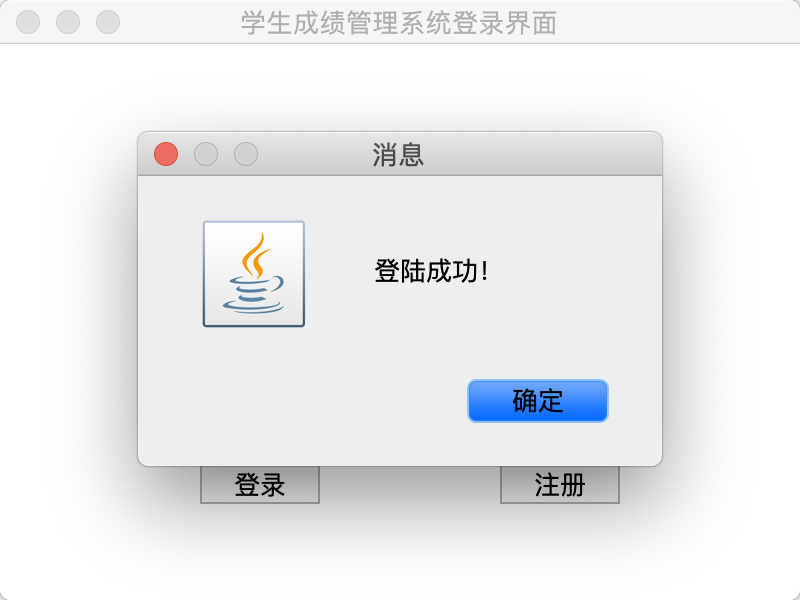
<https://blog.csdn.net/weixin_44193041/article/details/106575698>

****

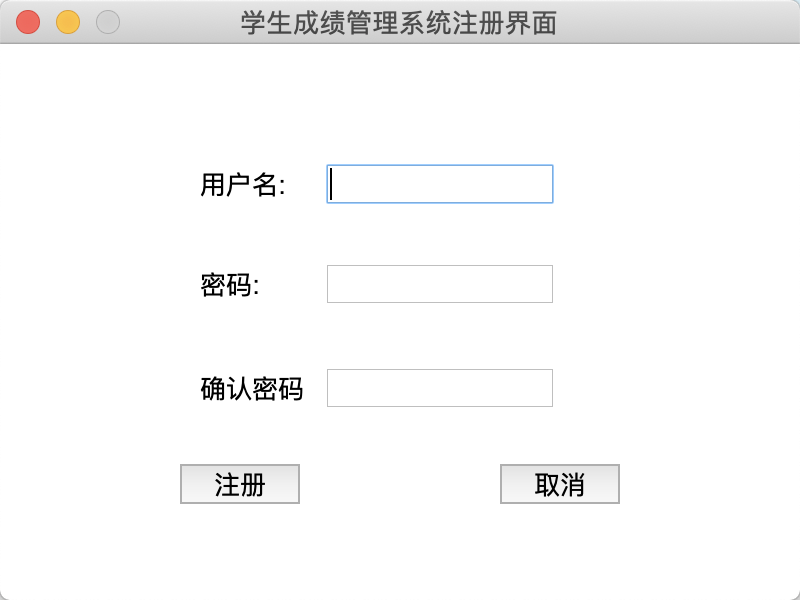
# 系统管理模块

# 3.1登陆





# 3.2注册



# 3.3修改密码



# 3.4退出登陆



# 教师管理模块

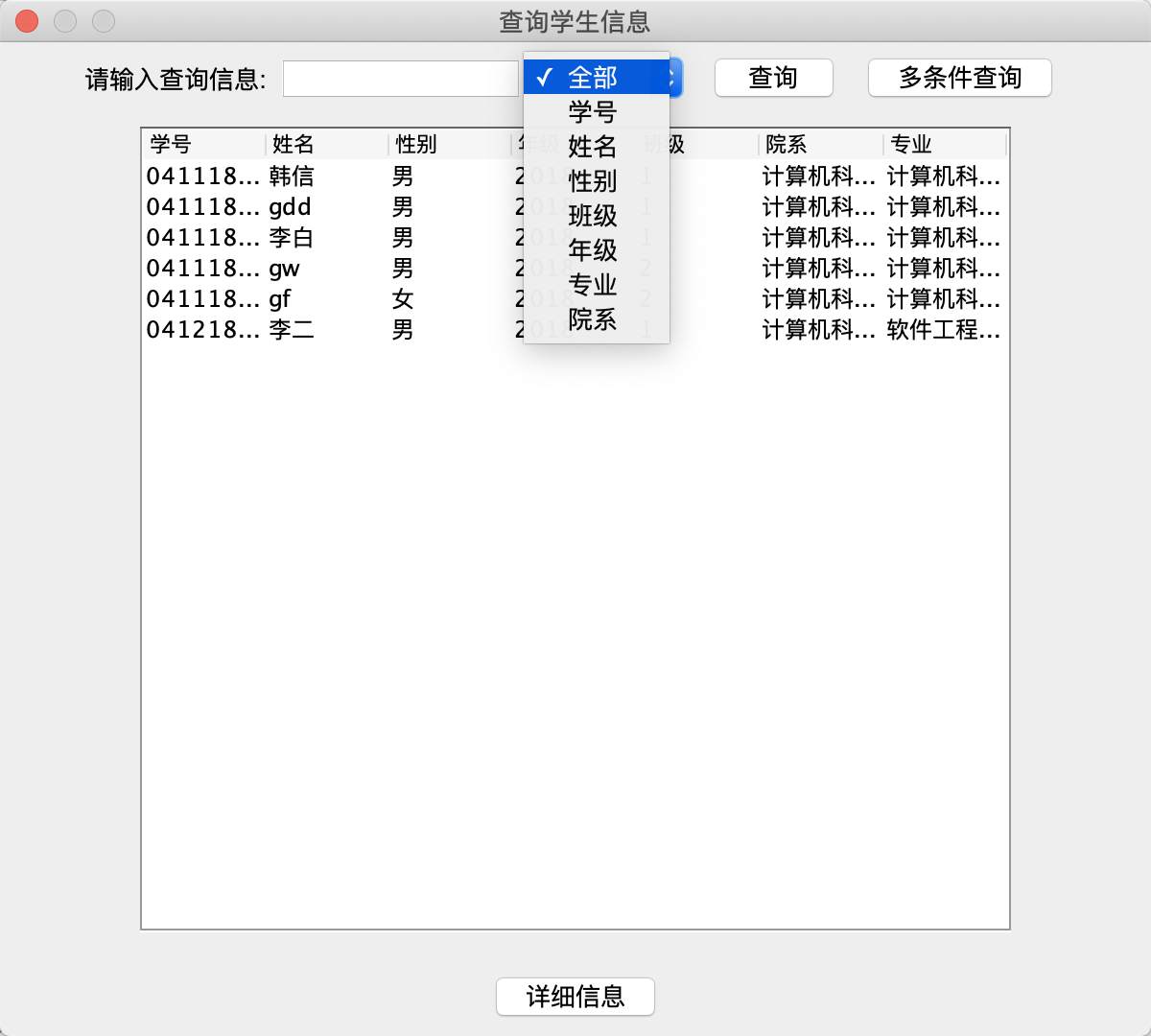


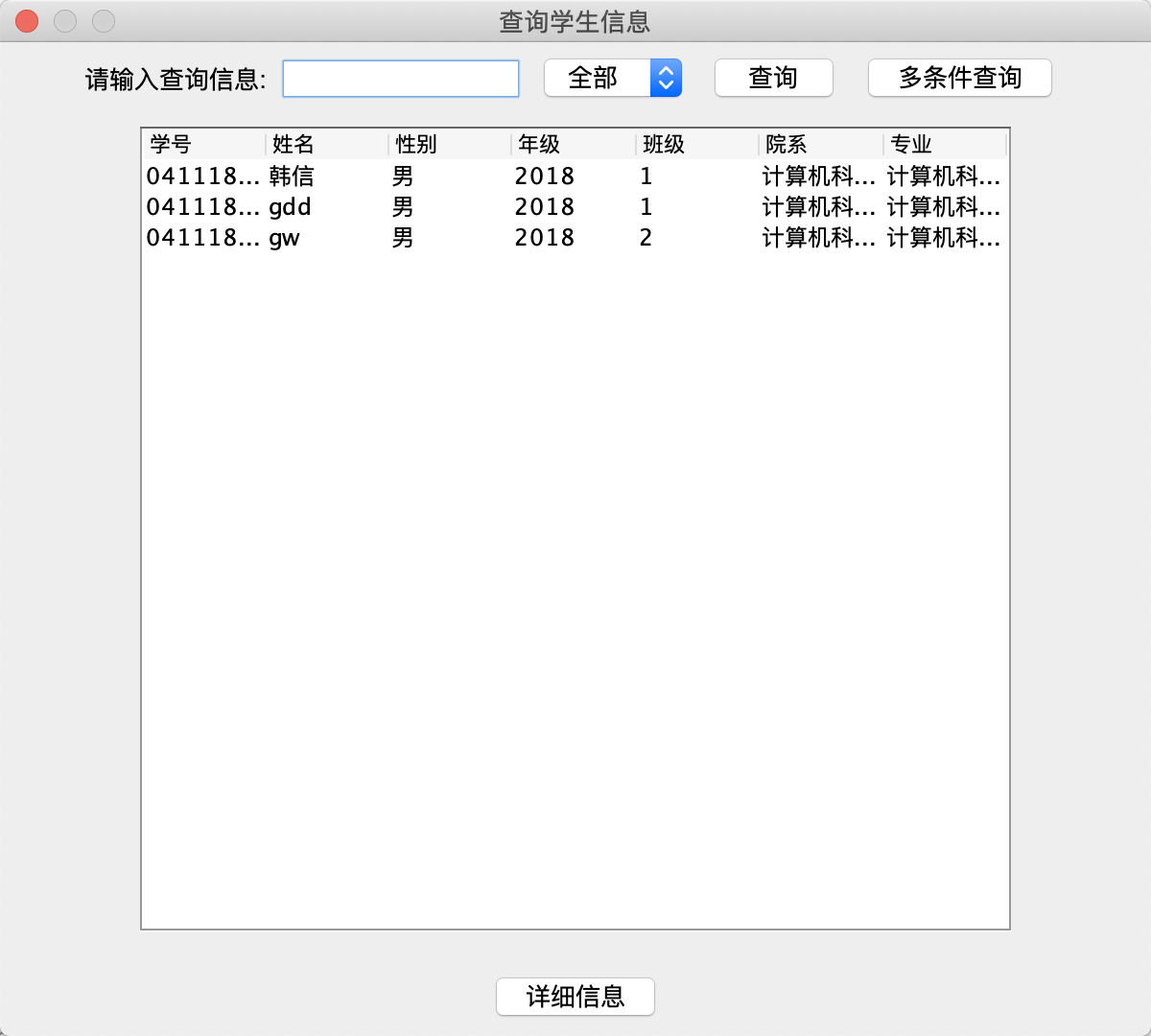
# 4.1添加学生



## **4.2查询学生**

## **4.2.1条件查询**



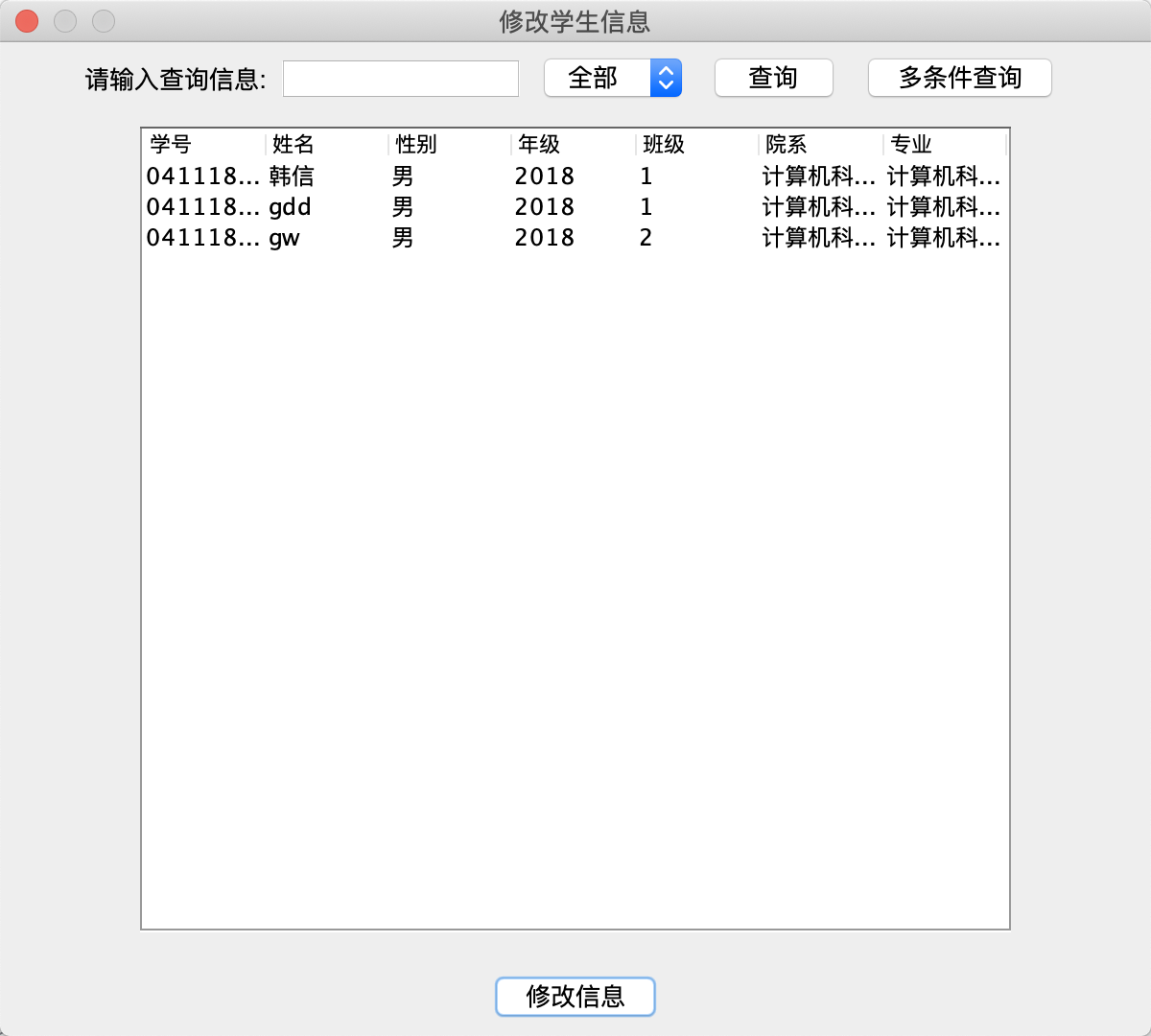


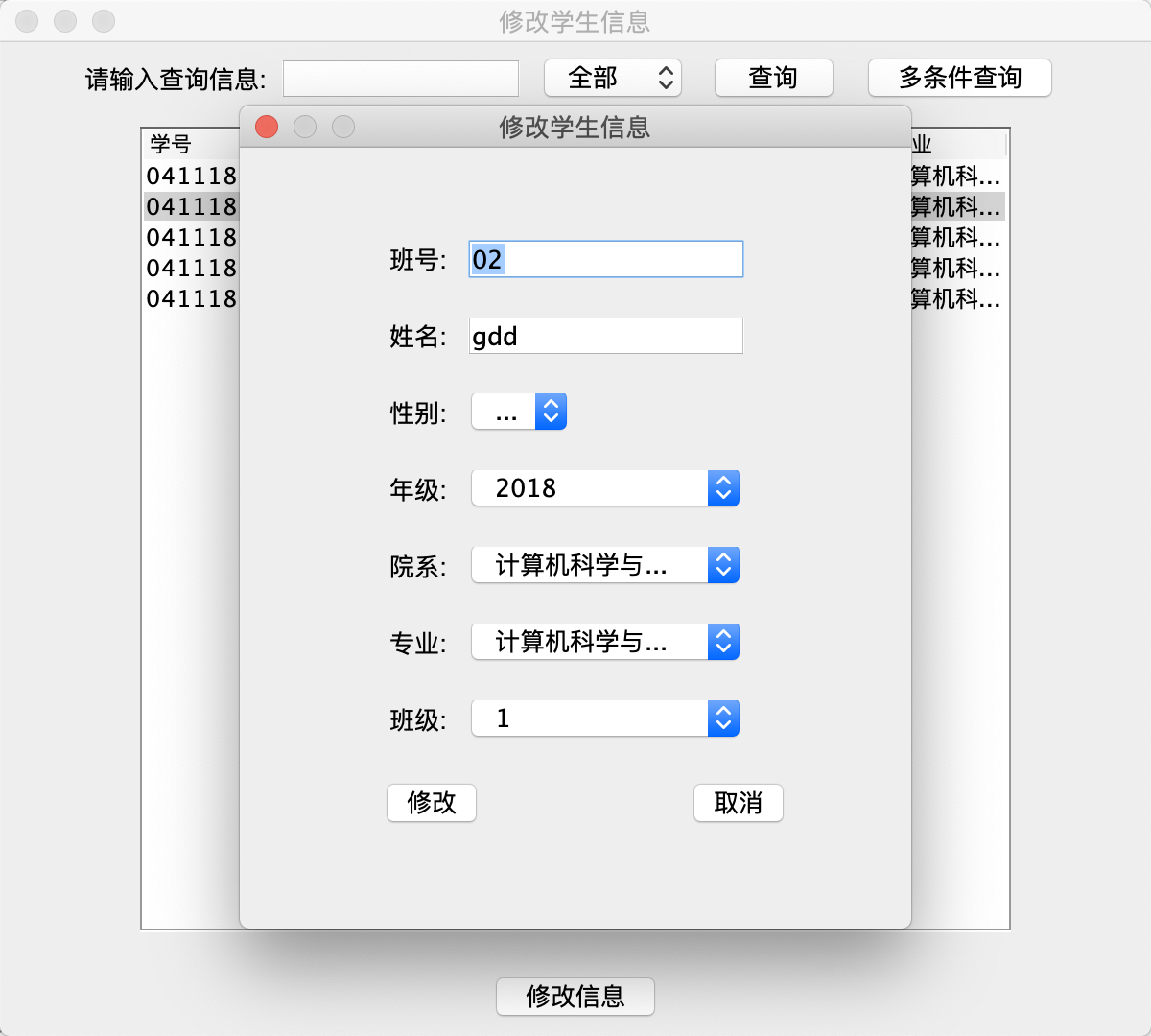


### 4.2.2多条件查询



## **4.3修改学生**

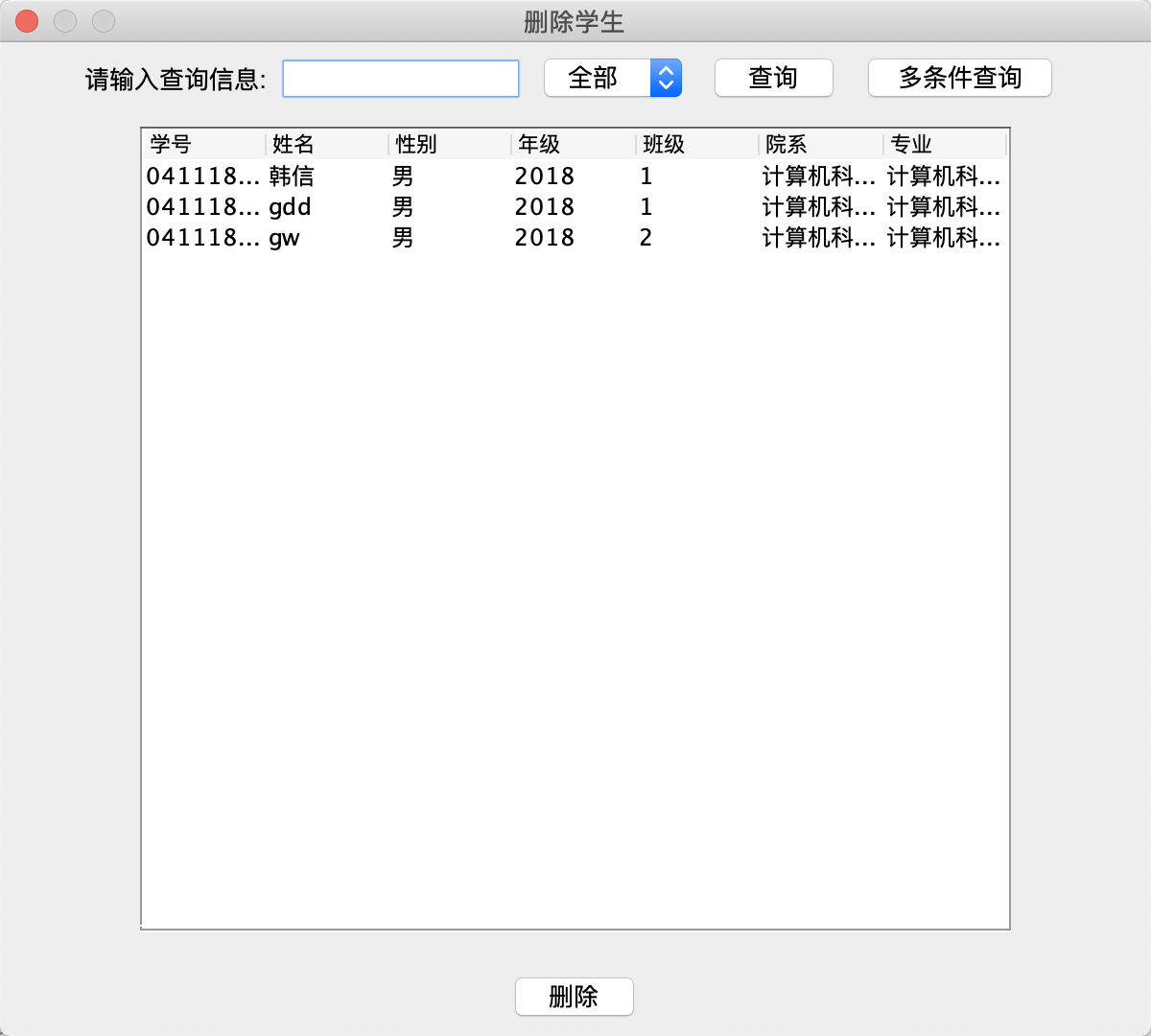




### 4.3.1条件查询

### 4.3.2多条件查询

## **4.4删除学生**





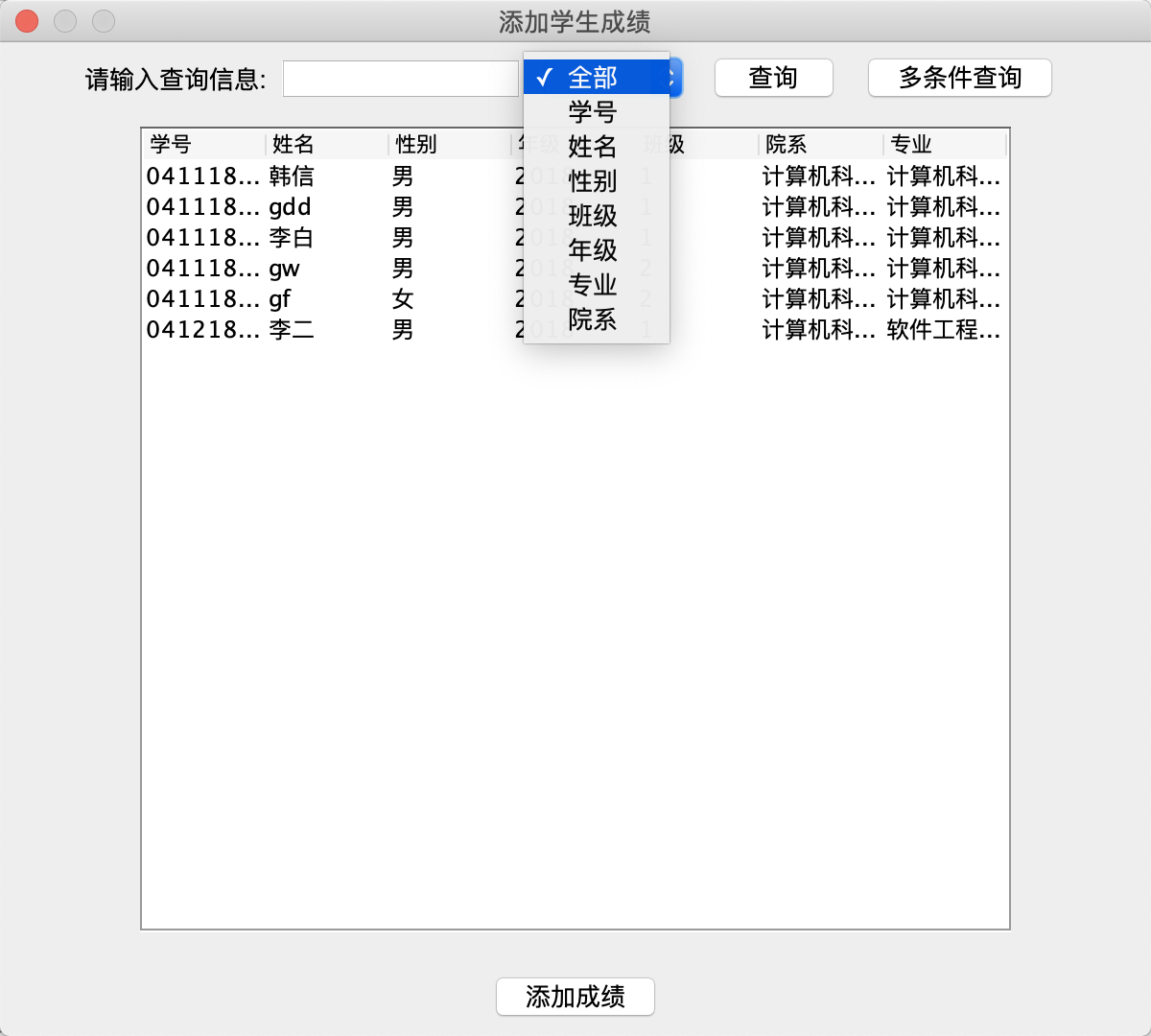
### 4.4.1条件查询

### 4.4.2多条件查询

# 成绩管理模块



## **5.1成绩添加**







### 5.1.1条件查询

### 5.1.2多条件查询

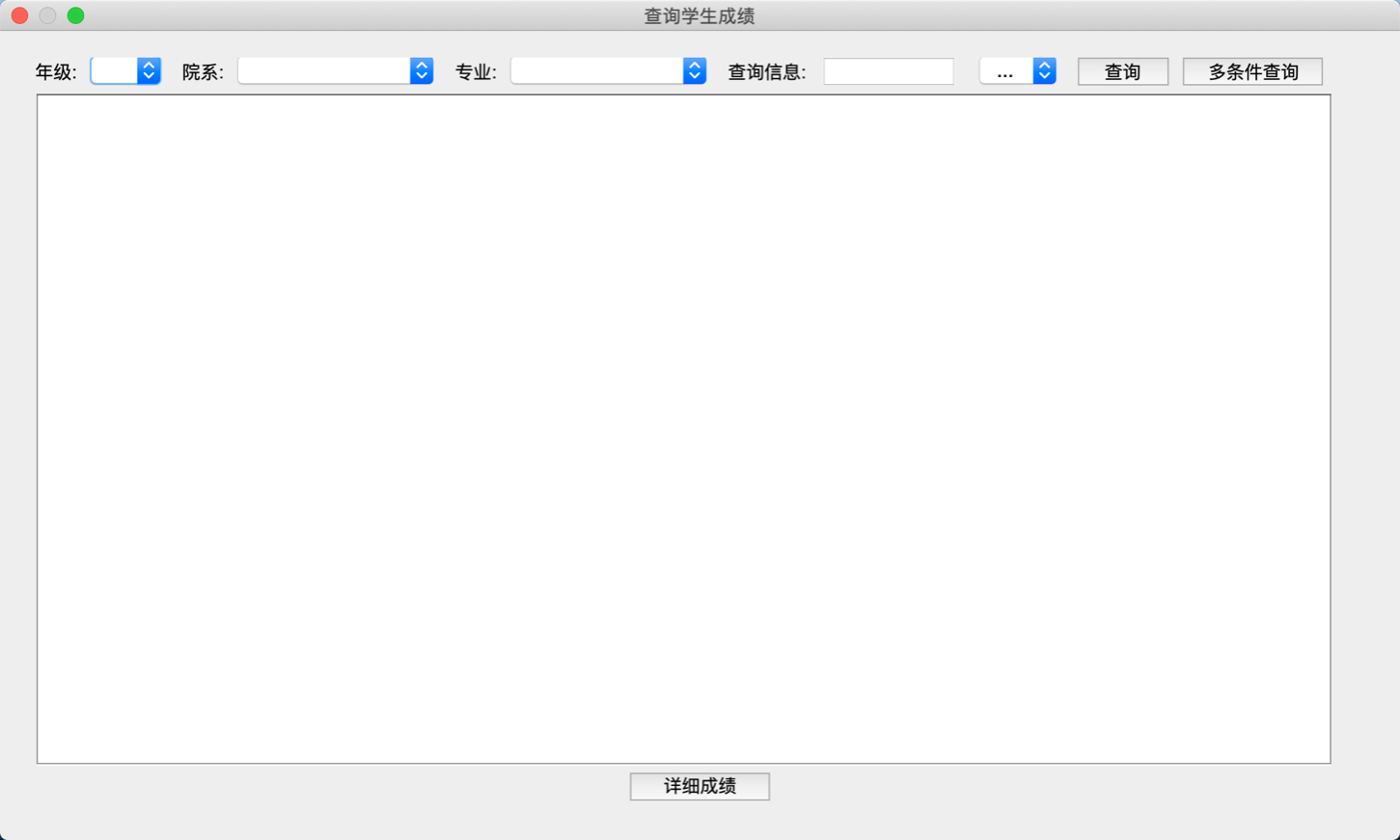
## **5.2成绩修改**



### 5.2.1条件查询

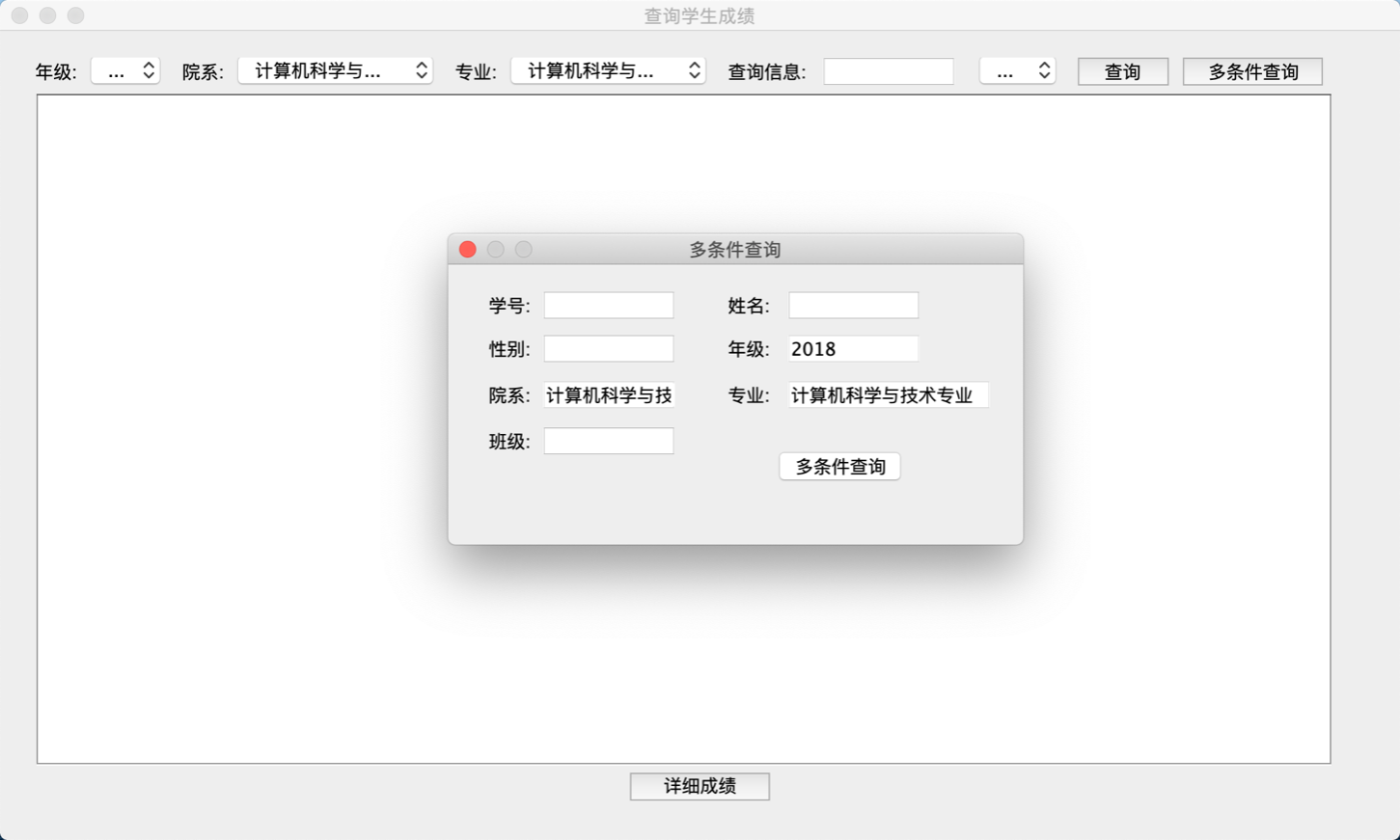
### 5.2.2多条件查询

## **5.3成绩查询**

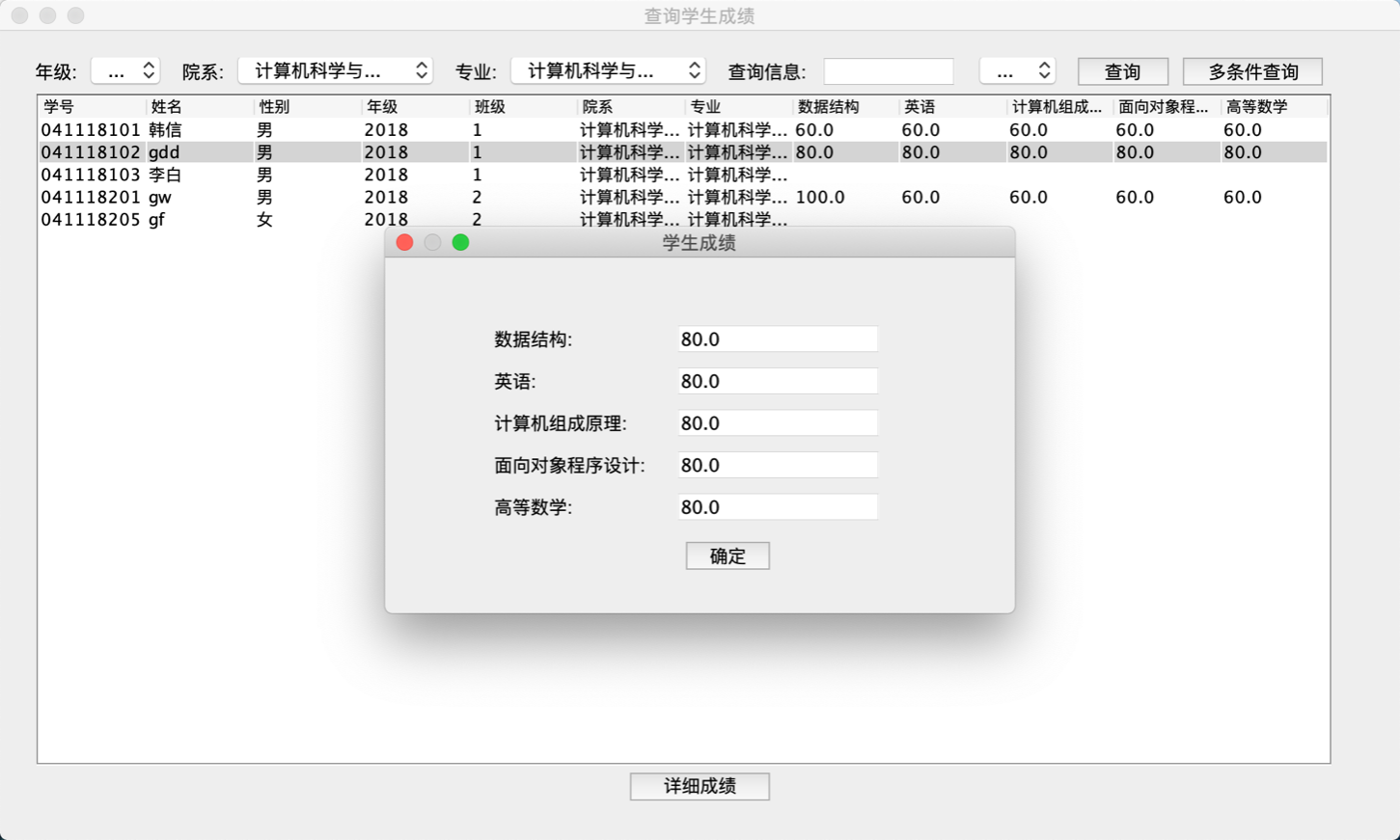


### 5.3.1条件查询

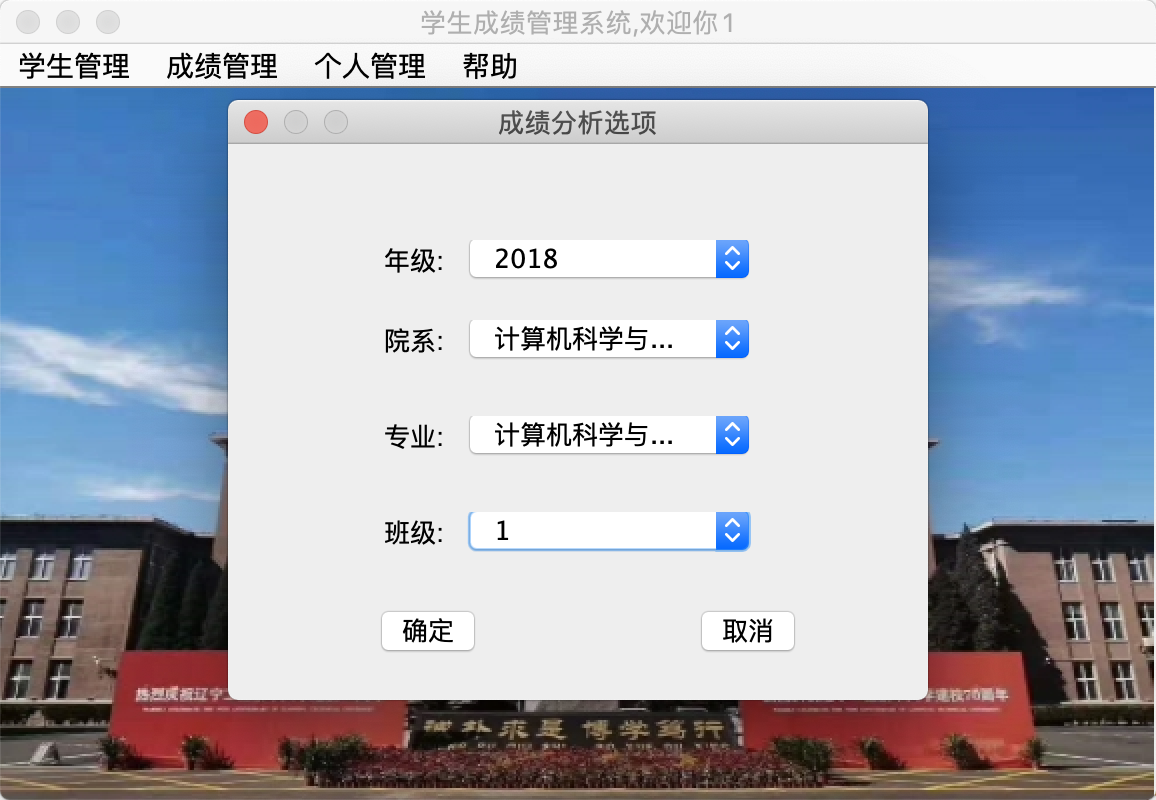
### 5.3.2多条件查询



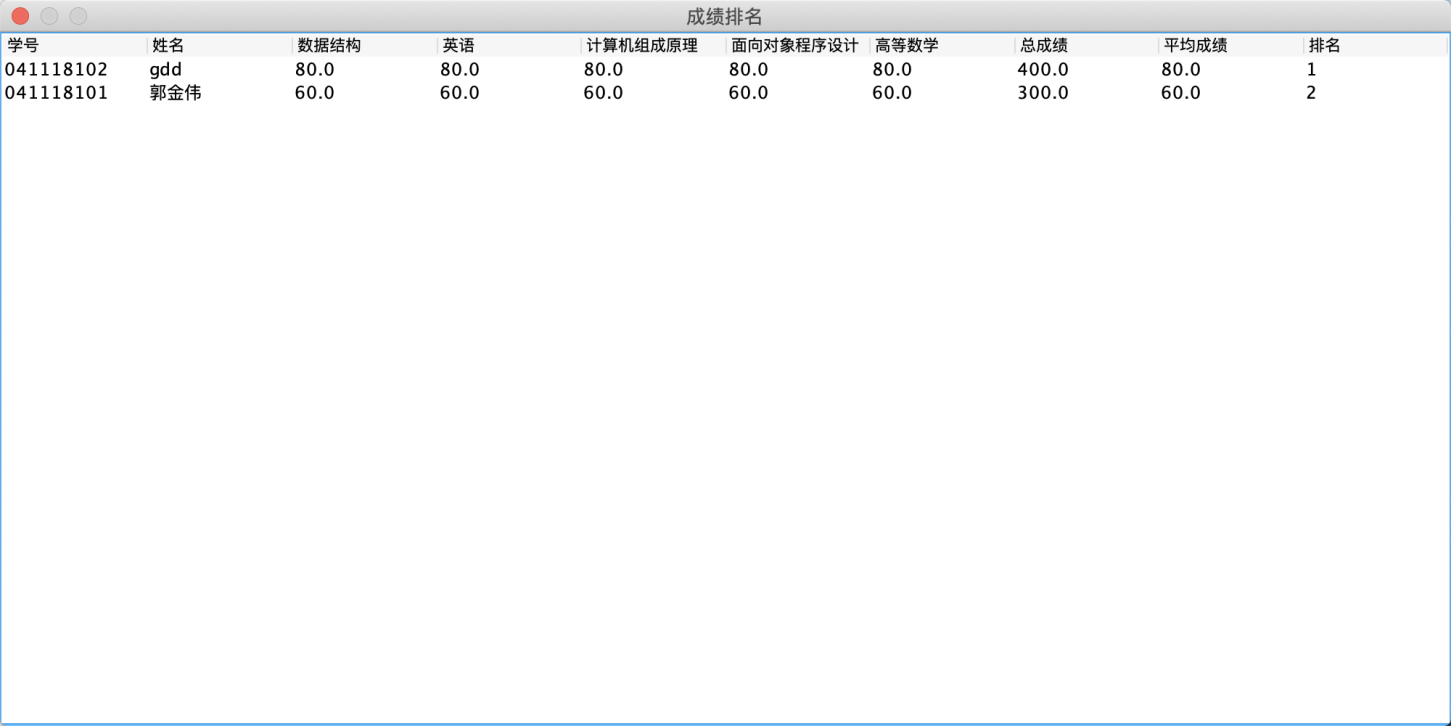
### 5.3.3显示各科成绩



## **5.4成绩统计**







### 5.4.1平均成绩

### 5.4.2总成绩

### 5.4.3最高成绩

### 5.4.4最低成绩

### 5.4.5成绩排名

### 5.4.6优秀人数

### 5.4.7良好人数

### 5.4.8及格人数

# 帮助模块

## **6.1获得可帮助的网页内容**

输入帮助文档网址即可进入本人的个人博客，获得帮助文档



1. **创建数据库表格的SQL语句**

|  |
| --- |
| **Mysql语句** |

create database StudentSystemDao;

use StudentSystemDao;

create table tb\_Department( #所属院系表

Department\_ID varchar(30) primary key, #院系编号

Department\_Name nvarchar(50) unique not null #院系名

);

use StudentSystemDao;

create table tb\_Major( #专业表

Major\_ID varchar(30) primary key, #专业编号

Major\_Name nvarchar(50) unique not null, #专业名称

Department\_ID varchar(30) not null, #所属院系ID

foreign key(Department\_ID) references tb\_Department(Department\_ID) #所属院系设置为院系表的外键

);

use StudentSystemDao;

create table tb\_Classe( #班级表

Classe int not null, #班级

Grade int not null, #年级

Major\_ID varchar(30) not null, #专业编号

primary key(Classe,Grade,Major\_ID) #班级，年级和专业编号共为主键

);

use StudentSystemDao;

create table tb\_Student( #学生表

Student\_Id varchar(30) primary key, #学生学号

Student\_Name nvarchar(50) not null, #学生姓名

Student\_Sex nchar(1) , #学生性别

Grade int not null, #年级

Classe int not null, #班级

Major\_ID varchar(30) not null, #专业编号

Major\_Name nvarchar(50) not null, #专业名称

Department\_ID varchar(30) not null, #所属院系ID

Department\_Name nvarchar(50) not null, #院系名称

foreign key(Department\_ID) references tb\_Department(Department\_ID), #所属院系编号设置为外键

foreign key(Major\_ID) references tb\_Major(Major\_ID), #专业编号设置为外键

foreign key(Department\_Name) references tb\_Department(Department\_Name), #所属院系名称设置为外键

foreign key(Major\_Name) references tb\_Major(Major\_Name) #专业名称设置为外键

);

use StudentSystemDao;

create table tb\_User( #用户表

User\_name nvarchar(30) primary key, #用户名

Password\_ varchar(30) not null, #密码

IsLogin bit not null DEFAULT 0 #是否重复登陆

);

use StudentSystemDao;

create table tb\_Course( #课程表

Course\_Name nvarchar(50) not null, #课程名称

Major\_ID varchar(30) not null, #所属专业编号

foreign key(Major\_ID) references tb\_Major(Major\_ID),#专业编号设置为专业表的外键

Grade int not null, #年级

primary key(Course\_Name,Major\_ID,Grade) #课程名称，专业，年级共为主键

);

use StudentSystemDao;

create table tb\_Score( #成绩表

Student\_Id varchar(30) not null, #学生ID

Student\_Name nvarchar(50) not null, #学生姓名

Course\_Name nvarchar(50) not null, #课程名称

primary key(Course\_Name,Student\_Id), #学生ID和课程名称共同组成主键

foreign key(Student\_Id) references tb\_Student(Student\_Id), #学生ID设置为学生表的外键

foreign key(Course\_Name) references tb\_Course(Course\_Name),

Score dec(4,1) default null #成绩在0到100之间

);

/\*插入院系\*/

insert into tb\_Department(Department\_ID ,Department\_Name) values('01','艺术系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('02','中国语言文学系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('03','电子信息系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('04','计算机科学与技术系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('05','物流与信息管理系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('06','旅游管理系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('07','机械与汽车工程系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('08','国际贸易与金融系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('09','工商管理系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('10','公共管理系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('11','建筑学系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('12','化学与药学系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('13','外语系');

insert into tb\_Department(Department\_ID ,Department\_Name) values('14','音乐舞蹈学院');

/\*插入专业\*/

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('01','视觉传达设计专业','01');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('02','动画专业','01');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('03','汉语言文学专业','02');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('04','广告学专业','02');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('05','汉语国际教育专业','02');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('06','电子信息科学与技术专业','03');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('07','微电子科学与工程专业','03');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('08','自动化专业','03');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('09','通信工程专业','03');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('10','测控技术与仪器专业','03');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('11','计算机科学与技术专业','04');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('12','软件工程专业','04');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('13','网络工程专业','04');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('14','物流管理专业','05');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('15','信息管理与信息系统专业','05');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('16','电子商务专业','05');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('17','旅游管理专业','06');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('18','机械设计制造及其自动化专业','07');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('19','汽车服务工程专业','07');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('20','车辆工程专业','07');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('21','工业工程专业','07');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('22','国际经济与贸易专业','08');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('23','金融学专业','08');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('24','工商管理专业','09');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('25','市场营销专业','09');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('26','人力资源管理专业','09');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('27','会计学专业','09');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('28','行政管理专业','10');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('29','劳动与社会保障专业','10');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('30','社会工作专业','10');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('31','建筑学专业','11');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('32','城乡规划专业','11');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('33','应用化学专业','12');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('34','制药工程(生物制药)专业','12');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('35','药物制剂专业','12');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('36','中药学专业','12');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('37','化学工程与工艺专业','12');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('38','英语专业','13');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('39','日语专业','13');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('40','朝鲜语（韩国语）专业','13');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('41','音乐表演专业','14');

insert into tb\_Major(Major\_ID,Major\_Name,Department\_ID) values('42','舞蹈编导专业','14');

/\*添加班级\*/

insert into tb\_Classe(Classe,Grade,Major\_ID) values(1,2018,'11');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(2,2018,'11');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(3,2018,'11');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(4,2018,'11');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(5,2018,'11');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(6,2018,'11');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(7,2018,'11');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(8,2018,'11');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(9,2018,'11');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(1,2018,'12');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(2,2018,'12');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(3,2018,'12');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(4,2018,'12');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(5,2018,'12');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(6,2018,'12');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(7,2018,'12');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(8,2018,'12');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(9,2018,'12');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(1,2018,'13');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(2,2018,'13');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(3,2018,'13');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(4,2018,'13');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(5,2018,'13');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(6,2018,'13');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(7,2018,'13');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(8,2018,'13');

insert into tb\_Classe(Classe,Grade,Major\_ID) values(9,2018,'13');

/\*添加课程\*/

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('高等数学','11',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('英语','11',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('数据结构','11',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('计算机组成原理','11',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('面向对象程序设计','11',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('高等数学','12',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('英语','12',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('数据结构','12',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('软件工程导论','12',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('面向对象程序设计','12',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('高等数学','13',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('英语','13',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('数据结构','13',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('计算机网络','13',2018);

insert into tb\_Course(Course\_Name,Major\_ID,Grade) values ('面向对象程序设计','13',2018);

1. **程序具体代码**

**8.1连接数据库的代码**

|  |
| --- |
| **JdbcConfig.java** |

**package** studentsystem.dao;

**public** **interface** JdbcConfig {

// 新版mysql要加cj连接方式

String ***DRIVER*** = "com.mysql.cj.jdbc.Driver";

// url格式

String ***URL*** = "jdbc:mysql://localhost:3306/StudentSystemDao";

// 用户名称

String ***USERNAME*** = "root";

// 用户密码

String ***PASSWORD*** = "12345678";

}

|  |
| --- |
| **JdbcHelper.java** |

**package** studentsystem.dao;

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

//与数据库通信的类

**import** java.util.HashMap;

**import** java.util.LinkedHashMap;

**import** java.util.Vector;

**import** studentsystem.bean.AnalyzeResult;

**import** studentsystem.bean.Student;

**import** studentsystem.bean.User;

**public** **class** JdbcHelper **implements** JdbcConfig{

//定义连接数据库所需要的对象

//预编译对象ps

**private** PreparedStatement ps = **null**;

//结果集对象

**private** ResultSet rs = **null**;

//连接对象

**private** Connection ct = **null**;

//获得数据库的连接

**private** **void** init(){

**try** {

//加载及注册JDBC驱动程序

Class.*forName*(DRIVER);

ct = DriverManager.getConnection(URL, USERNAME, PASSWORD);// 获得数据库连接

} **catch** (ClassNotFoundException e) {

// **TODO** Auto-generated catch block

//没找到DRIVER

e.printStackTrace();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

//没连接数据库报错

e.printStackTrace();

}

}

//无参构造函数

//初始化连接

**public** JdbcHelper(){

**this**.init();

}

/\*\*

\* 获取用户对象

\* 根据传入的用户名，获取对应的用户，并返回用户对象

\* **@return** 用户对象

\*/

**public** User getUser(User user){

//实例化一个用户对象

User newUser = **new** User();

**try** {

//获得预编译用户

ps = ct.prepareStatement("select \* from tb\_User where User\_name=?");

// select \* from tb\_User where User\_name="?"

// 以上第一个问号的参数设置为，user.getUsername();

ps.setString(1, user.getUsername());

// 产生上述查询的单个结果集

rs = ps.executeQuery();

**if**(rs.next()){

newUser.setUsername(rs.getString(1)); //设置用户名

newUser.setPassword(rs.getString(2)); //设置密码

newUser.setIsLogin(rs.getInt(3)); //设置是否登陆

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

// 获取完用户后返回用户的内容

**return** newUser;

}

/\*\*

\* 注册处理

\* **@param** user 用户对象

\* **@return** 返回是否注册成功

\*/

**public** **boolean** register(User user){

**boolean** b = **true**;

**try** {

// 预编译注册对象的用户名，和注册对象的密码

ps = ct.prepareStatement("insert into tb\_User(User\_name,Password\_) values(?,?)");

// 第一个问号的参数设置为用户名

ps.setString(1, user.getUsername());

// 第二个问好的参数设置为密码

ps.setString(2, user.getPassword());

// 如果ps处理后改变行数不为1，表示注册用户失败

**if**(ps.executeUpdate()!=1){ //执行sql语句

b = **false**;

}

} **catch** (SQLException e) {

b = **false**;

e.printStackTrace();

}

// 返回注册成功与否的状态

**return** b;

}

/\*\*

\* 修改用户"是否"登陆状态

\* **@param** user

\* **@return**

\*/

**public** **boolean** update\_IsLogin(User user){

**boolean** b = **true**;

**try** {

// 预编译用户名的登陆状态

ps = ct.prepareStatement("update tb\_User set IsLogin=? where User\_name=?");

// 第一个问号参数为登陆状态

ps.setInt(1, user.getIsLogin());

// 第二个问号参数为用户名

ps.setString(2, user.getUsername());

// 判断是否登陆

**if**(ps.executeUpdate()!=1){

b = **false**;

}

} **catch** (SQLException e) {

b = **false**;

e.printStackTrace();

}

**return** b;

}

/\*\*

\* 修改密码

\* **@param** user 用户对象

\* **@param** new\_Password 新密码

\* **@return** 返回是否修改成功

\*/

**public** **boolean** update\_Password(User user,String new\_Password){

**boolean** b = **true**;

**try** {

// 预编译用户名的密码

ps = ct.prepareStatement("update tb\_User set Password\_=? where User\_name=?");

// 第一个问号参数为用户新密码

ps.setString(1, new\_Password);

// 第二个参数为用户名

ps.setString(2, user.getUsername());

// 判断是否修改成功

**if**(ps.executeUpdate()!=1){ //执行sql语句

b = **false**;

}

} **catch** (SQLException e) {

b = **false**;

e.printStackTrace();

}

**return** b;

}

/\*\*

\* 获得所有院系

\* **@return** 返回所有院系的HashMap集合

\*/

**public** HashMap<String, String> getAllDepartment(){

// 创建一个Map集合

HashMap<String, String> map = **new** LinkedHashMap<String, String>();

map.put("", "");//添加一个空的元素

**try** {

// 预编译按照tb\_Department的Department\_ID的升序，获得结果集

ps = ct.prepareStatement("select \* from tb\_Department order by Department\_ID");

// rs返回查询的结果集

rs = ps.executeQuery();

// 将结果集保存到hash map中

**while**(rs.next()){

map.put(rs.getString(2),rs.getString(1));

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** map;

}

/\*\*

\* 获得对应院系的专业

\* **@return** 返回Vector<String>对象

\*/

**public** Vector<String> getMajor(String department\_ID){

Vector<String> vector = **new** Vector<String>();

vector.add("");//添加一个空的元素

**try** {

// 预编译按照专业表的对应的Department\_ID的Major\_ID的升序，获得结果集

ps = ct.prepareStatement("select \* from tb\_Major where Department\_ID=? order by Major\_ID");

// 第一个问号参数设置为department\_ID

ps.setString(1, department\_ID);

// 返回结果集

rs = ps.executeQuery();

**while**(rs.next()){

// 获得返回结果的第二个位置放到vector中

vector.add(rs.getString(2)); //获得专业名称

}

} **catch** (SQLException e) {

e.printStackTrace();

}

// 返回对应院系的专业

**return** vector;

}

/\*\*

\* 获得所有专业

\* **@return** 返回所有专业

\*/

**public** HashMap<String, String>getAllMajor(){

HashMap<String, String> map = **new** LinkedHashMap<String, String>();

map.put("", "");//添加一个空的元素

**try** {

// 预编译按照专业表对应的Major\_ID的升序获得结果集

ps = ct.prepareStatement("select \* from tb\_Major order by Major\_ID");

// 返回结果集

rs = ps.executeQuery();

**while**(rs.next()){

// 将结果集的第二个位置和第一个位置放到map中

map.put(rs.getString(2),rs.getString(1));

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

// 返回所有专业

**return** map;

}

/\*\*

\* 添加学生

\* **@param** student 学生对象

\* **@return** 返回是否添加成功

\*/

**public** **boolean** addStudent(Student student){

**boolean** b = **true**;

**try** {

//预编译插入学生

ps = ct.prepareStatement("insert into tb\_Student(Student\_Id,Student\_Name,Student\_Sex,Classe,Grade,Major\_ID,Department\_ID,Major\_Name,Department\_Name) values(?,?,?,?,?,?,?,?,?)");

// 将学生的信息插入表中，通过获得的学生信息

ps.setString(1, student.getStudent\_ID());

ps.setString(2, student.getStudent\_Name());

ps.setString(3, student.getSex());

ps.setString(4, student.getClasse());

ps.setString(5, student.getGrade());

ps.setString(6, student.getMajor\_ID());

ps.setString(7, student.getDepartment\_ID());

ps.setString(8, student.getMajor\_Name());

ps.setString(9, student.getDepartment\_Name());

**if**(ps.executeUpdate()!=1){

b = **false**;

}

} **catch** (SQLException e) {

b = **false**;

e.printStackTrace();

}

**return** b;

}

/\*\*

\* 修改学生信息

\* **@param** newStudent 新学生对象

\* **@param** oldStudentID 旧学生信息

\* **@return** 是否修改成功

\*/

**public** **boolean** updateStudent(Student newStudent,String oldStudentID){

**boolean** b = **true**;

**try** {

//update

// 预编译更新学生的信息

ps = ct.prepareStatement("update tb\_Student set Student\_Id=?, Student\_Name=?, Student\_Sex=? ,Classe=? ,Grade=? ,Major\_ID=? ,Department\_ID=? ,Major\_Name=? ,Department\_Name=? where Student\_Id=?");

ps.setString(1, newStudent.getStudent\_ID());

ps.setString(2, newStudent.getStudent\_Name());

ps.setString(3, newStudent.getSex());

ps.setString(4, newStudent.getClasse());

ps.setString(5, newStudent.getGrade());

ps.setString(6, newStudent.getMajor\_ID());

ps.setString(7, newStudent.getDepartment\_ID());

ps.setString(8, newStudent.getMajor\_Name());

ps.setString(9, newStudent.getDepartment\_Name());

// 对旧学生的id进行更新

ps.setString(10, oldStudentID);

// 判断是否执行成功

**if**(ps.executeUpdate()!=1){

b = **false**;

}

} **catch** (SQLException e) {

b = **false**;

e.printStackTrace();

}

**return** b;

}

/\*\*

\* 根据学生学号从数据库移除该学生

\* **@param** studentID 学生学号

\* **@return** 返回是否删除成功

\*/

**public** **boolean** deleteStudent(String studentID){

**boolean** b = **true**;

**try** {

// 预编译删除学生，根据Student\_Id

ps = ct.prepareStatement("delete from tb\_Student where Student\_Id=?");

ps.setString(1, studentID);

**if**(ps.executeUpdate()!=1){

b = **false**;

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

b = **false**;

e.printStackTrace();

}

**return** b;

}

/\*\*

\* 根据sql语句返回特定的学生集合

\* **@param** sql sql语句

\* **@return** 返回Vector<Student>对象

\*/

**public** Vector<Student> getStudent(String sql){

Vector<Student> students = **new** Vector<Student>();

**try** {

// 返回特定学生的集合根据sql语句

// 预编译

ps = ct.prepareStatement(sql);

// 编译，获得结果集

rs = ps.executeQuery();

**while**(rs.next()){

// 获得学生对象

Student student = **new** Student();

student.setStudent\_ID(rs.getString(1));

student.setStudent\_Name(rs.getString(2));

student.setSex(rs.getString(3));

student.setGrade(rs.getString(4));

student.setClasse(rs.getString(5));

student.setMajor\_ID(rs.getString(6));

student.setMajor\_Name(rs.getString(7));

student.setDepartment\_ID(rs.getString(8));

student.setDepartment\_Name(rs.getString(9));

// 将学生对象添加到结果集中

students.add(student);

}

}**catch**(SQLException e){

e.printStackTrace();

}

**return** students;

}

/\*\*

\* 根据学生(年级和编号)返回对应的班级

\* **@param** student

\* **@return**

\*/

**public** Vector<String> getAllClasse(String grade,String major\_ID){

Vector<String> vector = **new** Vector<String>();

vector.add(""); //添加一个空选项

**try** {

// 根据年级和编号，返回对应的班级

ps = ct.prepareStatement("select Classe from tb\_Classe where Grade=? and Major\_ID=?");

// 预编译，第一个问号设置为年级grade

ps.setString(1, grade);

// 预编译，第二个问号设置为专业ID

ps.setString(2, major\_ID);

rs = ps.executeQuery();

**while**(rs.next()){

// 返回其班级

vector.add(rs.getString(1));

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** vector;

}

/\*\*

\* 根据专业编号返回对应的所有课程

\* **@param** major\_Id

\* **@return** 返回课程集合

\*/

**public** Vector<String> getCourse(String major\_Id,String grade){

Vector<String> vector = **new** Vector<String>();

**try** {

// 预编译，根据专业编号和年级，获得结果集

ps = ct.prepareStatement("select Course\_Name from tb\_Course where Major\_ID=? and Grade=?");

// 预编译，第一个问号为专业编号

ps.setString(1, major\_Id);

// 预编译，第二个问号为年级

ps.setString(2, grade);

// 返回结果集

rs = ps.executeQuery();

**while**(rs.next()){

// 将课程添加到vector容器中

vector.add(rs.getString(1));

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** vector;

}

/\*\*

\* 添加学生成绩

\* **@param** student 学生对象

\* **@return** 返回是否添加成功

\*/

**public** **boolean** addStudentScore(Student student,Vector<String> courses){

**boolean** b = **true**;

HashMap<String, String> map = student.getScores();

**try** {

**int** i;

**for**(i=0;i<map.size();i++){

ps = ct.prepareStatement("insert into tb\_Score(Course\_Name,Student\_Id,Student\_Name,Score) values (?,?,?,?)");

// 预编译，第一个问号为课程的名字

ps.setString(1, courses.get(i));

// 预编译，第二个问号为学生的ID

ps.setString(2, student.getStudent\_ID());

// 第三个问号为学生的姓名

ps.setString(3, student.getStudent\_Name());

// 第四个问号为学生成绩

ps.setString(4, map.get(courses.get(i)));

// 如果编译失败，终止

**if**(ps.executeUpdate()!=1){

**break**;

}

}

// 如果i会小于容器的大小，则表示创建失败

**if**(i<map.size()){

b = **false**;

}

} **catch** (SQLException e) {

b = **false**;

e.printStackTrace();

}

**return** b;

}

/\*\*

\* 修改学生成绩

\* **@param** student\_Id 学生学号

\* **@param** course\_Name 课程名字

\* **@param** score 成绩

\* **@return** 返回是否修改成功

\*/

**public** **boolean** updateStudentScore(String student\_Id,String course\_Name,String score){

**boolean** b = **true**;

**try** {

// 根据课程的名字Course\_Name还有学生的id（Student\_Id）来更新学生的成绩

ps = ct.prepareStatement("update tb\_Score set Score=? where Course\_Name=? and Student\_Id=? ");

ps.setString(1, score);

ps.setString(2, course\_Name);

ps.setString(3, student\_Id);

**if**(ps.executeUpdate()!=1){

b = **false**;

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

b = **false**;

e.printStackTrace();

}

**return** b;

}

/\*\*

\* 查询学生所有成绩

\* **@param** student\_Id 学生学号

\* **@return** 返回成绩集合

\*/

**public** HashMap<String, String> getStudentScore(String student\_Id){

HashMap<String, String> scores = **new** HashMap<String, String>();

**try** {

// 按照ascii码升序排列根据学生的ID（Student\_Id）

ps = ct.prepareStatement("select Course\_Name,Score from tb\_Score where Student\_Id=? order by Student\_Id asc");

ps.setString(1, student\_Id);

// 返回结果集

rs = ps.executeQuery();

**while**(rs.next()){

scores.put(rs.getString(1), rs.getString(2));

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

// 返回成绩结果

**return** scores;

}

// /\*\*

// \* 根据学号和课程名查询指定的成绩

// \* @param student\_Id 学号

// \* @param course\_Name 课程名

// \* @return 返回成绩

// \*/

// public String getStudentScore(String student\_Id,String course\_Name){

// String score = null;

// try {

// ps = ct.prepareStatement("select Score from tb\_Score where Student\_Id=? and Course\_Name=?");

// ps.setString(1, student\_Id);

// ps.setString(2, course\_Name);

// rs = ps.executeQuery();

// rs.next();

// score = rs.getString(1);

// } catch (SQLException e) {

// // **TODO** Auto-generated catch block

// e.printStackTrace();

// }

// return score;

// }

//

//

/\*\*

\* 根据特定的sql语句返回学生集合

\* **@param** sql

\* **@return**

\*/

**public** Vector<Student> getScores(String sql){

Vector<Student> students = **new** Vector<Student>();

**try** {

ps = ct.prepareStatement(sql);

// 获得结果集

rs = ps.executeQuery();

**while**(rs.next()){

Student student = **new** Student();

HashMap<String, String>scores = **new** HashMap<String, String>();

// 成绩，课程名，还有得分

scores.put(rs.getString(3), rs.getString(4));

// 设置成绩

student.setScores(scores);

// 获得Student\_ID

student.setStudent\_ID(rs.getString(1));

// 获得Student\_Name

student.setStudent\_Name(rs.getString(2));

// 将学生添加到vector<Student>中

students.add(student);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

};

**return** students;

}

/\*\*

\* 获得班级总成绩

\* **@param** classe 班级

\* **@param** grade 年级

\* **@param** major 专业

\* **@return** 总成绩

\*/

**public** String getClasse\_SumScore(String classe,String grade,String major){

String sum = **null**;

**try** {

// 返回全班的成绩，根据年级和Major\_Name专业名字

ps = ct.prepareStatement("select SUM(Score) from tb\_Score where Student\_Id in(Select Student\_Id from tb\_Student where Classe=? and Grade=? and Major\_Name=?)");

ps.setString(1, classe);

ps.setString(2,grade);

ps.setString(3, major);

rs = ps.executeQuery();

**if**(rs.next()){

sum = rs.getString(1);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** sum;

}

/\*\*

\* 根据学生id返回该学生的平均成绩

\* **@param** student\_Id

\* **@return**

\*/

**public** String getStudentAvgScore(String student\_Id){

String avg = **null**;

**try** {

ps = ct.prepareStatement("select AVG(Score) from tb\_Score where Student\_Id=? ");

ps.setString(1, student\_Id);

rs = ps.executeQuery();

**if**(rs.next()){

avg = rs.getString(1);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** avg;

}

/\*\*

\* 根据学生专业，年级，班级，科目，考试成绩类型(优，良，中，及格，不及格)返回该类型的学生人数

\* **@param** grade

\* **@param** classe

\* **@param** major

\* **@param** type

\* **@param** course\_Name

\* **@return**

\*/

**public** String getCount\_ScoreType(String grade,String classe,String major,String type,String course\_Name){

String count = **null**;

String sql = **null**;

// 根据学生的年级，班级，专业，科目

**if**(type.equals("优")){

sql = "select Count(\*) from tb\_Score where Student\_Id in(select Student\_Id from tb\_Student where Grade=? and Classe=? and Major\_Name=?) and Course\_Name=? and Score>=90 and Score<=100";

}**else** **if**(type.equals("良")){

sql = "select Count(\*) from tb\_Score where Student\_Id in(select Student\_Id from tb\_Student where Grade=? and Classe=? and Major\_Name=?) and Course\_Name=? and Score>=80 and Score<90";

}**else** **if**(type.equals("中")){

sql = "select Count(\*) from tb\_Score where Student\_Id in(select Student\_Id from tb\_Student where Grade=? and Classe=? and Major\_Name=?) and Course\_Name=? and Score>=70 and Score<80";

}**else** **if**(type.equals("及格")){

sql = "select Count(\*) from tb\_Score where Student\_Id in(select Student\_Id from tb\_Student where Grade=? and Classe=? and Major\_Name=?) and Course\_Name=? and Score>=60 and Score<70";

}**else** **if**(type.equals("不及格")){

sql = "select Count(\*) from tb\_Score where Student\_Id in(select Student\_Id from tb\_Student where Grade=? and Classe=? and Major\_Name=?) and Course\_Name=? and Score>=0 and Score<60";

}

**try** {

ps = ct.prepareStatement(sql);

ps.setString(1, grade);

ps.setString(2, classe);

ps.setString(3, major);

ps.setString(4, course\_Name);

rs = ps.executeQuery();

**if**(rs.next()){

count = rs.getString(1);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** count+"人";

}

/\*\*

\* 获得课程的平均分

\* **@param** course 课程名

\* **@param** major 专业

\* **@return** 平均分

\*/

**public** String getCourseAvg(String course,String major,String classe,String grade){

String avg = **null**;

**try** {

// 根据课程名字，获得平均分（在年级，班级，和专业名选中的条件下）

ps = ct.prepareStatement("select AVG(Score) from tb\_Score where Course\_Name=? and Student\_Id in (select Student\_Id from tb\_Student where Grade=? and Classe=? and Major\_Name=?) ");

ps.setString(1, course);

ps.setString(2, grade);

ps.setString(3,classe);

ps.setString(4, major);

rs = ps.executeQuery();

**if**(rs.next()){

avg = rs.getString(1);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** avg;

}

/\*\*

\* 获得单科成绩的最高分

\* **@param** course

\* **@param** major

\* **@param** classe

\* **@param** grade

\* **@return**

\*/

**public** String getCourseHighestScore(String course,String major,String classe,String grade){

String highest = **null**;

**try** {

// 根据学生的ID，获得平均分（在年级，班级，和专业名选中的条件下）

ps = ct.prepareStatement("select MAX(Score) from tb\_Score where Student\_Id in(select Student\_Id from tb\_Student where Grade=? and Classe=? and Major\_Name=?) and Course\_Name=? " );

ps.setString(1, grade);

ps.setString(2, classe);

ps.setString(3, major);

ps.setString(4, course);

rs = ps.executeQuery();

**if**(rs.next()){

highest = rs.getString(1);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** highest;

}

/\*\*

\* 获得单科成绩的最低分

\* **@param** course

\* **@param** major

\* **@param** classe

\* **@param** grade

\* **@return**

\*/

**public** String getCourseLowestScore(String course,String major,String classe,String grade){

String lowest = **null**;

**try** {

ps = ct.prepareStatement("select MIN(Score) from tb\_Score where Student\_Id in(select Student\_Id from tb\_Student where Grade=? and Classe=? and Major\_Name=?) and Course\_Name=? " );

ps.setString(1, grade);

ps.setString(2, classe);

ps.setString(3, major);

ps.setString(4, course);

rs = ps.executeQuery();

**if**(rs.next()){

lowest = rs.getString(1);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** lowest;

}

/\*\*

\* 获得该班级有成绩的人数

\* **@param** major

\* **@param** classe

\* **@param** grade

\* **@return**

\*/

**public** String getHaveScoreCount(String major,String classe,String grade){

String count = **null**;

**try** {

// 获得本班级的人数，根据年级，班级，课程名字

ps = ct.prepareStatement("select Count(\*) from tb\_Score where Student\_Id in(select Student\_Id from tb\_Student where Grade=? and Classe=? and Major\_Name=?) GROUP BY Course\_Name");

ps.setString(1, grade);

ps.setString(2, classe);

ps.setString(3, major);

rs = ps.executeQuery();

**if**(rs.next()){

count = rs.getString(1);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** count;

}

/\*\*

\* 获得成绩分析的结果

\* **@param** major 专业

\* **@param** classe 班级

\* **@param** grade 年级

\* **@return** 返回分析集合

\*/

**public** Vector<AnalyzeResult> analyzeScore(String major,String classe,String grade ){

Vector<AnalyzeResult> vector = **new** Vector<AnalyzeResult>();

**try** {

// 根据年级的降序，根据学生ID来分组

ps = ct.prepareStatement("select Student\_Id,Student\_Name,SUM(Score),Avg(Score) as score from tb\_Score where Student\_Id in (Select Student\_Id from tb\_Student where Grade=? and Classe=? and Major\_Name=? )GROUP BY Student\_Id order by score desc");

ps.setString(1, grade);

ps.setString(2, classe);

ps.setString(3, major);

rs = ps.executeQuery();

**while**(rs.next()){

AnalyzeResult result = **new** AnalyzeResult();

result.setStudentId(rs.getString(1));

result.setStudentName(rs.getString(2));

result.setSumScore(rs.getString(3));

result.setAvgScore(rs.getString(4));

vector.add(result);

}

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

**return** vector;

}

//关闭数据库资源

**public** **void** close() {

**try** {

**if**(rs!=**null**) rs.close();

**if**(ps!=**null**) ps.close();

**if**(ct!=**null**) ct.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

|  |
| --- |
| **ManageHelper.java** |

**package** studentsystem.dao;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.util.HashMap;

**import** java.util.LinkedHashMap;

**import** java.util.Vector;

**import** studentsystem.bean.AnalyzeResult;

**import** studentsystem.bean.Student;

**import** studentsystem.bean.User;

//数据库业务处理类

**public** **class** ManageHelper {

**private** JdbcHelper helper; //与数据库通信的对象

/\*\*

\* 登陆业务处理

\* **@param** user 用户对象

\* **@return** 返回是否成功登陆

\*/

**public** **boolean** Login(User user){

**boolean** b = **true**;

helper = **new** JdbcHelper(); //创建与数据库通信的对象

User newUser = helper.getUser(user); //获得用户数据

**if**(!user.getPassword().equals(newUser.getPassword())){ //比对密码与数据库中的对应密码是否一致

b = **false**;

}

helper.close();//关闭资源

**return** b;

}

/\*\*

\* 注册业务处理

\* **@param** user 用户对象

\* **@return** 返回是否注册成功

\*/

**public** **boolean** Register(User user){

helper = **new** JdbcHelper();//创建与数据库通信的对象

**boolean** b = helper.register(user);

helper.close();//关闭资源

**return** b;

}

/\*\*

\* 检查是否重复登陆的方法

\* **@param** user 用户对象

\* **@return** 是否重复登陆,登陆过的返回true,否则返回false

\*/

**public** **boolean** Check\_IsLogin(User user){

**boolean** b = **true**;

**int** isLogin;

helper = **new** JdbcHelper();//创建与数据库通信的对象

User newUser = helper.getUser(user);

**if**(newUser.getIsLogin()==0){

b = **false**;

}

helper.close();//关闭资源

**return** b;

}

/\*\*

\* 返回成功修改登陆情况

\* **@param** user 用户对象

\*/

**public** **boolean** Update\_IsLogin(User user){

helper = **new** JdbcHelper();//创建与数据库通信的对象

**boolean** b = helper.update\_IsLogin(user);

helper.close();//关闭资源

**return** b;

}

/\*\*

\* 修改密码业务处理

\* **@param** user 用户对象

\* **@param** new\_Password 新密码

\* **@return** 返回是否修改成功

\*/

**public** **boolean** update\_Password(User user,String new\_Password){

**boolean** b;

helper = **new** JdbcHelper();//创建与数据库通信的对象

b = helper.update\_Password(user, new\_Password);

helper.close();//关闭资源

**return** b;

}

/\*\*

\* 获得所有院系

\* **@return** 返回院系HashMap集合

\*/

**public** HashMap<String, String>getAllDepartment(){

helper = **new** JdbcHelper();

HashMap<String, String> map;

map = helper.getAllDepartment();

helper.close();//关闭资源

**return** map;

}

/\*\*

\* 获得所有专业

\* **@return** 返回所有专业集合

\*/

**public** HashMap<String, String>getAllMajor(){

helper = **new** JdbcHelper();

HashMap<String, String> map;

map = helper.getAllMajor();

helper.close();//关闭资源

**return** map;

}

/\*\*

\* 根据department\_ID返回对应的专业

\* **@param** department\_ID 院系编号

\* **@return** 返回对应的专业集合

\*/

**public** Vector<String> getMajor(String department\_ID){

helper = **new** JdbcHelper();

Vector<String> vector;

vector = helper.getMajor(department\_ID);

helper.close();

**return** vector;

}

/\*\*

\* 添加学生业务

\* **@param** student 学生对象

\* **@return** 返回是否添加成功

\*/

**public** **boolean** addStudent(Student student){

**boolean** b = **true**;

helper = **new** JdbcHelper();

b = helper.addStudent(student);

helper.close();//关闭资源

**return** b;

}

/\*\*

\* 修改学生业务

\* **@param** newStudent 新学生对象

\* **@param** oldStudentID 旧学生信息

\* **@return** 返回是否添加成功

\*/

**public** **boolean** updateStudent(Student newStudent,String oldStudentID){

**boolean** b = **true**;

helper = **new** JdbcHelper();

b = helper.updateStudent(newStudent, oldStudentID);

helper.close();//关闭资源

**return** b;

}

/\*\*

\* 删除学生业务

\* **@param** studentID 学生学号

\* **@return** 返回是否删除成功

\*/

**public** **boolean** deleteStudent(String studentID){

**boolean** b = **true**;

helper = **new** JdbcHelper();

b = helper.deleteStudent(studentID);

helper.close();

**return** b;

}

/\*\*

\* 根据sql语句返回特定的学生对象集合

\* **@param** sql sql语句

\* **@return** 返回学生集合

\*/

**public** Vector<Student> getStudent(String sql){

Vector<Student> students;

helper = **new** JdbcHelper();

students = helper.getStudent(sql);

helper.close();

**return** students;

}

/\*\*

\* 根据年级和专业编号编号返回对应的班级

\* **@param**

\* **@return**

\*/

**public** Vector<String> getAllClasse(String grade,String major\_ID){

Vector<String> vector;

helper = **new** JdbcHelper();

vector = helper.getAllClasse(grade,major\_ID);

helper.close();

**return** vector;

}

/\*\*

\* 根据专业编号返回对应的所有课程

\* **@param** major\_Id

\* **@return** 返回课程集合

\*/

**public** Vector<String> getCourse(String major\_Id,String grade){

Vector<String> vector;

helper = **new** JdbcHelper();

vector = helper.getCourse(major\_Id,grade);

helper.close();

**return** vector;

}

/\*\*

\* 添加学生成绩

\*/

**public** **boolean** addStudentScore(Student student,Vector<String> courses){

**boolean** b;

helper = **new** JdbcHelper();

b = helper.addStudentScore(student,courses);

helper.close();

**return** b;

}

/\*\*

\* 修改学生成绩

\* **@param** student\_Id 学生学号

\* **@param** course\_Name 课程名字

\* **@param** score 成绩

\* **@return** 返回是否修改成功

\*/

**public** **boolean** updateStudentScore(String student\_Id,String course\_Name,String score){

**boolean** b;

helper = **new** JdbcHelper();

b = helper.updateStudentScore(student\_Id, course\_Name, score);

helper.close();

**return** b;

}

/\*\*

\* 查询学生成绩

\* **@param** student\_Id 学生学号

\* **@param** course\_Name 课程名称

\* **@return** 返回成绩集合

\*/

**public** HashMap<String, String> getStudentScore(String student\_Id){

HashMap<String, String> map;

helper = **new** JdbcHelper();

map = helper.getStudentScore(student\_Id);

helper.close();

**return** map;

}

// /\*\*

// \* 根据学号和课程名查询指定的成绩

// \* @param student\_Id 学号

// \* @param course\_Name 课程名

// \* @return 返回成绩

// \*/

// public String getStudentScore(String student\_Id,String course\_Name){

// String score = null;

// helper = new JdbcHelper();

// score = helper.getStudentScore(student\_Id, course\_Name);

// helper.close();

// return score;

// }

//

/\*\*

\* 根据特定的sql语句返回学生集合

\* **@param** sql

\* **@return**

\*/

**public** Vector<Student> getScores(String sql){

Vector<Student> students;

helper = **new** JdbcHelper();

students = helper.getScores(sql);

helper.close();

**return** students;

}

/\*\*

\* 获得班级总成绩

\* **@param** classe 班级

\* **@param** grade 年级

\* **@param** major 专业

\* **@return** 总成绩

\*/

**public** String getClasse\_SumScore(String classe,String grade,String major){

String sum = **null**;

helper = **new** JdbcHelper();

sum = helper.getClasse\_SumScore(classe, grade, major);

helper.close();

**return** sum;

}

/\*\*

\* 根据学生id返回该学生的平均成绩

\* **@param** student\_Id

\* **@return**

\*/

**public** String getStudentAvgScore(String student\_Id){

String avg = **null**;

helper = **new** JdbcHelper();

avg = helper.getStudentAvgScore(student\_Id);

helper.close();

**return** avg;

}

/\*\*

\* 根据学生专业，年级，班级，科目，考试成绩类型(优，良，中，及格，不及格)返回该类型的学生人数

\* **@param** grade

\* **@param** classe

\* **@param** major

\* **@param** type

\* **@param** course\_Name

\* **@return**

\*/

**public** String getCount\_ScoreType(String grade,String classe,String major,String type,String course\_Name){

String count = **null**;

helper = **new** JdbcHelper();

count = helper.getCount\_ScoreType(grade, classe, major, type, course\_Name);

helper.close();

**return** count;

}

/\*\*

\* 获得课程的平均分

\* **@param** course 课程名

\* **@param** major 专业

\* **@return** 平均分

\*/

**public** String getCourseAvg(String course,String major,String classe,String grade){

String avg = **null**;

helper = **new** JdbcHelper();

avg = helper.getCourseAvg(course, major, classe, grade);

helper.close();

**return** avg;

}

/\*\*

\* 获得单科成绩的最高分

\* **@param** course

\* **@param** major

\* **@param** classe

\* **@param** grade

\* **@return**

\*/

**public** String getCourseHighestScore(String course,String major,String classe,String grade){

String highest = **null**;

helper = **new** JdbcHelper();

highest = helper.getCourseHighestScore(course, major, classe, grade);

helper.close();

**return** highest;

}

/\*\*

\* 获得单科成绩的最低分

\* **@param** course

\* **@param** major

\* **@param** classe

\* **@param** grade

\* **@return**

\*/

**public** String getCourseLowestScore(String course,String major,String classe,String grade){

String lowest = **null**;

helper = **new** JdbcHelper();

lowest = helper.getCourseLowestScore(course, major, classe, grade);

helper.close();

**return** lowest;

}

/\*\*

\* 获得班级的平均成绩

\* **@param** classe

\* **@param** grade

\* **@param** major

\* **@return**

\*/

**public** String getClasseAvg(String classe,String grade,String major){

String sum = getClasse\_SumScore(classe,grade,major);

helper = **new** JdbcHelper();

String num = helper.getHaveScoreCount(major, classe, grade);

**if**(sum!=**null** && num!=**null**){

**return** String.*valueOf*((Double.*parseDouble*(sum)/Double.*parseDouble*(num)));

}**else**{

**return** "";

}

}

/\*\*

\* 获得成绩分析的结果

\* **@param** major 专业

\* **@param** classe 班级

\* **@param** grade 年级

\* **@return** 返回分析集合

\*/

**public** Vector<AnalyzeResult> analyzeScore(String major,String classe,String grade ){

Vector<AnalyzeResult> vector ;

helper = **new** JdbcHelper();

vector = helper.analyzeScore(major, classe, grade);

helper.close();

**return** vector;

}

}

**8.2创建数据库语句的代码**

|  |
| --- |
| **CreateSql.java** |

**package** studentsystem.util;

//生成sql语句的工具类

**public** **class** CreateSql {

// 重载创建数据库语句

//根据查询内容、选项从学生表里返回特定的sql语句

**public** **static** String getStudent\_Sql(String str,String option){

String sql = **null**;

**if**("全部".equals(option)){

sql = "select \* from tb\_Student" ;

}**else** **if**("学号".equals(option)){

sql = "select \* from tb\_Student where Student\_Id like '%"+str+"%'";

}**else** **if**("姓名".equals(option)){

sql = "select \* from tb\_Student where Student\_Name like '%"+str+"%'";

}**else** **if**("性别".equals(option)){

sql = "select \* from tb\_Student where Student\_Sex like '%"+str+"%'";

}**else** **if**("年级".equals(option)){

sql = "select \* from tb\_Student where Grade like '%"+str+"%'";

}**else** **if**("班级".equals(option)){

sql = "select \* from tb\_Student where Classe like '%"+str+"%'";

}**else** **if**("专业".equals(option)){

sql = "select \* from tb\_Student where Major\_Name like '%"+str+"%'";

}**else** **if**("院系".equals(option)){

sql = "select \* from tb\_Student where Department\_Name like '%"+str+"%'";

}

**return** sql;

}

//多条件查询的sql语句创建

**public** **static** String getConditions\_Sql(String id,String name,String sex,String grade,String department,String major,String classe){

StringBuilder sql = **new** StringBuilder("select \* from tb\_Student where 1=1");

**if**(!id.equals("")){

sql.append(" and Student\_Id like '%" + id + "%' ");

}

**if**(!name.equals("")){

sql.append(" and Student\_Name like '%" + name + "%' ");

}

**if**(!sex.equals("")){

sql.append(" and Student\_Sex like '%" + sex + "%' ");

}

**if**(!grade.equals("")){

sql.append(" and Grade like '%" + grade + "%' ");

}

**if**(!department.equals("")){

sql.append(" and Department\_Name like '%" + department + "%' ");

}

**if**(!major.equals("")){

sql.append(" and Major\_Name like '%" + major + "%' ");

}

**if**(!classe.equals("")){

sql.append(" and Classe like '%" + classe + "%' ");

}

**return** sql.toString();

}

//根据查询内容、选项从学生表里返回特定的sql语句

**public** **static** String getStudent\_Sql(String grade,String major,String str,String option){

String sql = **null**;

**if**("全部".equals(option)){

sql = "select \* from tb\_Student where Grade='"+grade+"' and Major\_Name='"+major+"'" ;

}**else** **if**("学号".equals(option)){

sql = "select \* from tb\_Student where Student\_Id like '%"+str+"%' and Grade='"+grade+"' and Major\_Name='"+major+"'" ;

}**else** **if**("姓名".equals(option)){

sql = "select \* from tb\_Student where Student\_Name like '%"+str+"%' and Grade='"+grade+"' and Major\_Name='"+major+"'" ;

}**else** **if**("性别".equals(option)){

sql = "select \* from tb\_Student where Student\_Sex like '%"+str+"%' and Grade='"+grade+"' and Major\_Name='"+major+"'" ;

}**else** **if**("年级".equals(option)){

sql = "select \* from tb\_Student where Grade like '%"+str+"%' and Grade='"+grade+"' and Major\_Name='"+major+"'" ;

}**else** **if**("班级".equals(option)){

sql = "select \* from tb\_Student where Classe like '%"+str+"%' and Grade='"+grade+"' and Major\_Name='"+major+"'" ;

}

**return** sql;

}

}

**8.3获得当前年的代码**

获得当前年份，程序的当前年份随着时间更新而更新，这样学生管理系统就能适应年份变化。

|  |
| --- |
| **Tools.java** |

**package** studentsystem.util;

**import** java.util.Calendar;

**import** java.util.Vector;

**public** **class** Tools {

//获取年级

**public** **static** Vector<String> CreateGrade(){

Vector<String> vector = **new** Vector<String>();

vector.add(""); //添加一个空选项

Calendar c = Calendar.*getInstance*();

**int** Year = c.get(Calendar.***YEAR***);

**for**(**int** i=0;i<4;i++){

vector.add(String.*valueOf*(Year--));

}

**return** vector;

}

//生成学生学号的方法(学号：department+major+grade+classe+num)

**public** **static** String CreateID(String grade,String classe,String major,String department,String num){

String id = department+major+grade.charAt(2)+""+grade.charAt(3)+classe+num;

**return** id;

}

}

**8.4设置窗口大小及位置代码**

|  |
| --- |
| **WindowUtil.java** |

**package** studentsystem.util;

**import** java.awt.Container;

**import** java.awt.Dimension;

**import** java.awt.Toolkit;

**public** **class** WindowUtil {

//设置窗体居中

**public** **static** **void** setFrameCenter(Container c) {

//获取工具对象

Toolkit tk = Toolkit.getDefaultToolkit();

//获取屏幕的宽和高

Dimension d = tk.getScreenSize();

**double** srceenWidth = d.getWidth();

**double** srceenHeigth = d.getHeight();

//获取窗体的宽和高

**int** frameWidth = c.getWidth();

**int** frameHeight = c.getHeight();

//获取新的宽和高

**int** width = (**int**) (srceenWidth - frameWidth) / 2;

**int** height = (**int**) (srceenHeigth - frameHeight) / 2;

//设置窗体坐标

c.setLocation(width, height);

}

}

**8.6 管理员学生信息类和构造方法**

|  |
| --- |
| **AnayzeResult.java** |

**package** studentsystem.bean;

//分析结果类

// 学生成绩分析的结果

**public** **class** AnalyzeResult {

**private** String studentId; //学生学号

**private** String studentName; //学生姓名

**private** String sumScore; //学生总成绩

**private** String avgScore; //学生平均成绩

**public** String getStudentId() {

**return** studentId;

}

**public** **void** setStudentId(String studentId) {

**this**.studentId = studentId;

}

**public** String getStudentName() {

**return** studentName;

}

**public** **void** setStudentName(String studentName) {

**this**.studentName = studentName;

}

**public** String getSumScore() {

**return** sumScore;

}

**public** **void** setSumScore(String sumScore) {

**this**.sumScore = sumScore;

}

**public** String getAvgScore() {

**return** avgScore;

}

**public** **void** setAvgScore(String avgScore) {

**this**.avgScore = avgScore;

}

}

|  |
| --- |
| **Student.java** |

**package** studentsystem.bean;

**import** java.util.HashMap;

**import** java.util.List;

// 学生的姓名，学号，性别，班级，年级，成绩的类

**public** **class** Student {

**private** String student\_Name; //学生姓名

**private** String student\_ID; //学生学号

**private** String sex; //学生性别

**private** String classe; //学生班级

**private** String grade; //学生年级

**public** HashMap<String, String> getScores() {

**return** scores;

}

**public** **void** setScores(HashMap<String, String> scores) {

**this**.scores = scores;

}

**private** String Major\_ID; //学生所属专业边编号

**private** String Department\_ID; //学生所属院系编号

**private** String Major\_Name; //学生所属院系名称

**private** String Department\_Name;//学生所属专业名称

**private** HashMap<String, String> scores; //学生成绩

**public** String getMajor\_Name() {

**return** Major\_Name;

}

**public** **void** setMajor\_Name(String major\_Name) {

Major\_Name = major\_Name;

}

**public** String getDepartment\_Name() {

**return** Department\_Name;

}

**public** **void** setDepartment\_Name(String department\_Name) {

Department\_Name = department\_Name;

}

**public** String getStudent\_Name() {

**return** student\_Name;

}

**public** **void** setStudent\_Name(String student\_Name) {

**this**.student\_Name = student\_Name;

}

**public** String getStudent\_ID() {

**return** student\_ID;

}

**public** **void** setStudent\_ID(String student\_ID) {

**this**.student\_ID = student\_ID;

}

**public** String getSex() {

**return** sex;

}

**public** **void** setSex(String sex) {

**this**.sex = sex;

}

**public** String getClasse() {

**return** classe;

}

**public** **void** setClasse(String classe) {

**this**.classe = classe;

}

**public** String getGrade() {

**return** grade;

}

**public** **void** setGrade(String grade) {

**this**.grade = grade;

}

**public** String getMajor\_ID() {

**return** Major\_ID;

}

**public** **void** setMajor\_ID(String major\_ID) {

Major\_ID = major\_ID;

}

**public** String getDepartment\_ID() {

**return** Department\_ID;

}

**public** **void** setDepartment\_ID(String department\_ID) {

Department\_ID = department\_ID;

}

}

|  |
| --- |
| **User.java** |

**package** studentsystem.bean;

// 一个用户类，包括用户名，密码，以及用户是否登陆添加其构造器

**public** **class** User {

**private** String username; //用户名

**private** String password; //密码

**private** **int** isLogin = 0; //用户是否登陆

**public** **int** getIsLogin() {

**return** isLogin;

}

**public** **void** setIsLogin(**int** isLogin) {

**this**.isLogin = isLogin;

}

**public** String getUsername() {

**return** username;

}

**public** **void** setUsername(String username) {

**this**.username = username;

}

**public** String getPassword() {

**return** password;

}

**public** **void** setPassword(String password) {

**this**.password = password;

}

}

|  |
| --- |
| **ScoreAnalyzeModel.java** |

**package** studentsystem.model;

**import** studentsystem.bean.AnalyzeResult;

**import** studentsystem.bean.Student;

**import** studentsystem.dao.ManageHelper;

**import** java.util.HashMap;

**import** java.util.Vector;

**import** javax.swing.JDialog;

**import** javax.swing.JOptionPane;

**import** javax.swing.table.AbstractTableModel;

**public** **class** ScoreAnalyzeModel **extends** AbstractTableModel{

**private** ManageHelper helper;

**private** Vector<Student> students;

**private** Vector<String> columnNames = **null**; //列名

**private** Vector<Vector<String>> rowData = **null**; //行数据

**public** ScoreAnalyzeModel(String major\_Name,String grade,String classe) {

helper = **new** ManageHelper();

Vector<String> courses = helper.getCourse(helper.getAllMajor().get(major\_Name),grade);//获得课程

Vector<AnalyzeResult> results = helper.analyzeScore(major\_Name, classe, grade); //得到分析的结果集合

columnNames = **new** Vector<String>();

rowData = **new** Vector<Vector<String>>();

columnNames.add("学号");

columnNames.add("姓名");

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

columnNames.add(courses.get(i));

}

columnNames.add("总成绩");

columnNames.add("平均成绩");

columnNames.add("排名");

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

Vector<String> hang = **new** Vector<String>();

AnalyzeResult result = results.get(i);

hang.add(result.getStudentId());

hang.add(result.getStudentName());

HashMap<String, String> scores = helper.getStudentScore(result.getStudentId()); //得到所有课程

**for**(**int** j=0;j<scores.size();j++){

String score = scores.get(courses.get(j)); //得到成绩

hang.add(score);

}

hang.add(result.getSumScore());

String avg = result.getAvgScore();

hang.add(avg.charAt(0)+""+avg.charAt(1)+""+avg.charAt(2)+""+avg.charAt(3));

hang.add(getRowCount()+1+"");

rowData.add(hang);

}

}

//得到共有多少行

@Override

**public** **int** getRowCount() {

// **TODO** Auto-generated method stub

**return** **this**.rowData.size();

}

//得到共有多少列

@Override

**public** **int** getColumnCount() {

// **TODO** Auto-generated method stub

**return** **this**.columnNames.size();

}

//得到某行某列的数据

@Override

**public** Object getValueAt(**int** rowIndex, **int** columnIndex) {

// **TODO** Auto-generated method stub

**return** ((Vector)**this**.rowData.get(rowIndex)).get(columnIndex);

}

//重写方法 getColumnName

@Override

**public** String getColumnName(**int** column) {

// **TODO** Auto-generated method stub

**return** (String)**this**.columnNames.get(column);

}

}

|  |
| --- |
| **ScoreModel.java** |

**package** studentsystem.model;

**import** studentsystem.bean.Student;

**import** studentsystem.dao.ManageHelper;

**import** java.util.HashMap;

**import** java.util.Vector;

**import** javax.swing.JDialog;

**import** javax.swing.JOptionPane;

**import** javax.swing.table.AbstractTableModel;

**public** **class** ScoreModel **extends** AbstractTableModel{

**private** ManageHelper helper;

**private** Vector<Student> students;

**private** Vector<String> columnNames = **null**; //列名

**private** Vector<Vector<String>> rowData = **null**; //行数据

**public** ScoreModel(String sql,String major\_Name,String grade,JDialog jd) {

helper = **new** ManageHelper();

students = helper.getStudent(sql);

Vector<String> courses = helper.getCourse(helper.getAllMajor().get(major\_Name),grade);//获得课程

columnNames = **new** Vector<String>();

rowData = **new** Vector<Vector<String>>();

columnNames.add("学号");

columnNames.add("姓名");

columnNames.add("性别");

columnNames.add("年级");

columnNames.add("班级");

columnNames.add("院系");

columnNames.add("专业");

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

columnNames.add(courses.get(i));

}

**for**(Student student : students){

Vector<String> hang = **new** Vector<String>();

hang.add(student.getStudent\_ID());

hang.add(student.getStudent\_Name());

hang.add(student.getSex());

hang.add(student.getGrade());

hang.add(student.getClasse());

hang.add(student.getDepartment\_Name());

hang.add(student.getMajor\_Name());

HashMap<String, String> scores = helper.getStudentScore(student.getStudent\_ID()); //得到所有课程

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

**if**(scores.size()!=0){

String score = scores.get(courses.get(i)); //得到成绩

hang.add(score);

}**else**{

hang.add("");

}

}

rowData.add(hang);

}

**if**(getRowCount()!=0){

JOptionPane.showMessageDialog(jd, "一共有"+getRowCount()+"条记录！");

**return** ;

}**else**{

JOptionPane.showMessageDialog(jd, "没有任何记录！");

**return** ;

}

}

//得到共有多少行

@Override

**public** **int** getRowCount() {

// **TODO** Auto-generated method stub

**return** **this**.rowData.size();

}

//得到共有多少列

@Override

**public** **int** getColumnCount() {

// **TODO** Auto-generated method stub

**return** **this**.columnNames.size();

}

//得到某行某列的数据

@Override

**public** Object getValueAt(**int** rowIndex, **int** columnIndex) {

// **TODO** Auto-generated method stub

**return** ((Vector)**this**.rowData.get(rowIndex)).get(columnIndex);

}

//重写方法 getColumnName

@Override

**public** String getColumnName(**int** column) {

// **TODO** Auto-generated method stub

**return** (String)**this**.columnNames.get(column);

}

}

|  |
| --- |
| **StudentModel.java** |

**package** studentsystem.model;

**import** studentsystem.bean.Student;

**import** studentsystem.dao.ManageHelper;

**import** java.util.Vector;

**import** javax.swing.JDialog;

**import** javax.swing.JOptionPane;

**import** javax.swing.table.AbstractTableModel;

//rowData用来存放行数据

//columnNames存放列名

**public** **class** StudentModel **extends** AbstractTableModel {

**private** ManageHelper helper;

**private** Vector<Student> students;

**private** Vector<String> columnNames = **null**; //列名

**private** Vector<Vector<String>> rowData = **null**; //行数据

**public** StudentModel(String sql,JDialog jd) {

helper = **new** ManageHelper();

students = helper.getStudent(sql);

columnNames = **new** Vector<String>();

rowData = **new** Vector<Vector<String>>();

columnNames.add("学号");

columnNames.add("姓名");

columnNames.add("性别");

columnNames.add("年级");

columnNames.add("班级");

columnNames.add("院系");

columnNames.add("专业");

**for**(Student student : students){

Vector<String> hang = **new** Vector<String>();

hang.add(student.getStudent\_ID());

hang.add(student.getStudent\_Name());

hang.add(student.getSex());

hang.add(student.getGrade());

hang.add(student.getClasse());

hang.add(student.getDepartment\_Name());

hang.add(student.getMajor\_Name());

rowData.add(hang);

}

**if**(getRowCount()!=0){

JOptionPane.showMessageDialog(jd, "一共有"+getRowCount()+"条记录！");

**return** ;

}**else**{

JOptionPane.showMessageDialog(jd, "没有任何记录！");

**return** ;

}

}

//得到共有多少行

@Override

**public** **int** getRowCount() {

// **TODO** Auto-generated method stub

**return** **this**.rowData.size();

}

//得到共有多少列

@Override

**public** **int** getColumnCount() {

// **TODO** Auto-generated method stub

**return** **this**.columnNames.size();

}

//得到某行某列的数据

@Override

**public** Object getValueAt(**int** rowIndex, **int** columnIndex) {

// **TODO** Auto-generated method stub

**return** ((Vector)**this**.rowData.get(rowIndex)).get(columnIndex);

}

//重写方法 getColumnName

@Override

**public** String getColumnName(**int** column) {

// **TODO** Auto-generated method stub

**return** (String)**this**.columnNames.get(column);

}

}

**8.6 各类窗口的代码**

**8.6.1 添加学生窗口**

|  |
| --- |
| **AddStudentFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.bean.Student;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.util.Tools;

**import** studentsystem.util.WindowUtil;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.HashMap;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

//添加学生界面

**public** **class** AddStudentFrame **extends** JDialog{

**private** JButton add\_Button; //"添加"按钮。

**private** JButton cancel\_Button; //"取消"按钮。

**private** JComboBox sex\_Box; //"性别"选项。

**private** JComboBox major\_Box; //"专业"选项。

**private** JComboBox department\_Box; //"所属院系"选项。

**private** JLabel student\_ID; //"学号"标签。

**private** JLabel student\_Name; //"姓名"标签。

**private** JLabel sex\_Label; //"性别"标签。

**private** JLabel classe\_Label; //"班级"标签。

**private** JLabel grade\_Label; //"年级标签"。

**private** JLabel major\_Label; //"专业"标签。

**private** JLabel department\_Label; //"所属院系"标签。

**private** JTextField student\_IDText; //"学号"文本域。

**private** JTextField student\_NameText; //"姓名"文本域。

**private** JComboBox classe\_Box; //"班级"文本域。

**private** JComboBox grade\_Box; //"年级"文本域。

**private** JDialog jd; //当前窗口。

**private** HashMap<String, String> departments; //所有院系集合

**private** HashMap<String, String> all\_Major; //所有专业集合

**private** Vector<String> majors; //专业名称集合

**private** ManageHelper helper; //数据库业务处理对象

**private** Vector<String> classes; //班级集合

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** AddStudentFrame(JFrame owner, String title, **boolean** modal){

**super**(owner, title, modal);

helper = **new** ManageHelper(); //创建数据库业务处理对象

departments = **this**.helper.getAllDepartment(); //获得所有院系

all\_Major = **this**.helper.getAllMajor(); //获得所有的专业

**this**.jd = **this**;

**this**.setSize(350,429); //设置窗体大小。

**this**.setLayout(**null**); //设置空布局。

student\_ID = **new** JLabel("班号:");

student\_ID.setBounds(78, 48, 30, 20);

**this**.add(student\_ID);

student\_IDText = **new** JTextField();

student\_IDText.setBounds(116, 48, 150, 20);

**this**.add(student\_IDText);

student\_Name = **new** JLabel("姓名:");

student\_Name.setBounds(78, 88, 30, 20);

**this**.add(student\_Name);

student\_NameText = **new** JTextField();

student\_NameText.setBounds(116, 88, 150, 20);

**this**.add(student\_NameText);

sex\_Label = **new** JLabel("性别:");

sex\_Label.setBounds(78, 128, 30, 20);

**this**.add(sex\_Label);

sex\_Box = **new** JComboBox(**new** String[]{"","男","女"});

sex\_Box.setBounds(116, 128, 60, 20);

**this**.add(sex\_Box);

grade\_Label = **new** JLabel("年级:");

grade\_Label.setBounds(78, 168, 30, 20);

**this**.add(grade\_Label);

grade\_Box = **new** JComboBox<String>(Tools.CreateGrade()); //需要获得获得年级选项

grade\_Box.setBounds(116, 168, 150, 20);

//注册"年级选项"事件监听

grade\_Box.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

// **TODO** Auto-generated method stub

classe\_Box.removeAllItems();//移除"班级选项"的内容

String option = major\_Box.getSelectedItem().toString();

String major\_id = all\_Major.get(option); //专业编号

String grade = grade\_Box.getSelectedItem().toString();

**if**(!grade.equals("")){

classes = helper.getAllClasse(grade,major\_id); //获得班级

**for**(String s : classes){

classe\_Box.addItem(s);

}

}

}

});

**this**.add(grade\_Box);

department\_Label = **new** JLabel("院系:");

department\_Label.setBounds(78, 208, 30, 20);

**this**.add(department\_Label);

department\_Box = **new** JComboBox(departments.keySet().toArray());//获得键的集合

//注册"院系"选项事件监听

department\_Box.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

major\_Box.removeAllItems();//移除"专业选项"的内容

String option = department\_Box.getSelectedItem().toString();//获得选项名称

String department\_ID = departments.get(option); //获得院系编号

**if**(!department\_ID.equals("")){

majors = helper.getMajor(department\_ID); //获得专业

**for**(String s : majors){

major\_Box.addItem(s);

}

}

}

});

department\_Box.setBounds(116, 208, 150, 20);

**this**.add(department\_Box);

major\_Label = **new** JLabel("专业:");

major\_Label.setBounds(78, 248, 30, 20);

**this**.add(major\_Label);

major\_Box = **new** JComboBox(**new** String[]{""});//防止空指针异常

//注册"专业"选项事件监听

major\_Box.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

**if**(major\_Box.getSelectedItem()!=**null**){ //防止空指针

**if**(!major\_Box.getSelectedItem().toString().equals("")){

**if**(major\_Box.getSelectedItem().toString().equals("") || grade\_Box.getSelectedItem()==**null** || grade\_Box.getSelectedItem().toString().equals("")){

JOptionPane.showMessageDialog(jd, "年级不能为空", "", JOptionPane.WARNING\_MESSAGE);

major\_Box.setSelectedIndex(0); //设置为空选项

**return** ;

}

classe\_Box.removeAllItems();//移除"班级选项"的内容

String option = major\_Box.getSelectedItem().toString();

String major\_id = all\_Major.get(option); //专业编号

String grade = grade\_Box.getSelectedItem().toString();

**if**(!grade.equals("")){

classes = helper.getAllClasse(grade,major\_id); //获得班级

**for**(String s : classes){

classe\_Box.addItem(s);

}

}

}

}

}

});

major\_Box.setBounds(116, 248, 150, 20);

**this**.add(major\_Box);

classe\_Label = **new** JLabel("班级:"); //需要获得班级选项

classe\_Label.setBounds(78, 288, 30, 20);

**this**.add(classe\_Label);

classe\_Box = **new** JComboBox(**new** String[]{""});//防止空指针异常

classe\_Box.setBounds(116, 288, 150, 20);

**this**.add(classe\_Box);

add\_Button = **new** JButton("添加");

add\_Button.setBounds(70, 330, 60, 25);

//注册"确认"按钮事件监听

add\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

Student student = **new** Student();

String sid = student\_IDText.getText().trim();

String name = student\_NameText.getText().trim();

String sex = sex\_Box.getSelectedItem().toString();

String classe = classe\_Box.getSelectedItem().toString();

String grade = grade\_Box.getSelectedItem().toString();

String department\_ID = **null**;

String major\_ID = **null**;

String department\_Name = **null**;

String major\_Name = **null**;

//数据校验部分

**if**(sid.equals("")){

JOptionPane.showMessageDialog(jd, "班号不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(sid.length()!=2){

JOptionPane.showMessageDialog(jd, "班号必须是两位数！", "", JOptionPane.WARNING\_MESSAGE);

student\_IDText.setText("");

**return** ;

}

**if**(name.equals("")){

JOptionPane.showMessageDialog(jd, "姓名不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(sex.equals("")){

JOptionPane.showMessageDialog(jd, "性别不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(grade.equals("")){

JOptionPane.showMessageDialog(jd, "年级不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(classe.equals("")){

JOptionPane.showMessageDialog(jd, "班级不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(department\_Box.getSelectedItem()==**null**){ //先检查再用

JOptionPane.showMessageDialog(jd, "院系不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

department\_Name = department\_Box.getSelectedItem().toString(); //获得院系名称

department\_ID = departments.get(department\_Name); //获得院系编号

}

**if**(department\_ID.equals("")){

JOptionPane.showMessageDialog(jd, "院系不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(major\_Box.getSelectedItem()==**null**){ //先检查再用

JOptionPane.showMessageDialog(jd, "专业不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

major\_Name = major\_Box.getSelectedItem().toString();//获得专业名称

major\_ID = all\_Major.get(major\_Name); //获得专业编号

}

**if**(major\_ID.equals("")){

JOptionPane.showMessageDialog(jd, "专业不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

String id = Tools.CreateID(grade, classe, major\_ID, department\_ID, sid);//生成学号

student\_ID.setText("学号:");

student\_IDText.setText(id); //设置学号文本域

JOptionPane.showMessageDialog(jd, "该学生的id为:"+id);

student.setStudent\_ID(id);

student.setStudent\_Name(name);

student.setSex(sex);

student.setGrade(grade);

student.setClasse(classe);

student.setMajor\_Name(major\_Name);

student.setDepartment\_Name(department\_Name);

student.setMajor\_ID(major\_ID);

student.setDepartment\_ID(department\_ID);

**if**(helper.addStudent(student)){

JOptionPane.showMessageDialog(jd, "添加成功！");

jd.dispose(); //关闭当前窗口

**return** ;

}**else**{

JOptionPane.showMessageDialog(jd, "添加失败！", "", JOptionPane.WARNING\_MESSAGE);

jd.dispose(); //关闭当前窗口

**return** ;

}

}

});

**this**.add(add\_Button);

cancel\_Button = **new** JButton("取消");

cancel\_Button.setBounds(230, 330, 60, 25);

//注册"取消"按钮事件监听

cancel\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

jd.dispose();

}

});

**this**.add(cancel\_Button);

WindowUtil.setFrameCenter(**this**);

**this**.setResizable(**false**);

**this**.setVisible(**true**);

}

}

**8.6.2 查询学生窗口**

|  |
| --- |
| **ConditionsQueryFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.CreateSql;

**import** studentsystem.util.WindowUtil;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

//多条件查询界面

**public** **class** ConditionsQueryFrame **extends** JDialog{

**private** JLabel student\_ID; //"学号"标签

**private** JLabel student\_Name; //"姓名"标签

**private** JLabel sex\_Label; //"性别"标签。l

**private** JLabel classe\_Label; //"班级"标签

**private** JLabel grade\_Label; //"年级标签"

**private** JLabel major\_Label; //"专业"标签

**private** JLabel department\_Label; //"所属院系"标签

**private** JTextField student\_IDText; //"学号"文本域

**private** JTextField student\_NameText; //"姓名"文本域

**private** JTextField sex\_Text; //性别选项

**private** JTextField grade\_Text; //年级选项

**private** JTextField department\_Text; //系别

**private** JTextField major\_Text; //专业

**private** JTextField classe\_Text; //班级

**private** JButton conditions\_button; //多条件查询按钮

**private** ManageHelper helper; //数据库业务处理对象

**private** JDialog jd; //当前窗口

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** ConditionsQueryFrame(JDialog owner, String title, **boolean** modal,JTable jt){

**super**(owner, title, modal);

**this**.jd = **this**;

**this**.setLayout(**null**);

student\_ID = **new** JLabel("学号:");

student\_ID.setBounds(29, 19, 30, 20);

**this**.add(student\_ID);

student\_IDText = **new** JTextField();

student\_IDText.setBounds(65, 19, 100, 20);

**this**.add(student\_IDText);

student\_Name = **new** JLabel("姓名:");

student\_Name.setBounds(200, 19, 30, 20);

**this**.add(student\_Name);

student\_NameText = **new** JTextField();

student\_NameText.setBounds(240, 19, 100, 20);

**this**.add(student\_NameText);

sex\_Label = **new** JLabel("性别:");

sex\_Label.setBounds(29, 50, 30, 20);

**this**.add(sex\_Label);

sex\_Text = **new** JTextField();

sex\_Text.setBounds(65, 50, 100, 20);

**this**.add(sex\_Text);

grade\_Label = **new** JLabel("年级:");

grade\_Label.setBounds(200, 50, 30, 20);

**this**.add(grade\_Label);

grade\_Text = **new** JTextField();

grade\_Text.setBounds(240, 50, 100, 20);

**this**.add(grade\_Text);

department\_Label = **new** JLabel("院系:");

department\_Label.setBounds(29, 83, 30, 20);

**this**.add(department\_Label);

department\_Text = **new** JTextField();

department\_Text.setBounds(65, 83, 100, 20);

**this**.add(department\_Text);

major\_Label = **new** JLabel("专业:");

major\_Label.setBounds(200, 83, 30, 20);

**this**.add(major\_Label);

major\_Text = **new** JTextField();

major\_Text.setBounds(240, 83, 100, 20);

**this**.add(major\_Text);

classe\_Label = **new** JLabel("班级:");

classe\_Label.setBounds(29,116, 30, 20);

**this**.add(classe\_Label);

classe\_Text = **new** JTextField();

classe\_Text.setBounds(65, 116, 100, 20);

**this**.add(classe\_Text);

conditions\_button = **new** JButton("多条件查询");

conditions\_button.setBounds(230, 130, 100, 30);

//注册"多条件查询"按钮事件监听

conditions\_button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String id = student\_IDText.getText().trim();

String name = student\_NameText.getText().trim();

String sex = sex\_Text.getText().trim();

String grade = grade\_Text.getText().trim();

String department = department\_Text.getText().trim();

String major = major\_Text.getText().trim();

String classe = classe\_Text.getText().trim();

**if**(id.equals("")&&name.equals("")&&sex.equals("")&&grade.equals("")&&department.equals("")&&major.equals("")&&classe.equals("")){

JOptionPane.showMessageDialog(jd, "条件不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

String sql = CreateSql.getConditions\_Sql(id, name, sex, grade, department, major, classe);

StudentModel sm = **new** StudentModel(sql,jd);

jt.setModel(sm);

jd.dispose();

}

}

});

**this**.add(conditions\_button);

**this**.setSize(411, 222);

// 设置窗体大小不可调整

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);

**this**.setVisible(**true**);

}

}

|  |
| --- |
| **QueryStudentFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.CreateSql;

**import** studentsystem.util.WindowUtil;

**import** java.awt.BorderLayout;

**import** java.awt.Container;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

//查询学生信息界面

**public** **class** QueryStudentFrame **extends** JDialog{

**private** JPanel jp1,jp2,jp3; //面板。

**private** JLabel query\_Label; //标签。

**private** JButton query\_Button; //"查询"按钮。

**private** JComboBox query\_List; //"查询"选项。

**private** JButton preciseQuery\_Button; //"精确查询"按钮。

**private** JButton details\_Button; //"详细信息"按钮。

**private** JTextField query\_Text; //"查询"文本域。

**private** JTable jt; //表格。

**private** JScrollPane jsp; //滚动条。

**private** JDialog jd; //当前窗口。

**private** StudentModel studentModel;//学生数据模型类

**private** **static** Vector<String> *query\_Option*;

**static** {

*query\_Option* = **new** Vector<String>();

*query\_Option*.add("全部");

*query\_Option*.add("学号");

*query\_Option*.add("姓名");

*query\_Option*.add("性别");

*query\_Option*.add("班级");

*query\_Option*.add("年级");

*query\_Option*.add("专业");

*query\_Option*.add("院系");

}

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** QueryStudentFrame(JFrame owner, String title, **boolean** modal){

**super**(owner, title, modal);

**this**.jd = **this**;

Container c = **this**.getContentPane();

jp1 = **new** JPanel();

query\_Label = **new** JLabel("请输入查询信息:");

jp1.add(query\_Label);

query\_Text = **new** JTextField(10);

jp1.add(query\_Text);

query\_List = **new** JComboBox<String>(*query\_Option*);

jp1.add(query\_List);

query\_Button = **new** JButton("查询");

//注册"查询"按钮事件监听

query\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String str = query\_Text.getText().trim(); //查询内容

String option = query\_List.getSelectedItem().toString(); //查询选项

String sql = CreateSql.getStudent\_Sql(str, option); //获得sql语句

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);//更新Jtable

}

});

jp1.add(query\_Button);

preciseQuery\_Button = **new** JButton("多条件查询");

//注册"多条件查询"按钮事件监听

preciseQuery\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

ConditionsQueryFrame frame = **new** ConditionsQueryFrame(jd, "多条件查询", **true**, jt);

}

});

jp1.add(preciseQuery\_Button);

c.add(jp1,BorderLayout.NORTH); //添加面板

jp2 = **new** JPanel();

jt = **new** JTable();

String sql = CreateSql.getStudent\_Sql(**null**, "全部");//查询全部内容

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);

jsp = **new** JScrollPane(jt);

jp2.add(jsp);

c.add(jp2,BorderLayout.CENTER); //添加面板

jp3 = **new** JPanel();

details\_Button = **new** JButton("详细信息");

//注册"详细信息"按钮事件监听

details\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**int** rowNum = jt.getSelectedRow();

**if**(rowNum==-1){

JOptionPane.showMessageDialog(jd, "请选择一行！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

DetailsFrame detailsFrame = **new** DetailsFrame(jd, "详细学生信息", **true**,rowNum,studentModel);

}

});

jp3.add(details\_Button);

c.add(jp3,BorderLayout.SOUTH);

**this**.setSize(600,540);

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);//设置窗体居中。

**this**.setVisible(**true**);

}

}

**8.6.3 修改学生窗口**

|  |
| --- |
| **ModifyStudentFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.CreateSql;

**import** studentsystem.util.WindowUtil;

**import** java.awt.BorderLayout;

**import** java.awt.Container;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

//修改学生信息界面

**public** **class** ModifyStudentFrame **extends** JDialog{

**private** JPanel jp1,jp2,jp3; //面板。

**private** JLabel query\_Label; //标签。

**private** JButton query\_Button; //"查询"按钮。

**private** JComboBox query\_List; //"查询"选项。

**private** JButton preciseQuery\_Button; //"精确查询"按钮。

**private** JButton modify\_Button; //"详细信息"按钮。

**private** JTextField query\_Text; //"查询"文本域。

**private** JTable jt; //表格。

**private** JScrollPane jsp; //滚动条。

**private** JDialog jd; //当前窗口。

**private** StudentModel studentModel; //学生数据模型

**private** **static** Vector<String> *query\_Option*;

**static** {

*query\_Option* = **new** Vector<String>();

*query\_Option*.add("全部");

*query\_Option*.add("学号");

*query\_Option*.add("姓名");

*query\_Option*.add("性别");

*query\_Option*.add("班级");

*query\_Option*.add("年级");

*query\_Option*.add("专业");

*query\_Option*.add("院系");

}

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** ModifyStudentFrame(JFrame owner, String title, **boolean** modal){

**super**(owner, title, modal);

**this**.jd = **this**;

Container c = **this**.getContentPane();

jp1 = **new** JPanel();

query\_Label = **new** JLabel("请输入查询信息:");

jp1.add(query\_Label);

query\_Text = **new** JTextField(10);

jp1.add(query\_Text);

query\_List = **new** JComboBox<String>(*query\_Option*);

jp1.add(query\_List);

query\_Button = **new** JButton("查询");

//注册"查询"按钮事件监听

query\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String str = query\_Text.getText().trim(); //查询内容

String option = query\_List.getSelectedItem().toString(); //查询选项

String sql = CreateSql.getStudent\_Sql(str, option); //获得sql语句

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);//更新Jtable

}

});

jp1.add(query\_Button);

preciseQuery\_Button = **new** JButton("多条件查询");

//注册"多条件查询"按钮事件监听

preciseQuery\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

ConditionsQueryFrame frame = **new** ConditionsQueryFrame(jd, "多条件查询", **true**, jt);

}

});

jp1.add(preciseQuery\_Button);

c.add(jp1,BorderLayout.NORTH); //添加面板

jp2 = **new** JPanel();

jt = **new** JTable();

String sql = CreateSql.getStudent\_Sql(**null**, "全部");//查询全部内容

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);

jsp = **new** JScrollPane(jt);

jp2.add(jsp);

c.add(jp2,BorderLayout.CENTER); //添加面板

jp3 = **new** JPanel();

modify\_Button = **new** JButton("修改信息");

//注册"修改信息"按钮事件监听

modify\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

**int** rowNum = jt.getSelectedRow();

**if**(rowNum==-1){

JOptionPane.showMessageDialog(jd, "请选择一行！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

ModifyFrame frame = **new** ModifyFrame(jd, "修改学生信息", **true**, rowNum, studentModel);

//更新

//构建新的数据模型类，并更新

String sql = CreateSql.getStudent\_Sql(**null**, "全部");//查询全部内容

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);

}

});

jp3.add(modify\_Button);

c.add(jp3,BorderLayout.SOUTH);

**this**.setSize(600,540);

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);//设置窗体居中。

**this**.setVisible(**true**);

}

}

|  |
| --- |
| **ModifyFrame.java** |

**package** studentsystem.frame;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.HashMap;

**import** java.util.Vector;

**import** studentsystem.bean.Student;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.Tools;

**import** studentsystem.util.WindowUtil;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

//修改学生信息界面

**public** **class** ModifyFrame **extends** JDialog{

**private** JButton modify\_Button; //"修改"按钮。

**private** JButton cancel\_Button; //"取消"按钮。

**private** JComboBox sex\_Box; //"性别"选项。

**private** JComboBox major\_Box; //"专业"选项。

**private** JComboBox department\_Box; //"所属院系"选项。

**private** JLabel student\_ID; //"学号"标签。

**private** JLabel student\_Name; //"姓名"标签。

**private** JLabel sex\_Label; //"性别"标签。

**private** JLabel classe\_Label; //"班级"标签。

**private** JLabel grade\_Label; //"年级标签"。

**private** JLabel major\_Label; //"专业"标签。

**private** JLabel department\_Label; //"所属院系"标签。

**private** JTextField student\_IDText; //"学号"文本域。

**private** JTextField student\_NameText; //"姓名"文本域。

**private** JComboBox classe\_Box; //"班级"选项。

**private** JComboBox grade\_Box; //"年级"选项。

**private** JDialog jd; //当前窗口。

**private** HashMap<String, String> departments; //所有院系集合

**private** HashMap<String, String> all\_Major; //所有专业集合

**private** Vector<String> majors; //专业名称集合

**private** ManageHelper helper; //数据库业务处理对象

**private** Vector<String> classes; //班级集合

**private** StudentModel sm; //学生数据模型对象

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** ModifyFrame(JDialog owner, String title, **boolean** modal,**int** rowNum,StudentModel sm){

**super**(owner, title, modal);

helper = **new** ManageHelper(); //创建数据库业务处理对象

departments = **this**.helper.getAllDepartment(); //获得所有院系

all\_Major = **this**.helper.getAllMajor(); //获得所有的专业

**this**.jd = **this**;

**this**.sm = sm;

**this**.setSize(350,429); //设置窗体大小。

**this**.setLayout(**null**); //设置空布局。

//获取信息

String student\_id = sm.getValueAt(rowNum, 0).toString();

String student\_name = sm.getValueAt(rowNum, 1).toString();

String student\_sex = sm.getValueAt(rowNum, 2).toString();

String student\_grade = sm.getValueAt(rowNum, 3).toString();

String student\_department = sm.getValueAt(rowNum, 5).toString();

String student\_major = sm.getValueAt(rowNum,6).toString();

String student\_classe = sm.getValueAt(rowNum, 4).toString();

String classid = String.*valueOf*(student\_id.charAt(student\_id.length()-2))+String.*valueOf*(student\_id.charAt(student\_id.length()-1));

student\_ID = **new** JLabel("班号:");

student\_ID.setBounds(78, 48, 30, 20);

**this**.add(student\_ID);

student\_IDText = **new** JTextField();

student\_IDText.setBounds(116, 48, 150, 20);

student\_IDText.setText(classid); //获取班号并显示

**this**.add(student\_IDText);

student\_Name = **new** JLabel("姓名:");

student\_Name.setBounds(78, 88, 30, 20);

**this**.add(student\_Name);

student\_NameText = **new** JTextField();

student\_NameText.setBounds(116, 88, 150, 20);

student\_NameText.setText(student\_name); //设置学生姓名并显示

**this**.add(student\_NameText);

sex\_Label = **new** JLabel("性别:");

sex\_Label.setBounds(78, 128, 30, 20);

**this**.add(sex\_Label);

sex\_Box = **new** JComboBox(**new** String[]{"","男","女"});

sex\_Box.setSelectedItem(sm.getValueAt(rowNum, 2));

sex\_Box.setBounds(116, 128, 60, 20);

**this**.add(sex\_Box);

grade\_Label = **new** JLabel("年级:");

grade\_Label.setBounds(78, 168, 30, 20);

**this**.add(grade\_Label);

grade\_Box = **new** JComboBox(Tools.CreateGrade());

grade\_Box.setBounds(116, 168, 150, 20);

//注册"年级选项"事件监听

grade\_Box.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

// **TODO** Auto-generated method stub

**if**(classe\_Box!=**null** ){ //防止空指针异常

classe\_Box.removeAllItems();//移除"班级选项"的内容

String option = major\_Box.getSelectedItem().toString();

String major\_id = all\_Major.get(option); //专业编号

String grade = grade\_Box.getSelectedItem().toString();

**if**(!grade.equals("")){

classes = helper.getAllClasse(grade,major\_id); //获得班级

**for**(String s : classes){

classe\_Box.addItem(s);

}

}

}

}

});

grade\_Box.setSelectedItem(sm.getValueAt(rowNum, 3)); //设置年级并显示

**this**.add(grade\_Box);

department\_Label = **new** JLabel("院系:");

department\_Label.setBounds(78, 208, 30, 20);

**this**.add(department\_Label);

department\_Box = **new** JComboBox(departments.keySet().toArray());//获得键的集合

department\_Box.setSelectedItem(sm.getValueAt(rowNum, 5));

//注册"院系"选项事件监听

department\_Box.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

addMajorOption();//增加专业的选项

}

});

department\_Box.setBounds(116, 208, 150, 20);

**this**.add(department\_Box);

major\_Label = **new** JLabel("专业:");

major\_Label.setBounds(78, 248, 30, 20);

**this**.add(major\_Label);

major\_Box = **new** JComboBox(**new** String[]{""});//防止空指针异常

addMajorOption(); //初始化专业的选项

major\_Box.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

addClasseOption(); //增加班级选项

}

});

major\_Box.setSelectedItem(sm.getValueAt(rowNum,6));

major\_Box.setBounds(116, 248, 150, 20);

**this**.add(major\_Box);

classe\_Label = **new** JLabel("班级:");

classe\_Label.setBounds(78, 288, 30, 20);

**this**.add(classe\_Label);

classe\_Box = **new** JComboBox(**new** String[]{""});//防止空指针异常

addClasseOption();//初始化班级选项

//注册"班级选项"事件监听

classe\_Box.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**if**(classe\_Box.getSelectedItem()!=**null**){ //防止空指针

**if**(!classe\_Box.getSelectedItem().toString().equals("")){

String classe = classe\_Box.getSelectedItem().toString();

String grade = grade\_Box.getSelectedItem().toString();

String major\_Name = major\_Box.getSelectedItem().toString();//获得专业名称

String major\_ID = all\_Major.get(major\_Name); //获得专业编号

}

}

}

});

classe\_Box.setBounds(116, 288, 150, 20);

classe\_Box.setSelectedItem(sm.getValueAt(rowNum, 4));

**this**.add(classe\_Box);

modify\_Button = **new** JButton("修改");

modify\_Button.setBounds(70, 330, 60, 25);

//注册"修改"按钮事件监听

modify\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

Student newStudent = **new** Student();

String id = student\_IDText.getText().trim();

String name = student\_NameText.getText().trim();

String sex = sex\_Box.getSelectedItem().toString();

String classe = classe\_Box.getSelectedItem().toString();

String grade = grade\_Box.getSelectedItem().toString();

String department\_ID = **null**;

String major\_ID = **null**;

String department\_Name = **null**;

String major\_Name = **null**;

//数据校验部分

**if**(id.equals("")){

JOptionPane.showMessageDialog(jd, "班号不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(id.length()!=2){

JOptionPane.showMessageDialog(jd, "班号必须是两位数！", "", JOptionPane.WARNING\_MESSAGE);

student\_IDText.setText("");

**return** ;

}

**if**(name.equals("")){

JOptionPane.showMessageDialog(jd, "姓名不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(sex.equals("")){

JOptionPane.showMessageDialog(jd, "性别不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(grade.equals("")){

JOptionPane.showMessageDialog(jd, "年级不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(classe.equals("")){

JOptionPane.showMessageDialog(jd, "班级不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(department\_Box.getSelectedItem()==**null**){ //先检查再用

JOptionPane.showMessageDialog(jd, "院系不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

department\_Name = department\_Box.getSelectedItem().toString(); //获得院系名称

department\_ID = departments.get(department\_Name); //获得院系编号

}

**if**(department\_ID.equals("")){

JOptionPane.showMessageDialog(jd, "院系不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(major\_Box.getSelectedItem()==**null**){ //先检查再用

JOptionPane.showMessageDialog(jd, "专业不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

major\_Name = major\_Box.getSelectedItem().toString();//获得专业名称

major\_ID = all\_Major.get(major\_Name); //获得专业编号

}

**if**(major\_ID.equals("")){

JOptionPane.showMessageDialog(jd, "专业不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

String ID = Tools.CreateID(grade, classe, major\_ID, department\_ID, id);//生成学号

student\_IDText.setText(ID); //设置学号文本域

JOptionPane.showMessageDialog(jd, "该学生的id为:"+ID);

newStudent.setStudent\_ID(ID);

newStudent.setStudent\_Name(name);

newStudent.setSex(sex);

newStudent.setGrade(grade);

newStudent.setClasse(classe);

newStudent.setMajor\_ID(major\_ID);

newStudent.setDepartment\_ID(department\_ID);

newStudent.setDepartment\_Name(department\_Name);

newStudent.setMajor\_Name(major\_Name);

**if**(helper.updateStudent(newStudent, student\_id)){

JOptionPane.showMessageDialog(jd, "修改成功！");

jd.dispose(); //关闭当前窗口

**return** ;

}**else**{

JOptionPane.showMessageDialog(jd, "修改失败！", "", JOptionPane.WARNING\_MESSAGE);

jd.dispose(); //关闭当前窗口

**return** ;

}

}

});

**this**.add(modify\_Button);

cancel\_Button = **new** JButton("取消");

cancel\_Button.setBounds(230, 330, 60, 25);

//注册"取消"按钮事件监听

cancel\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

jd.dispose();

}

});

**this**.add(cancel\_Button);

WindowUtil.setFrameCenter(**this**);

**this**.setResizable(**false**);

**this**.setVisible(**true**);

}

//增加专业的选项

**public** **void** addMajorOption(){

major\_Box.removeAllItems();//移除"专业选项"的内容（会触发事件监听）

String option = department\_Box.getSelectedItem().toString();//获得选项名称

String department\_ID = departments.get(option); //获得院系编号

**if**(!department\_ID.equals("")){

majors = helper.getMajor(department\_ID); //获得专业

**for**(String s : majors){

major\_Box.addItem(s);

}

}

}

//增加班级选项

**public** **void** addClasseOption(){

**if**(major\_Box.getSelectedItem()!=**null**){ //防止空指针

**if**(!major\_Box.getSelectedItem().toString().equals("")){

**if**(major\_Box.getSelectedItem().toString().equals("") || grade\_Box.getSelectedItem()==**null** || grade\_Box.getSelectedItem().toString().equals("")){

JOptionPane.showMessageDialog(jd, "年级不能为空", "", JOptionPane.WARNING\_MESSAGE);

major\_Box.setSelectedIndex(0); //设置为空选项

**return** ;

}

**if**(classe\_Box!=**null** ){//防止空指针异常

classe\_Box.removeAllItems();//移除"班级选项"的内容

String option = major\_Box.getSelectedItem().toString();

String major\_id = all\_Major.get(option); //专业编号

String grade = grade\_Box.getSelectedItem().toString();

**if**(!grade.equals("")){

classes = helper.getAllClasse(grade,major\_id); //获得班级

**for**(String s : classes){

classe\_Box.addItem(s);

}

}

}

}

}

}

}

**8.6.4 删除学生窗口**

|  |
| --- |
| **DeleteStudentFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.CreateSql;

**import** studentsystem.util.WindowUtil;

**import** java.awt.BorderLayout;

**import** java.awt.Container;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

//删除学生界面

**public** **class** DeleteStudentFrame **extends** JDialog{

**private** JPanel jp1,jp2,jp3; //面板。

**private** JLabel query\_Label; //标签。

**private** JButton query\_Button; //"查询"按钮

**private** JComboBox query\_List; //"查询"选项

**private** JButton preciseQuery\_Button; //"精确查询"按钮

**private** JButton delete\_Button; //"删除"按钮

**private** JTextField query\_Text; //"查询"文本域

**private** JTable jt; //表格

**private** JScrollPane jsp; //滚动条

**private** JDialog jd; //当前窗口

**private** StudentModel studentModel; //学生数据模型

**private** **static** Vector<String> *query\_Option*;

**static** {

*query\_Option* = **new** Vector<String>();

*query\_Option*.add("全部");

*query\_Option*.add("学号");

*query\_Option*.add("姓名");

*query\_Option*.add("性别");

*query\_Option*.add("班级");

*query\_Option*.add("年级");

*query\_Option*.add("专业");

*query\_Option*.add("院系");

}

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** DeleteStudentFrame(JFrame owner, String title, **boolean** modal){

**super**(owner, title, modal);

**this**.jd = **this**;

Container c = **this**.getContentPane();

jp1 = **new** JPanel();

query\_Label = **new** JLabel("请输入查询信息:");

jp1.add(query\_Label);

query\_Text = **new** JTextField(10);

jp1.add(query\_Text);

query\_List = **new** JComboBox<String>(*query\_Option*);

jp1.add(query\_List);

query\_Button = **new** JButton("查询");

//注册"查询"按钮事件监听

query\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String str = query\_Text.getText().trim(); //查询内容

String option = query\_List.getSelectedItem().toString(); //查询选项

String sql = CreateSql.getStudent\_Sql(str, option); //获得sql语句

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

// 将查询到的学生放入Jtable表中

jt.setModel(studentModel);//更新Jtable

}

});

jp1.add(query\_Button);

preciseQuery\_Button = **new** JButton("多条件查询");

preciseQuery\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

ConditionsQueryFrame frame = **new** ConditionsQueryFrame(jd, "多条件查询", **true**, jt);

}

});

jp1.add(preciseQuery\_Button);

c.add(jp1,BorderLayout.NORTH); //添加面板

jp2 = **new** JPanel();

jt = **new** JTable();

String sql = CreateSql.getStudent\_Sql(**null**, "全部");//查询全部内容

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);

//

jsp = **new** JScrollPane(jt);

jp2.add(jsp);

c.add(jp2,BorderLayout.CENTER); //添加面板

jp3 = **new** JPanel();

delete\_Button = **new** JButton("删除");

//注册"删除"按钮事件监听

delete\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

**int** rowNum = jt.getSelectedRow();

**if**(rowNum==-1){

JOptionPane.showMessageDialog(jd, "请选择一行！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

String studentID = jt.getValueAt(rowNum,0).toString();

ManageHelper helper = **new** ManageHelper();

**if**(helper.deleteStudent(studentID)){

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

//更新

//构建新的数据模型类，并更新

String sql = CreateSql.getStudent\_Sql(**null**, "全部");//查询全部内容

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);

**return** ;

}**else**{

JOptionPane.showMessageDialog(jd, "删除失败！","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}

}

});

jp3.add(delete\_Button);

c.add(jp3,BorderLayout.SOUTH);

**this**.setSize(600,540);

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);//设置窗体居中

**this**.setVisible(**true**);

}

}

**8.6.5 添加学生成绩窗口**

|  |
| --- |
| **AddStudentScoreFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.CreateSql;

**import** studentsystem.util.WindowUtil;

**import** java.awt.BorderLayout;

**import** java.awt.Container;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

**public** **class** AddStudentScoreFrame **extends** JDialog{

**private** JPanel jp1,jp2,jp3; //面板。

**private** JLabel query\_Label; //标签。

**private** JButton query\_Button; //"查询"按钮。

**private** JComboBox query\_List; //"查询"选项。

**private** JButton preciseQuery\_Button; //"精确查询"按钮。

**private** JButton addScore\_button; //"添加成绩"按钮。

**private** JTextField query\_Text; //"查询"文本域。

**private** JTable jt; //表格。

**private** JScrollPane jsp; //滚动条。

**private** JDialog jd; //当前窗口。

**private** StudentModel studentModel;//学生数据模型类

**private** **static** Vector<String> *query\_Option*;

**static** {

*query\_Option* = **new** Vector<String>();

*query\_Option*.add("全部");

*query\_Option*.add("学号");

*query\_Option*.add("姓名");

*query\_Option*.add("性别");

*query\_Option*.add("班级");

*query\_Option*.add("年级");

*query\_Option*.add("专业");

*query\_Option*.add("院系");

}

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** AddStudentScoreFrame(JFrame owner, String title, **boolean** modal){

**super**(owner, title, modal);

**this**.jd = **this**;

Container c = **this**.getContentPane();

jp1 = **new** JPanel();

query\_Label = **new** JLabel("请输入查询信息:");

jp1.add(query\_Label);

query\_Text = **new** JTextField(10);

jp1.add(query\_Text);

query\_List = **new** JComboBox<String>(*query\_Option*);

jp1.add(query\_List);

query\_Button = **new** JButton("查询");

//注册"查询"按钮事件监听

query\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String str = query\_Text.getText().trim(); //查询内容

String option = query\_List.getSelectedItem().toString(); //查询选项

String sql = CreateSql.getStudent\_Sql(str, option); //获得sql语句

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);//更新Jtable

}

});

jp1.add(query\_Button);

preciseQuery\_Button = **new** JButton("多条件查询");

//注册"多条件查询"按钮事件监听

preciseQuery\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

ConditionsQueryFrame frame = **new** ConditionsQueryFrame(jd, "多条件查询", **true**, jt);

}

});

jp1.add(preciseQuery\_Button);

c.add(jp1,BorderLayout.NORTH); //添加面板

jp2 = **new** JPanel();

jt = **new** JTable();

String sql = CreateSql.getStudent\_Sql(**null**, "全部");//查询全部内容

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);

jsp = **new** JScrollPane(jt);

jp2.add(jsp);

c.add(jp2,BorderLayout.CENTER); //添加面板

jp3 = **new** JPanel();

addScore\_button = **new** JButton("添加成绩");

//注册"详细信息"按钮事件监听

addScore\_button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**int** rowNum = jt.getSelectedRow();

**if**(rowNum==-1){

JOptionPane.showMessageDialog(jd, "请选择一行！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

AddScoreFrame frame = **new** AddScoreFrame(jd, "添加成绩", **true**, studentModel, rowNum);

}

});

jp3.add(addScore\_button);

c.add(jp3,BorderLayout.SOUTH);

**this**.setSize(600,540);

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);//设置窗体居中。

**this**.setVisible(**true**);

}

}

|  |
| --- |
| **AddScoreFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.bean.Student;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.WindowUtil;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.HashMap;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

//添加成绩界面

**public** **class** AddScoreFrame **extends** JDialog{

**private** ManageHelper helper;

**private** StudentModel sm;

**private** Vector<String> courses;

**private** JDialog jd;

**private** JButton add\_button; //添加按钮

**private** JButton cancel\_button; //取消

**private** HashMap<String, JTextField> jtextFieldHashMap;//管理文本域的集合

**private** HashMap<String, String> scores; //学生所有成绩

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** AddScoreFrame(JDialog owner, String title, **boolean** modal,StudentModel sm,**int** rowNum){

**super**(owner, title, modal);

**this**.jd = **this**;

**this**.setLayout(**null**);

helper = **new** ManageHelper();

**this**.sm = sm;

jtextFieldHashMap = **new** HashMap<String, JTextField>();

courses = helper.getCourse(helper.getAllMajor().get(sm.getValueAt(rowNum, 6)),sm.getValueAt(rowNum, 3).toString());//获得所有课目

scores = helper.getStudentScore(sm.getValueAt(rowNum, 0).toString());//根据学号获得该学生的所有科目成绩

**int** vgap = 0; //垂直间距

**if**(!scores.isEmpty()){

// 显示错误信息

JOptionPane.showMessageDialog(jd, "该学生已经有成绩了！","",JOptionPane.WARNING\_MESSAGE);

// 释放错误信息

jd.dispose();

// 返回

**return** ;

}

// setBounds(int x,int y,int width,int height);设置组件的x坐标，y坐标，及宽还有高

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

JLabel jLabel = **new** JLabel(courses.get(i)+":");

// 设置间隙大小

jLabel.setBounds(78, 48+vgap, 120, 20);

JTextField field = **new** JTextField();

field.setBounds(206, 48+vgap, 150, 20);

jtextFieldHashMap.put(courses.get(i),field); //添加入管理文本域的集合

**this**.add(jLabel);

**this**.add(field);

vgap += 30;

}

add\_button = **new** JButton("添加");

add\_button.setBounds(120, 48+vgap+5, 60, 20);

//注册"添加"按钮事件监听

add\_button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

// **TODO** Auto-generated method stub

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

JTextField field = jtextFieldHashMap.get(courses.get(i));

String score = field.getText().trim();

**if**(score.equals("")){

JOptionPane.showMessageDialog(jd, "成绩不能为空！","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(Double.*parseDouble*(score)<0 || Double.*parseDouble*(score)>100){

JOptionPane.showMessageDialog(jd, "请输入0~100之间的成绩", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

}

Student student = **new** Student();

student.setStudent\_ID(sm.getValueAt(rowNum, 0).toString());

student.setStudent\_Name(sm.getValueAt(rowNum, 1).toString());

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

JTextField field = jtextFieldHashMap.get(courses.get(i));

String score = field.getText().trim();

scores.put(courses.get(i), score);

}

student.setScores(scores);

**boolean** b = helper.addStudentScore(student,courses);

**if**(!b){

JOptionPane.showMessageDialog(jd, "添加失败！","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

JOptionPane.showMessageDialog(jd, "添加成功！","",JOptionPane.WARNING\_MESSAGE);

jd.dispose();

**return** ;

}

}

});

**this**.add(add\_button);

cancel\_button = **new** JButton("取消");

cancel\_button.setBounds(280, 48+vgap+5, 60, 20);

cancel\_button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

// **TODO** Auto-generated method stub

jd.dispose();

}

});

**this**.add(cancel\_button);

**this**.setSize(450, 48+vgap+78);

WindowUtil.setFrameCenter(**this**);

**this**.setVisible(**true**);

}

}

**8.6.6 修改学生成绩窗口**

|  |
| --- |
| **UpdateScoreFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.WindowUtil;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.File;

**import** java.util.HashMap;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

//修改成绩界面

**public** **class** UpdateScoreFrame **extends** JDialog{

**private** ManageHelper helper;

**private** StudentModel sm;

**private** Vector<String> courses;

**private** JDialog jd;

**private** JButton update\_button; //修改按钮

**private** JButton cancel\_button; //取消按钮

**private** HashMap<String, JTextField> jtextFieldHashMap;//管理文本域的集合

**private** HashMap<String, String> scores; //学生所有成绩

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** UpdateScoreFrame(JDialog owner, String title, **boolean** modal,StudentModel sm,**int** rowNum){

**super**(owner, title, modal);

**this**.jd = **this**;

**this**.setLayout(**null**);

helper = **new** ManageHelper();

**this**.sm = sm;

jtextFieldHashMap = **new** HashMap<String, JTextField>();

courses = helper.getCourse(helper.getAllMajor().get(sm.getValueAt(rowNum, 6)),sm.getValueAt(rowNum, 3).toString());//获得所有课程

**int** vgap = 0; //垂直间距

scores = helper.getStudentScore(sm.getValueAt(rowNum, 0).toString());//根据学号获得该学生的所有科目成绩

**if**(scores.isEmpty()){

JOptionPane.showMessageDialog(jd, "该同学还没有成绩！", "", JOptionPane.WARNING\_MESSAGE);

jd.dispose();

**return** ;

}

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

JLabel jLabel = **new** JLabel(courses.get(i)+":");

jLabel.setBounds(78, 48+vgap, 120, 20);

JTextField field = **new** JTextField();

field.setText(scores.get(courses.get(i)));//给文本域添加成绩

field.setBounds(206, 48+vgap, 150, 20);

jtextFieldHashMap.put(courses.get(i),field); //添加入管理文本域的集合

**this**.add(jLabel);

**this**.add(field);

vgap += 30;

}

update\_button = **new** JButton("修改");

update\_button.setBounds(120, 48+vgap+5, 60, 20);

//注册"修改"按钮事件监听

update\_button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

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

JTextField field = jtextFieldHashMap.get(courses.get(i));

String score = field.getText().trim();

**if**(score.equals("")){

JOptionPane.showMessageDialog(jd, "成绩不能为空！","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(Double.*parseDouble*(score)<0 || Double.*parseDouble*(score)>100){

JOptionPane.showMessageDialog(jd, "请输入0~100之间的成绩", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

}

**int** i;

**for**(i=0;i<courses.size();i++){

JTextField field = jtextFieldHashMap.get(courses.get(i));

String score = field.getText().trim();

**boolean** b = helper.updateStudentScore(sm.getValueAt(rowNum, 0).toString(),courses.get(i),score);

**if**(!b){

**break**;

}

}

**if**(i<courses.size()){

JOptionPane.showMessageDialog(jd, "修改失败！","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

JOptionPane.showMessageDialog(jd, "修改成功！","",JOptionPane.WARNING\_MESSAGE);

jd.dispose();

**return** ;

}

}

});

**this**.add(update\_button);

cancel\_button = **new** JButton("取消");

cancel\_button.setBounds(280, 48+vgap+5, 60, 20);

cancel\_button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

jd.dispose();

}

});

**this**.add(cancel\_button);

**this**.setSize(450, 48+vgap+78);

WindowUtil.setFrameCenter(**this**);

**this**.setVisible(**true**);

}

}

|  |
| --- |
| **UpdateStudentScoreFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.CreateSql;

**import** studentsystem.util.WindowUtil;

**import** java.awt.BorderLayout;

**import** java.awt.Container;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

**public** **class** UpdateStudentScoreFrame **extends** JDialog{

**private** JPanel jp1,jp2,jp3; //面板。

**private** JLabel query\_Label; //标签。

**private** JButton query\_Button; //"查询"按钮。

**private** JComboBox query\_List; //"查询"选项。

**private** JButton preciseQuery\_Button; //"精确查询"按钮。

**private** JButton update\_Button; //"修改成绩"按钮。

**private** JTextField query\_Text; //"查询"文本域。

**private** JTable jt; //表格。

**private** JScrollPane jsp; //滚动条。

**private** JDialog jd; //当前窗口。

**private** StudentModel studentModel;//学生数据模型类

**private** **static** Vector<String> *query\_Option*;

**static** {

*query\_Option* = **new** Vector<String>();

*query\_Option*.add("全部");

*query\_Option*.add("学号");

*query\_Option*.add("班级");

*query\_Option*.add("性别");

*query\_Option*.add("班级");

*query\_Option*.add("年级");

*query\_Option*.add("专业");

*query\_Option*.add("院系");

}

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** UpdateStudentScoreFrame(JFrame owner, String title, **boolean** modal){

**super**(owner, title, modal);

**this**.jd = **this**;

Container c = **this**.getContentPane();

jp1 = **new** JPanel();

query\_Label = **new** JLabel("请输入查询信息:");

jp1.add(query\_Label);

query\_Text = **new** JTextField(10);

jp1.add(query\_Text);

query\_List = **new** JComboBox<String>(*query\_Option*);

jp1.add(query\_List);

query\_Button = **new** JButton("查询");

//注册"查询"按钮事件监听

query\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String str = query\_Text.getText().trim(); //查询内容

String option = query\_List.getSelectedItem().toString(); //查询选项

String sql = CreateSql.getStudent\_Sql(str, option); //获得sql语句

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);//更新Jtable

}

});

jp1.add(query\_Button);

preciseQuery\_Button = **new** JButton("多条件查询");

//注册"多条件查询"按钮事件监听

preciseQuery\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

ConditionsQueryFrame frame = **new** ConditionsQueryFrame(jd, "多条件查询", **true**, jt);

}

});

jp1.add(preciseQuery\_Button);

c.add(jp1,BorderLayout.NORTH); //添加面板

jp2 = **new** JPanel();

jt = **new** JTable();

String sql = CreateSql.getStudent\_Sql(**null**, "全部");//查询全部内容

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);

jsp = **new** JScrollPane(jt);

jp2.add(jsp);

c.add(jp2,BorderLayout.CENTER); //添加面板

jp3 = **new** JPanel();

update\_Button = **new** JButton("修改成绩");

//注册"详细信息"按钮事件监听

update\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**int** rowNum = jt.getSelectedRow();

**if**(rowNum==-1){

JOptionPane.showMessageDialog(jd, "请选择一行！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

UpdateScoreFrame frame = **new** UpdateScoreFrame(jd, "修改成绩", **true**, studentModel, rowNum);

}

});

jp3.add(update\_Button);

c.add(jp3,BorderLayout.SOUTH);

**this**.setSize(600,540);

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);//设置窗体居中。

**this**.setVisible(**true**);

}

}

**8.6.7 学生成绩查询窗口**

|  |
| --- |
| **ConditionsQueryScoreFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.CreateSql;

**import** studentsystem.util.WindowUtil;

**import** java.awt.Color;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

//多条件查询成绩

**public** **class** ConditionsQueryScoreFrame **extends** JDialog{

**private** JLabel student\_ID; //"学号"标签

**private** JLabel student\_Name; //"姓名"标签

**private** JLabel sex\_Label; //"性别"标签。

**private** JLabel classe\_Label; //"班级"标签

**private** JLabel grade\_Label; //"年级标签"

**private** JLabel major\_Label; //"专业"标签

**private** JLabel department\_Label; //"所属院系"标签

**private** JTextField student\_IDText; //"学号"文本域

**private** JTextField student\_NameText; //"姓名"文本域

**private** JTextField sex\_Text; //性别选项

**private** JTextField grade\_Text; //年级选项

**private** JTextField department\_Text; //系别

**private** JTextField major\_Text; //专业

**private** JTextField classe\_Text; //班级

**private** JLabel course\_Label; //课程标签

**private** JTextField course\_Text; //课程文本域

**private** JButton conditions\_button; //多条件查询按钮

**private** ManageHelper helper; //数据库处理业务对象

**private** JDialog jd; //当前窗口

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** ConditionsQueryScoreFrame(JDialog owner, String title, **boolean** modal,JTable jt){

**super**(owner, title, modal);

**this**.jd = **this**;

**this**.setLayout(**null**);

setBackground(Color.WHITE);//设置背景颜色

student\_ID = **new** JLabel("学号:");

student\_ID.setBounds(29, 19, 30, 20);

**this**.add(student\_ID);

student\_IDText = **new** JTextField();

student\_IDText.setBounds(65, 19, 100, 20);

**this**.add(student\_IDText);

student\_Name = **new** JLabel("姓名:");

student\_Name.setBounds(200, 19, 30, 20);

**this**.add(student\_Name);

student\_NameText = **new** JTextField();

student\_NameText.setBounds(240, 19, 100, 20);

**this**.add(student\_NameText);

sex\_Label = **new** JLabel("性别:");

sex\_Label.setBounds(29, 50, 30, 20);

**this**.add(sex\_Label);

sex\_Text = **new** JTextField();

sex\_Text.setBounds(65, 50, 100, 20);

**this**.add(sex\_Text);

grade\_Label = **new** JLabel("年级:");

grade\_Label.setBounds(200, 50, 30, 20);

**this**.add(grade\_Label);

grade\_Text = **new** JTextField();

grade\_Text.setBounds(240, 50, 100, 20);

**this**.add(grade\_Text);

department\_Label = **new** JLabel("院系:");

department\_Label.setBounds(29, 83, 30, 20);

**this**.add(department\_Label);

department\_Text = **new** JTextField();

department\_Text.setBounds(65, 83, 100, 20);

**this**.add(department\_Text);

major\_Label = **new** JLabel("专业:");

major\_Label.setBounds(200, 83, 30, 20);

**this**.add(major\_Label);

major\_Text = **new** JTextField();

major\_Text.setBounds(240, 83, 100, 20);

**this**.add(major\_Text);

classe\_Label = **new** JLabel("班级:");

classe\_Label.setBounds(29,116, 30, 20);

**this**.add(classe\_Label);

classe\_Text = **new** JTextField();

classe\_Text.setBounds(65, 116, 100, 20);

**this**.add(classe\_Text);

conditions\_button = **new** JButton("多条件查询");

conditions\_button.setBounds(230, 130, 100, 30);

//注册"多条件查�?"按钮事件监听

conditions\_button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String id = student\_IDText.getText().trim();

String name = student\_NameText.getText().trim();

String sex = sex\_Text.getText().trim();

String grade = grade\_Text.getText().trim();

String department = department\_Text.getText().trim();

String major = major\_Text.getText().trim();

String classe = classe\_Text.getText().trim();

**if**(id.equals("")&&name.equals("")&&sex.equals("")&&grade.equals("")&&department.equals("")&&major.equals("")&&classe.equals("")){

JOptionPane.showMessageDialog(jd, "条件不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

String sql = CreateSql.getConditions\_Sql(id, name, sex, grade, department, major, classe);

StudentModel sm = **new** StudentModel(sql,jd);

jt.setModel(sm);

jd.dispose();

}

}

});

**this**.add(conditions\_button);

**this**.setSize(411, 222);

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);

**this**.setVisible(**true**);

}

}

|  |
| --- |
| **ScoreConditionsQueryFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.model.ScoreAnalyzeModel;

**import** studentsystem.model.ScoreModel;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.CreateSql;

**import** studentsystem.util.WindowUtil;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

//成绩多条件查询界面

**public** **class** ScoreConditionsQueryFrame **extends** JDialog{

**private** JLabel student\_ID; //"学号"标签。

**private** JLabel student\_Name; //"姓名"标签。

**private** JLabel sex\_Label; //"性别"标签。l

**private** JLabel classe\_Label; //"班级"标签。

**private** JLabel grade\_Label; //"年级标签"。

**private** JLabel major\_Label; //"专业"标签。

**private** JLabel department\_Label; //"所属院系"标签。

**private** JTextField student\_IDText; //"学号"文本域。

**private** JTextField student\_NameText; //"姓名"文本域。

**private** JTextField sex\_Text; //性别选项

**private** JTextField grade\_Text; //年级选项

**private** JTextField department\_Text; //系别

**private** JTextField major\_Text; //专业

**private** JTextField classe\_Text; //班级

**private** JButton conditions\_button; //多条件查询按钮

**private** ManageHelper helper;

**private** JDialog jd; //当前窗口

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** ScoreConditionsQueryFrame(JDialog owner, String title, **boolean** modal,JTable jt,String department,String major,String grade){

**super**(owner, title, modal);

**this**.jd = **this**;

**this**.setLayout(**null**);

student\_ID = **new** JLabel("学号:");

student\_ID.setBounds(29, 19, 30, 20);

**this**.add(student\_ID);

student\_IDText = **new** JTextField();

student\_IDText.setBounds(65, 19, 100, 20);

**this**.add(student\_IDText);

student\_Name = **new** JLabel("姓名:");

student\_Name.setBounds(200, 19, 30, 20);

**this**.add(student\_Name);

student\_NameText = **new** JTextField();

student\_NameText.setBounds(240, 19, 100, 20);

**this**.add(student\_NameText);

sex\_Label = **new** JLabel("性别:");

sex\_Label.setBounds(29, 50, 30, 20);

**this**.add(sex\_Label);

sex\_Text = **new** JTextField();

sex\_Text.setBounds(65, 50, 100, 20);

**this**.add(sex\_Text);

grade\_Label = **new** JLabel("年级:");

grade\_Label.setBounds(200, 50, 30, 20);

**this**.add(grade\_Label);

grade\_Text = **new** JTextField();

grade\_Text.setBounds(240, 50, 100, 20);

grade\_Text.setText(grade);

grade\_Text.setEditable(**false**);

**this**.add(grade\_Text);

department\_Label = **new** JLabel("院系:");

department\_Label.setBounds(29, 83, 30, 20);

**this**.add(department\_Label);

department\_Text = **new** JTextField();

department\_Text.setBounds(65, 83, 100, 20);

department\_Text.setText(department);

department\_Text.setEditable(**false**);

**this**.add(department\_Text);

major\_Label = **new** JLabel("专业:");

major\_Label.setBounds(200, 83, 30, 20);

**this**.add(major\_Label);

major\_Text = **new** JTextField();

major\_Text.setBounds(240, 83, 150, 20);

major\_Text.setText(major);

major\_Text.setEditable(**false**);

**this**.add(major\_Text);

classe\_Label = **new** JLabel("班级:");

classe\_Label.setBounds(29,116, 30, 20);

**this**.add(classe\_Label);

classe\_Text = **new** JTextField();

classe\_Text.setBounds(65, 116, 100, 20);

**this**.add(classe\_Text);

conditions\_button = **new** JButton("多条件查询");

conditions\_button.setBounds(230, 130, 100, 30);

//注册"多条件查询"按钮事件监听

conditions\_button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String id = student\_IDText.getText().trim();

String name = student\_NameText.getText().trim();

String sex = sex\_Text.getText().trim();

String grade = grade\_Text.getText().trim();

String department = department\_Text.getText().trim();

String major = major\_Text.getText().trim();

String classe = classe\_Text.getText().trim();

**if**(id.equals("")&&name.equals("")&&sex.equals("")&&grade.equals("")&&department.equals("")&&major.equals("")&&classe.equals("")){

JOptionPane.showMessageDialog(jd, "条件不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

String sql = CreateSql.getConditions\_Sql(id, name, sex, grade, department, major, classe);

ScoreModel sm = **new** ScoreModel(sql, major, grade, jd);

jt.setModel(sm);

jd.dispose();

}

}

});

**this**.add(conditions\_button);

**this**.setSize(411, 222);

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);

**this**.setVisible(**true**);

}

}

|  |
| --- |
| **QueryStudentScoreFrame.java** |

**package** studentsystem.frame;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.HashMap;

**import** java.util.Vector;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.model.ScoreModel;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.CreateSql;

**import** studentsystem.util.Tools;

**import** studentsystem.util.WindowUtil;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

// 查询学生成绩界面

**public** **class** QueryStudentScoreFrame **extends** JDialog{

**private** JComboBox major\_Box; //"专业"选项。

**private** JComboBox department\_Box; //"所属院系"选项。

**private** JLabel classe\_Label; //"班级"标签。

**private** JLabel grade\_Label; //"年级标签"。

**private** JLabel major\_Label; //"专业"标签。

**private** JLabel department\_Label; //"所属院系"标签。

**private** JComboBox classe\_Box; //"班级"文本域。

**private** JComboBox grade\_Box; //"年级"文本域。

**private** JButton query\_Button; //"查询"按钮。

**private** JButton conditions\_button; //多条件查询按钮

**private** JComboBox query\_List; //"查询"选项。

**private** JLabel query\_Label; //标签。

**private** JTextField query\_Text; //"查询"文本域。

**private** JButton details\_Button; //"详细信息"按钮

**private** JTable jt; //表格。

**private** JScrollPane jsp; //滚动条。

**private** JDialog jd; //当前窗口。

**private** HashMap<String, String> departments; //所有院系集合

**private** HashMap<String, String> all\_Major; //所有专业集合

**private** Vector<String> majors; //专业名称集合

**private** ManageHelper helper; //数据库业务处理对象

**private** ScoreModel scoreModel;

**private** **static** Vector<String> *query\_Option*;

**static** {

*query\_Option* = **new** Vector<String>();

*query\_Option*.add("全部");

*query\_Option*.add("学号");

*query\_Option*.add("姓名");

*query\_Option*.add("性别");

*query\_Option*.add("班级");

}

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** QueryStudentScoreFrame(JFrame owner, String title, **boolean** modal){

**super**(owner, title, modal);

**this**.setLayout(**null**);

**this**.jd = **this**;

helper = **new** ManageHelper(); //创建数据库业务处理对象

departments = **this**.helper.getAllDepartment(); //获得所有院系

all\_Major = **this**.helper.getAllMajor(); //获得所有的专业

grade\_Label = **new** JLabel("年级:");

grade\_Label.setBounds(25, 19, 30, 20);

**this**.add(grade\_Label);

grade\_Box = **new** JComboBox(Tools.CreateGrade());

grade\_Box.setBounds(60, 19, 60, 20);

**this**.add(grade\_Box);

department\_Label = **new** JLabel("院系:");

department\_Label.setBounds(130, 19, 30, 20);

**this**.add(department\_Label);

department\_Box = **new** JComboBox(departments.keySet().toArray());//获得键的集合

//注册"院系"选项事件监听

department\_Box.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

major\_Box.removeAllItems();//移除"专业选项"的内容

String option = department\_Box.getSelectedItem().toString();//获得选项名称

String department\_ID = departments.get(option); //获得院系编号

**if**(!department\_ID.equals("")){

majors = helper.getMajor(department\_ID); //获得专业

**for**(String s : majors){

major\_Box.addItem(s);

}

}

}

});

department\_Box.setBounds(165, 19, 150, 20);

**this**.add(department\_Box);

major\_Label = **new** JLabel("专业:");

major\_Label.setBounds(325, 19, 30, 20);

**this**.add(major\_Label);

major\_Box = **new** JComboBox(**new** String[]{""});//防止空指针异常

major\_Box.setBounds(360, 19, 150, 20);

**this**.add(major\_Box);

query\_Label = **new** JLabel("查询信息:");

query\_Label.setBounds(520, 19, 60, 20);

**this**.add(query\_Label);

query\_Text = **new** JTextField();

query\_Text.setBounds(585, 19, 100, 20);

**this**.add(query\_Text);

query\_List = **new** JComboBox(*query\_Option*);

query\_List.setBounds(695, 19, 65, 20);

**this**.add(query\_List);

query\_Button = **new** JButton("查询");

query\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

String grade = grade\_Box.getSelectedItem().toString();

String major = **null**;

String str = query\_Text.getText().trim();

String option = query\_List.getSelectedItem().toString();

**if**(grade.equals("")){

JOptionPane.showMessageDialog(jd, "年级不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**if**(department\_Box.getSelectedItem()==**null**){ //先检查再用

JOptionPane.showMessageDialog(jd, "院系不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(department\_Box.getSelectedItem().toString().equals("")){

JOptionPane.showMessageDialog(jd, "院系不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(major\_Box.getSelectedItem()==**null**){ //先检查再用

JOptionPane.showMessageDialog(jd, "专业不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

major = major\_Box.getSelectedItem().toString();//获得专业名称

}

**if**(major.equals("")){

JOptionPane.showMessageDialog(jd, "专业不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

//创建查询的sql语句

String sql = CreateSql.getStudent\_Sql(grade, major, str, option);

scoreModel = **new** ScoreModel(sql, major, grade, jd);

jt.setModel(scoreModel);

}

});

query\_Button.setBounds(770, 19, 65, 20);

**this**.add(query\_Button);

conditions\_button = **new** JButton("多条件查询");

conditions\_button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

String grade = grade\_Box.getSelectedItem().toString();

String major = **null**;

**if**(grade.equals("")){

JOptionPane.showMessageDialog(jd, "年级不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**if**(department\_Box.getSelectedItem()==**null**){ //先检查再用

JOptionPane.showMessageDialog(jd, "院系不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(department\_Box.getSelectedItem().toString().equals("")){

JOptionPane.showMessageDialog(jd, "院系不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(major\_Box.getSelectedItem()==**null**){ //先检查再用

JOptionPane.showMessageDialog(jd, "专业不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

major = major\_Box.getSelectedItem().toString();//获得专业名称

}

**if**(major.equals("")){

JOptionPane.showMessageDialog(jd, "专业不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

ScoreConditionsQueryFrame frame = **new** ScoreConditionsQueryFrame(jd, "多条件查询", **true**, jt, department\_Box.getSelectedItem().toString(), major,grade);

}

});

conditions\_button.setBounds(845, 19, 100, 20);

**this**.add(conditions\_button);

jt = **new** JTable();

jsp = **new** JScrollPane(jt);

jsp.setBounds(26, 45, 925, 480);

**this**.add(jsp);

details\_Button = **new** JButton("详细成绩");

//注册"详细信息"按钮事件监听

details\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**int** rowNum = jt.getSelectedRow();

**if**(rowNum==-1){

JOptionPane.showMessageDialog(jd, "请选择一行！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

QueryScoreFrame frame = **new** QueryScoreFrame(jd, "学生成绩", **true**, scoreModel, rowNum);

}

});

details\_Button.setBounds(450, 530, 100, 20);

**this**.add(details\_Button);

**this**.setSize(1000,600);

WindowUtil.setFrameCenter(**this**);

**this**.setVisible(**true**);

}

}

|  |
| --- |
| **QueryScoreFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.model.ScoreModel;

**import** studentsystem.util.WindowUtil;

**import** java.awt.Button;

**import** java.awt.GridLayout;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.HashMap;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JTextField;

//查询成绩界面

**public** **class** QueryScoreFrame **extends** JDialog{

**private** ManageHelper helper; //数据库处理业务对象

**private** ScoreModel sm; //学生成绩对象

**private** Vector<String> courses;

**private** JDialog jd;

**private** HashMap<String, JTextField> jtextFieldHashMap;//管理文本域的集合

**private** JButton confirm\_button; //"确定按钮"

**private** HashMap<String, String> scores; //学生所有成绩

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** QueryScoreFrame(JDialog owner, String title, **boolean** modal,ScoreModel sm,**int** rowNum){

**super**(owner, title, modal);

**this**.jd = **this**;

**this**.setLayout(**null**);

helper = **new** ManageHelper();

**this**.sm = sm;

jtextFieldHashMap = **new** HashMap<String, JTextField>();

// 根据专业和年级获得所有课程的信息

courses = helper.getCourse(helper.getAllMajor().get(sm.getValueAt(rowNum, 6)),sm.getValueAt(rowNum, 3).toString());//获得所有课程

// 根据学号获得学生的所有科目成绩

scores = helper.getStudentScore(sm.getValueAt(rowNum, 0).toString());//根据学号获得该学生的所有科目成绩

**int** vgap = 0; //垂直间距

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

JLabel jLabel = **new** JLabel(courses.get(i)+":");

jLabel.setBounds(78, 48+vgap, 120, 20);

JTextField field = **new** JTextField();

// 文本域设置为不可编辑

field.setEditable(**false**);

field.setText(scores.get(courses.get(i)));//给文本域添加成绩

field.setBounds(206, 48+vgap, 150, 20);

jtextFieldHashMap.put(courses.get(i),field); //添加入管理文本域的集合

**this**.add(jLabel);

**this**.add(field);

vgap += 30;

}

confirm\_button = **new** JButton("确定");

confirm\_button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

jd.dispose();

}

});

confirm\_button.setBounds(215, 48+vgap+5, 60, 20);

**this**.add(confirm\_button);

**this**.setSize(450, 48+vgap+78);

WindowUtil.setFrameCenter(**this**);

**this**.setVisible(**true**);

}

}

**8.6.8 学生成绩统计窗口**

|  |
| --- |
| **DetailsFrame.java** |

**package** studentsystem.frame;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.WindowUtil;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JTextField;

//详细信息界面

**public** **class** DetailsFrame **extends** JDialog{

**private** JButton confirm\_Button; //"确认"按钮。

**private** JTextField sex\_Text; //"性别"选项。

**private** JTextField major\_Text; //"专业"选项。

**private** JTextField department\_Text; //"所属院系"选项。

**private** JLabel student\_ID; //"学号"标签。

**private** JLabel student\_Name; //"姓名"标签。

**private** JLabel sex\_Label; //"性别"标签。

**private** JLabel classe\_Label; //"班级"标签。

**private** JLabel grade\_Label; //"年级标签"。

**private** JLabel major\_Label; //"专业"标签。

**private** JLabel department\_Label; //"所属院系"标签。

**private** JTextField student\_IDText; //"学号"文本域。

**private** JTextField student\_NameText; //"姓名"文本域。

**private** JTextField classe\_Text; //"班级"文本域。

**private** JTextField grade\_Text; //"年级"文本域。

**private** StudentModel sm; //传入的学生数据模型

**private** JDialog jd; //当前窗口。

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** DetailsFrame(JDialog owner, String title, **boolean** modal, **int** rowNum,StudentModel sm){

**super**(owner, title, modal);

**this**.sm = sm; //传入学生数据模型

**this**.jd = **this**;

**this**.setSize(350,429); //设置窗体大小。

**this**.setLayout(**null**); //设置空布局。

student\_ID = **new** JLabel("学号:");

student\_ID.setBounds(78, 48, 30, 20);

**this**.add(student\_ID);

student\_IDText = **new** JTextField();

student\_IDText.setEditable(**false**); //不可编辑

student\_IDText.setText(sm.getValueAt(rowNum, 0).toString()); //获取学号并显示

student\_IDText.setBounds(116, 48, 150, 20);

**this**.add(student\_IDText);

student\_Name = **new** JLabel("姓名:");

student\_Name.setBounds(78, 88, 30, 20);

**this**.add(student\_Name);

student\_NameText = **new** JTextField();

student\_NameText.setBounds(116, 88, 150, 20);

student\_NameText.setEditable(**false**);//不可编辑

student\_NameText.setText(sm.getValueAt(rowNum, 1).toString()); //设置学生姓名并显示

**this**.add(student\_NameText);

sex\_Label = **new** JLabel("性别:");

sex\_Label.setBounds(78, 128, 30, 20);

**this**.add(sex\_Label);

sex\_Text = **new** JTextField();

sex\_Text.setBounds(116, 128, 60, 20);

sex\_Text.setEditable(**false**);//不可编辑

sex\_Text.setText(sm.getValueAt(rowNum, 2).toString()); //设置学生性别并显示

**this**.add(sex\_Text);

grade\_Label = **new** JLabel("年级:");

grade\_Label.setBounds(78, 168, 30, 20);

**this**.add(grade\_Label);

grade\_Text = **new** JTextField();

grade\_Text.setBounds(116, 168, 150, 20);

grade\_Text.setEditable(**false**);

grade\_Text.setText(sm.getValueAt(rowNum, 3).toString()); //设置年级并显示

**this**.add(grade\_Text);

classe\_Label = **new** JLabel("班级:");

classe\_Label.setBounds(78, 208, 30, 20);

**this**.add(classe\_Label);

classe\_Text = **new** JTextField();

classe\_Text.setBounds(116, 208, 150, 20);

classe\_Text.setEditable(**false**);

classe\_Text.setText(sm.getValueAt(rowNum, 4).toString());

**this**.add(classe\_Text);

department\_Label = **new** JLabel("院系:");

department\_Label.setBounds(78, 248, 30, 20);

**this**.add(department\_Label);

department\_Text = **new** JTextField();

department\_Text.setEditable(**false**);

department\_Text.setText(sm.getValueAt(rowNum, 5).toString());

department\_Text.setBounds(116, 248, 150, 20);

**this**.add(department\_Text);

major\_Label = **new** JLabel("专业:");

major\_Label.setBounds(78, 288, 30, 20);

**this**.add(major\_Label);

major\_Text = **new** JTextField();

major\_Text.setEditable(**false**);

major\_Text.setBounds(116, 288, 150, 20);

major\_Text.setText(sm.getValueAt(rowNum, 6).toString());

**this**.add(major\_Text);

confirm\_Button = **new** JButton("确认");

//注册"确定"按钮事件监听

confirm\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

jd.dispose();//关闭当前窗口

}

});

confirm\_Button.setBounds(150, 330, 60, 25);

**this**.add(confirm\_Button);

WindowUtil.setFrameCenter(**this**);

**this**.setResizable(**false**);

**this**.setVisible(**true**);

}

}

|  |
| --- |
| **ScoreAnalyzeFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.model.ScoreAnalyzeModel;

**import** studentsystem.util.WindowUtil;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JDialog;

**import** javax.swing.JLabel;

**import** javax.swing.JTextField;

**public** **class** ScoreAnalyzeFrame **extends** JDialog{

**private** ManageHelper helper; //数据库业务处理对象

**private** Vector<String> courses;

**private** JLabel average\_Scores; //平均成绩标签

**private** JTextField average\_ScoresText; //平均成绩文本域

**private** JLabel total\_Score; //总成绩标签

**private** JTextField total\_ScoreText; //总成绩文本域

**private** JButton scores\_Ranking; //成绩排名按钮

**private** JDialog jd;

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** ScoreAnalyzeFrame(JDialog owner, String title, **boolean** modal,String major\_Id,String major\_Name,String grade,String classe){

**super**(owner, title, modal);

**this**.jd = **this**;

**this**.setLayout(**null**);

helper = **new** ManageHelper();

courses = helper.getCourse(major\_Id,grade);//获得所有课程

average\_Scores = **new** JLabel("班级平均成绩:");

average\_Scores.setBounds(200, 15, 90, 20);//班级平均成绩文本域

**this**.add(average\_Scores);

average\_ScoresText = **new** JTextField();

average\_ScoresText.setEditable(**false**);

//设置班级的平均成绩

average\_ScoresText.setText(helper.getClasseAvg(classe, grade, major\_Name));

average\_ScoresText.setBounds(295, 15, 60, 20);

**this**.add(average\_ScoresText);

total\_Score = **new** JLabel("班级总成绩:");

total\_Score.setBounds(435, 15, 80, 20);

**this**.add(total\_Score);

total\_ScoreText = **new** JTextField(); //班级总成绩文本域

total\_ScoreText.setEditable(**false**);

//设置班级总成绩

total\_ScoreText.setText(helper.getClasse\_SumScore(classe, grade, major\_Name));

total\_ScoreText.setBounds(520, 15, 60, 20);

**this**.add(total\_ScoreText);

**int** vgap = 0; //垂直间距

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

JTextField course = **new** JTextField(courses.get(i)+":"); //科目标签

course.setEditable(**false**);

course.setBounds(26, 48+vgap, 120, 20);

**this**.add(course);

JLabel highestScore = **new** JLabel("最高成绩:"); //最高成绩标签

highestScore.setBounds(156, 48+vgap, 60, 20);

**this**.add(highestScore);

JTextField highestScoreText = **new** JTextField();//最高成绩文本域

highestScoreText.setBounds(221, 48+vgap, 40, 20);

//设置最高成绩

highestScoreText.setText(helper.getCourseHighestScore(courses.get(i), major\_Name, classe, grade));

highestScoreText.setEditable(**false**);

**this**.add(highestScoreText);

JLabel lowest\_Score = **new** JLabel("最低成绩:"); //最低成绩标签

lowest\_Score.setBounds(261, 48+vgap, 60, 20);

**this**.add(lowest\_Score);

JTextField lowest\_ScoreText = **new** JTextField();//最低成绩文本域

lowest\_ScoreText.setBounds(326, 48+vgap, 40, 20);

//设置最低成绩

lowest\_ScoreText.setText(helper.getCourseLowestScore(courses.get(i), major\_Name, classe, grade));

lowest\_ScoreText.setEditable(**false**);

**this**.add(lowest\_ScoreText);

JLabel avg\_Score = **new** JLabel("平均成绩:"); //平均成绩标签

avg\_Score.setBounds(366, 48+vgap, 60, 20);

**this**.add(avg\_Score);

JTextField avg\_ScoreText = **new** JTextField(); //平均成绩文本域

avg\_ScoreText.setBounds(431, 48+vgap, 40, 20);

avg\_ScoreText.setEditable(**false**);

//计算平均成绩

String score = helper.getCourseAvg(courses.get(i), major\_Name, classe, grade);

**if**(score!=**null** && !score.equals("")){

avg\_ScoreText.setText(score.charAt(0)+""+score.charAt(1)+""+score.charAt(2)+""+score.charAt(3));

}

**this**.add(avg\_ScoreText);

JLabel j1 = **new** JLabel("优:"); //优人数标签

j1.setBounds(476, 48+vgap, 20, 20);

**this**.add(j1);

JTextField t1 = **new** JTextField();//优文本域

t1.setEditable(**false**);

t1.setBounds(498, 48+vgap, 40, 20);

//计算优秀人数

t1.setText(helper.getCount\_ScoreType(grade, classe, major\_Name, "优", courses.get(i)));

**this**.add(t1);

JLabel j2 = **new** JLabel("良:"); //良人数标签

j2.setBounds(548, 48+vgap, 20, 20);

**this**.add(j2);

JTextField t2 = **new** JTextField();//良文本域

t2.setEditable(**false**);

t2.setBounds(570, 48+vgap, 40, 20);

//计算良好人数

t2.setText(helper.getCount\_ScoreType(grade, classe, major\_Name, "良", courses.get(i)));

**this**.add(t2);

JLabel j3 = **new** JLabel("中:"); //中人数标签

j3.setBounds(620, 48+vgap, 20, 20);

**this**.add(j3);

JTextField t3 = **new** JTextField();//中文本域

t3.setEditable(**false**);

//计算中人数

t3.setText(helper.getCount\_ScoreType(grade, classe, major\_Name, "中", courses.get(i)));

t3.setBounds(642, 48+vgap, 40, 20);

**this**.add(t3);

JLabel j4 = **new** JLabel("及格:"); //及格人数标签

j4.setBounds(692, 48+vgap, 30, 20);

**this**.add(j4);

JTextField t4 = **new** JTextField();//及格文本域

t4.setEditable(**false**);

//计算及格人数

t4.setText(helper.getCount\_ScoreType(grade, classe, major\_Name, "及格", courses.get(i)));

t4.setBounds(724, 48+vgap, 40, 20);

**this**.add(t4);

JLabel j5 = **new** JLabel("不及格:"); //不及格人数标签

j5.setBounds(774, 48+vgap, 50, 20);

**this**.add(j5);

JTextField t5 = **new** JTextField();//不及格文本域

t5.setEditable(**false**);

t5.setBounds(825, 48+vgap, 40, 20);

//计算不及格人数

t5.setText(helper.getCount\_ScoreType(grade, classe, major\_Name, "不及格", courses.get(i)));

**this**.add(t5);

vgap += 30;

}

scores\_Ranking = **new** JButton("成绩排名");

scores\_Ranking.setBounds(420, 48+vgap+10, 100, 25);

//注册"成绩排名"按钮事件监听

scores\_Ranking.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

// **TODO** Auto-generated method stub

ScoreAnalyzeModel model = **new** ScoreAnalyzeModel(major\_Name, grade, classe);

ScoresRankingFrame frame = **new** ScoresRankingFrame(jd, "成绩排名", **true**, model);

}

});

**this**.add(scores\_Ranking);

**this**.setSize(900, 48+vgap+78);

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);

**this**.setVisible(**true**);

}

}

|  |
| --- |
| **ScoreAnalyzeOption.java** |

**package** studentsystem.frame;

**import** studentsystem.bean.Student;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.util.Tools;

**import** studentsystem.util.WindowUtil;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.HashMap;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JTextField;

//选择年级，院系，专业，班级进行成绩分析

**public** **class** ScoreAnalyzeOption **extends** JDialog{

**private** JButton confirm\_Button; //"确认"按钮。

**private** JButton cancel\_Button; //"取消"按钮。

**private** JComboBox major\_Box; //"专业"选项。

**private** JComboBox department\_Box; //"所属院系"选项。

**private** JLabel classe\_Label; //"班级"标签。

**private** JLabel grade\_Label; //"年级标签"。

**private** JLabel major\_Label; //"专业"标签。

**private** JLabel department\_Label; //"所属院系"标签。

**private** JTextField student\_IDText; //"学号"文本域。

**private** JTextField student\_NameText; //"姓名"文本域。

**private** JComboBox classe\_Box; //"班级"文本域。

**private** JComboBox grade\_Box; //"年级"文本域。

**private** JDialog jd; //当前窗口。

**private** HashMap<String, String> departments; //所有院系集合

**private** HashMap<String, String> all\_Major; //所有专业集合

**private** Vector<String> majors; //专业名称集合

**private** ManageHelper helper; //数据库业务处理对象

**private** Vector<String> classes; //班级集合

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** ScoreAnalyzeOption(JFrame owner, String title, **boolean** modal){

**super**(owner, title, modal);

helper = **new** ManageHelper(); //创建数据库业务处理对象

departments = **this**.helper.getAllDepartment(); //获得所有院系

all\_Major = **this**.helper.getAllMajor(); //获得所有的专业

**this**.jd = **this**;

**this**.setSize(350,300); //设置窗体大小。

**this**.setLayout(**null**); //设置空布局。

grade\_Label = **new** JLabel("年级:");

grade\_Label.setBounds(78, 48, 30, 20);

**this**.add(grade\_Label);

grade\_Box = **new** JComboBox<String>(Tools.CreateGrade()); //需要获得获得年级选项

grade\_Box.setBounds(116, 48, 150, 20);

//注册"年级选项"事件监听

grade\_Box.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

// **TODO** Auto-generated method stub

classe\_Box.removeAllItems();//移除"班级选项"的内容

String option = major\_Box.getSelectedItem().toString();

String major\_id = all\_Major.get(option); //专业编号

String grade = grade\_Box.getSelectedItem().toString();

**if**(!grade.equals("")){

classes = helper.getAllClasse(grade,major\_id); //获得班级

**for**(String s : classes){

classe\_Box.addItem(s);

}

}

}

});

**this**.add(grade\_Box);

department\_Label = **new** JLabel("院系:");

department\_Label.setBounds(78, 88, 30, 20);

**this**.add(department\_Label);

department\_Box = **new** JComboBox(departments.keySet().toArray());//获得键的集合

//注册"院系"选项事件监听

department\_Box.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

major\_Box.removeAllItems();//移除"专业选项"的内容

String option = department\_Box.getSelectedItem().toString();//获得选项名称

String department\_ID = departments.get(option); //获得院系编号

**if**(!department\_ID.equals("")){

majors = helper.getMajor(department\_ID); //获得专业

**for**(String s : majors){

major\_Box.addItem(s);

}

}

}

});

department\_Box.setBounds(116, 88, 150, 20);

**this**.add(department\_Box);

major\_Label = **new** JLabel("专业:");

major\_Label.setBounds(78, 136, 30, 20);

**this**.add(major\_Label);

major\_Box = **new** JComboBox(**new** String[]{""});//防止空指针异常

//注册"专业"选项事件监听

major\_Box.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

**if**(major\_Box.getSelectedItem()!=**null**){ //防止空指针

**if**(!major\_Box.getSelectedItem().toString().equals("")){

**if**(major\_Box.getSelectedItem().toString().equals("") || grade\_Box.getSelectedItem()==**null** || grade\_Box.getSelectedItem().toString().equals("")){

JOptionPane.showMessageDialog(jd, "年级不能为空", "", JOptionPane.WARNING\_MESSAGE);

major\_Box.setSelectedIndex(0); //设置为空选项

**return** ;

}

classe\_Box.removeAllItems();//移除"班级选项"的内容

String option = major\_Box.getSelectedItem().toString();

String major\_id = all\_Major.get(option); //专业编号

String grade = grade\_Box.getSelectedItem().toString();

**if**(!grade.equals("")){

classes = helper.getAllClasse(grade,major\_id); //获得班级

**for**(String s : classes){

classe\_Box.addItem(s);

}

}

}

}

}

});

major\_Box.setBounds(116, 136, 150, 20);

**this**.add(major\_Box);

classe\_Label = **new** JLabel("班级:"); //需要获得班级选项

classe\_Label.setBounds(78, 184, 30, 20);

**this**.add(classe\_Label);

classe\_Box = **new** JComboBox(**new** String[]{""});//防止空指针异常

classe\_Box.setBounds(116, 184, 150, 20);

**this**.add(classe\_Box);

confirm\_Button = **new** JButton("确定");

//注册"确认"按钮事件监听

confirm\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

String classe = classe\_Box.getSelectedItem().toString();

String grade = grade\_Box.getSelectedItem().toString();

String major\_ID = **null**;

String major\_Name = **null**;

//数据校验部分

**if**(grade.equals("")){

JOptionPane.showMessageDialog(jd, "年级不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(department\_Box.getSelectedItem()==**null**){ //先检查再用

JOptionPane.showMessageDialog(jd, "院系不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(department\_Box.getSelectedItem().toString().equals("")){

JOptionPane.showMessageDialog(jd, "院系不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(major\_Box.getSelectedItem()==**null**){ //先检查再用

JOptionPane.showMessageDialog(jd, "专业不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

major\_Name = major\_Box.getSelectedItem().toString();//获得专业名称

major\_ID = all\_Major.get(major\_Name); //获得专业编号

}

**if**(major\_ID.equals("")){

JOptionPane.showMessageDialog(jd, "专业不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(classe.equals("")){

JOptionPane.showMessageDialog(jd, "班级不能为空！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

jd.dispose();//关闭当前窗口

ScoreAnalyzeFrame analyzeFrame = **new** ScoreAnalyzeFrame(jd, major\_Name+"——"+classe+"班的成绩分析", **false**, major\_ID,major\_Name, grade,classe);

}

});

confirm\_Button.setBounds(70, 232, 60, 25);

**this**.add(confirm\_Button);

cancel\_Button = **new** JButton("取消");

cancel\_Button.setBounds(230, 232, 60, 25);

//注册"取消"按钮事件监听

cancel\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

jd.dispose();

}

});

**this**.add(cancel\_Button);

WindowUtil.setFrameCenter(**this**);

**this**.setResizable(**false**);

**this**.setVisible(**true**);

}

}

|  |
| --- |
| **ScoresRankingFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.model.ScoreAnalyzeModel;

**import** studentsystem.util.WindowUtil;

**import** javax.swing.JDialog;

**import** javax.swing.JScrollPane;

//成绩排名界面

**import** javax.swing.JTable;

**public** **class** ScoresRankingFrame **extends** JDialog{

**private** JTable jt;

**private** JScrollPane jsp;

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** ScoresRankingFrame(JDialog owner, String title, **boolean** modal,ScoreAnalyzeModel model){

**super**(owner, title, modal);

jt = **new** JTable();

jsp = **new** JScrollPane(jt);

jt.setModel(model);

jsp.setBounds(20, 20, 860, 460);

**this**.add(jsp);

**this**.setSize(1000,500);

WindowUtil.setFrameCenter(**this**);

**this**.setResizable(**false**);

**this**.setVisible(**true**);

}

}

**8.6.9 背景窗口**

|  |
| --- |
| **ImagePanel.java** |

**package** studentsystem.frame;

**import** java.awt.Graphics;

**import** java.awt.Image;

**import** javax.swing.ImageIcon;

**import** javax.swing.JPanel;

**public** **class** ImagePanel **extends** JPanel{

**private** ImageIcon icon;

**private** Image img;

**public** ImagePanel(){

icon=**new** ImageIcon(StudentSystemMainFrame.**class**.getResource("/1.png" ));

img=icon.getImage();

}

**public** **void** paintComponent(Graphics g)

{

**super**.paintComponent(g);

g.drawImage(img,0,0,**null** );

}

}

**8.6.10 个人管理窗口**

|  |
| --- |
| **StudentSystemLoginFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.bean.User;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.util.WindowUtil;

**import** java.awt.Color;

**import** java.awt.Container;

**import** java.awt.Image;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.IOException;

**import** javax.imageio.ImageIO;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPasswordField;

**import** javax.swing.JTextField;

//登陆界面

**public** **class** StudentSystemLoginFrame **extends** JFrame{

**private** JLabel username\_Label; //用户名标签。

**private** JLabel password\_Label; //密码标签。

**private** JTextField username\_Text; //用户名文本域。

**private** JPasswordField password\_Text; //密码文本域。

**private** JButton login\_Button; //登陆按钮。

**private** JButton register\_Button; //注册按钮。

**private** JFrame jf; //当前窗口 。

**public** StudentSystemLoginFrame(){

**super**("学生成绩管理系统登录界面");

**this**.jf = **this**;

**this**.setLayout(**null**);//设置为空布局。

**this**.setSize(400,300);//设置大小。

Container c = **this**.getContentPane();

c.setBackground(Color.WHITE);//设置背景颜色。

username\_Label = **new** JLabel("用户名:"); //创建"用户名"标签。

username\_Label.setBounds(100, 60, 50, 20); //设置"用户名"标签位置。

c.add(username\_Label); //添加"用户名"标签。

username\_Text = **new** JTextField(); //创建"用户名"文本域。

username\_Text.setBounds(160, 60, 120, 20); //设置"用户名"文本域位置。

username\_Text.grabFocus();//获得光标。

c.add(username\_Text); //添加"用户名"文本域。

password\_Label = **new** JLabel("密码:"); //创建"密码"标签。

password\_Label.setBounds(100, 140, 50, 20);

c.add(password\_Label); //添加"密码"标签。

password\_Text = **new** JPasswordField(); //创建"密码"文本域。

password\_Text.setBounds(160, 140, 120, 20); //设置"密码"文本域位置。

c.add(password\_Text); //添加"密码"文本域。

login\_Button = **new** JButton("登录"); //创建"登录"按钮。

login\_Button.setBounds(100, 210, 60, 20); //设置"登录"按钮位置。

//注册"登录"按钮事件监听。

login\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String username = username\_Text.getText().trim();

String password = **new** String(password\_Text.getPassword());

**if**(username.equals("")){

JOptionPane.showMessageDialog(jf, "用户名不能为空","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(password.equals("")){

JOptionPane.showMessageDialog(jf, "密码不能为空","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}

//登录业务处理。

User user = **new** User();//创建用户对象

user.setUsername(username);

user.setPassword(password);

ManageHelper helper = **new** ManageHelper();//创建数据库业务处理对象

**if**(helper.Login(user)){ //登陆业务处理

**if**(helper.Check\_IsLogin(user)){

JOptionPane.showMessageDialog(jf, "重复登陆！","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}**else**{

JOptionPane.showMessageDialog(jf, "登陆成功！");

jf.dispose();//关闭当前窗口。

//修改登陆情况

user.setIsLogin(1);//修改成为已经登陆。

helper.Update\_IsLogin(user);

//打开主界面

user.setPassword("");//重置密码

StudentSystemMainFrame frame = **new** StudentSystemMainFrame(user);

**return** ;

}

}**else**{

JOptionPane.showMessageDialog(jf, "用户名或密码错误！");

Reset(); //重置

**return** ;

}

}

});

c.add(login\_Button); //添加"登录"按钮 。

register\_Button = **new** JButton("注册"); //创建"注册"按钮。

register\_Button.setBounds(250, 210, 60, 20); //设置"注册"按钮位置。

//注册"注册"按钮事件监听。

register\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

jf.dispose(); //当前窗口关闭。

StudentSystemRegisterFrame studentSystemRegisterFrame = **new** StudentSystemRegisterFrame(); //打开注册界面

}

});

c.add(register\_Button); //添加"注册"按钮。

**this**.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

**this**.setResizable(**false**); //设置大小不可改变。

WindowUtil.setFrameCenter(**this**);//设置窗口居中。

**try** {

Image img = ImageIO.read(**this**.getClass().getResource("/2.png"));

**this**.setIconImage(img);

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

**this**.setVisible(**true**); //设置窗体可见。

}

//重置

**public** **void** Reset(){

username\_Text.setText("");

password\_Text.setText("");

}

}

|  |
| --- |
| **StudentSystemRegisterFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.bean.User;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.util.WindowUtil;

**import** java.awt.Color;

**import** java.awt.Container;

**import** java.awt.Image;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.io.IOException;

**import** javax.imageio.ImageIO;

**import** javax.swing.JButton;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPasswordField;

**import** javax.swing.JTextField;

//注册界面

**public** **class** StudentSystemRegisterFrame **extends** JFrame {

**private** JLabel username\_Label; //用户名标签。

**private** JLabel password\_Label; //密码标签。

**private** JLabel repassword\_Label; //确认密码标签。

**private** JTextField username\_Text; //用户名文本域。

**private** JPasswordField password\_Text; //密码文本域。

**private** JPasswordField repassword\_Text; //确认密码文本域。

**private** JButton register\_Button; //注册按钮。

**private** JButton cancel\_Button; //取消按钮。

**private** JFrame jf; //当前窗口 。

**public** StudentSystemRegisterFrame(){

**super**("学生成绩管理系统注册界面");

**this**.jf = **this**;

**this**.setLayout(**null**);//设置为空布局。

**this**.setSize(400,300);//设置大小。

Container c = **this**.getContentPane();

c.setBackground(Color.WHITE);//设置背景颜色。

username\_Label = **new** JLabel("用户名:"); //创建"用户名"标签。

username\_Label.setBounds(100, 60, 50, 20); //设置"用户名"标签位置。

c.add(username\_Label); //添加"用户名"标签。

username\_Text = **new** JTextField(); //创建"用户名"文本域。

username\_Text.setBounds(160, 60, 120, 20); //设置"用户名"文本域位置。

username\_Text.grabFocus();//获得光标。

c.add(username\_Text); //添加"用户名"文本域。

password\_Label = **new** JLabel("密码:"); //创建"密码"标签。

password\_Label.setBounds(100, 110, 50, 20);

c.add(password\_Label); //添加"密码"标签。

password\_Text = **new** JPasswordField(); //创建"密码"文本域。

password\_Text.setBounds(160, 110, 120, 20); //设置"密码"文本域位置。

c.add(password\_Text); //添加"密码"文本域。

repassword\_Label = **new** JLabel("确认密码"); //创建"确认密码"标签。

repassword\_Label.setBounds(100, 162, 70, 20); //设置"确认密码"标签位置。

c.add(repassword\_Label); //添加"确认密码"标签。

repassword\_Text = **new** JPasswordField(); //创建"确认密码"文本域。

repassword\_Text.setBounds(160, 162, 120, 20); //设置"确认密码"文本域位置。

c.add(repassword\_Text); //添加"确认密码"文本域。

register\_Button = **new** JButton("注册"); //创建"注册"按钮。

register\_Button.setBounds(90, 210, 60, 20); //设置"注册"按钮位置。

//注册"注册"按钮事件监听。

register\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

String username = username\_Text.getText().trim();//获得用户名。

String password = **new** String(password\_Text.getPassword()); //获得密码。

String repassword = **new** String(repassword\_Text.getPassword()); //获得确认密码。

**if**(username.equals("")){

JOptionPane.showMessageDialog(jf, "用户名不能为空！","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(password.equals("")){

JOptionPane.showMessageDialog(jf, "密码不能为空！","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(repassword.equals("")){

JOptionPane.showMessageDialog(jf, "确认密码不能为空！","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}

**if**(!password.equals(repassword)){

JOptionPane.showMessageDialog(jf, "两次密码不一致！","",JOptionPane.WARNING\_MESSAGE);

**return** ;

}

//注册业务处理。

User user = **new** User();//创建用户对象

user.setUsername(username);

user.setPassword(password);

ManageHelper helper = **new** ManageHelper();//创建数据库业务处理对象

**if**(helper.Register(user)){//注册处理

JOptionPane.showMessageDialog(jf, "注册成功！");

jf.dispose();//关闭当前窗口

StudentSystemLoginFrame frame = **new** StudentSystemLoginFrame();//返回登陆页面。

**return** ;

}**else**{

JOptionPane.showMessageDialog(jf, "注册失败！");

Reset(); //重置

**return** ;

}

}

});

c.add(register\_Button); //添加"注册"按钮。

cancel\_Button = **new** JButton("取消"); //创建"取消"按钮。

cancel\_Button.setBounds(250, 210, 60, 20); //设置"取消"按钮位置。

//注册"取消"按钮事件监听。

cancel\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

jf.dispose();//关闭当前页面。

StudentSystemLoginFrame studentSystemLoginFrame = **new** StudentSystemLoginFrame();//打开登陆页面。

}

});

c.add(cancel\_Button); //添加"取消"按钮。

**this**.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

**this**.setResizable(**false**); //设置大小不可改变。

WindowUtil.setFrameCenter(**this**);//设置窗口居中。

**try** {

Image img = ImageIO.read(**this**.getClass().getResource("/2.png"));

**this**.setIconImage(img);

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

**this**.setVisible(**true**); //设置窗体可见。

}

//重置

**public** **void** Reset(){

username\_Text.setText("");

password\_Text.setText("");

repassword\_Text.setText("");

}

}

|  |
| --- |
| **UpdatePasswordFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.CreateSql;

**import** studentsystem.util.WindowUtil;

**import** java.awt.BorderLayout;

**import** java.awt.Container;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

**public** **class** UpdateStudentScoreFrame **extends** JDialog{

**private** JPanel jp1,jp2,jp3; //面板。

**private** JLabel query\_Label; //标签。

**private** JButton query\_Button; //"查询"按钮。

**private** JComboBox query\_List; //"查询"选项。

**private** JButton preciseQuery\_Button; //"精确查询"按钮。

**private** JButton update\_Button; //"修改成绩"按钮。

**private** JTextField query\_Text; //"查询"文本域。

**private** JTable jt; //表格。

**private** JScrollPane jsp; //滚动条。

**private** JDialog jd; //当前窗口。

**private** StudentModel studentModel;//学生数据模型类

**private** **static** Vector<String> query\_Option;

**static** {

query\_Option = **new** Vector<String>();

query\_Option.add("全部");

query\_Option.add("学号");

query\_Option.add("班级");

query\_Option.add("性别");

query\_Option.add("班级");

query\_Option.add("年级");

query\_Option.add("专业");

query\_Option.add("院系");

}

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** UpdateStudentScoreFrame(JFrame owner, String title, **boolean** modal){

**super**(owner, title, modal);

**this**.jd = **this**;

Container c = **this**.getContentPane();

jp1 = **new** JPanel();

query\_Label = **new** JLabel("请输入查询信息:");

jp1.add(query\_Label);

query\_Text = **new** JTextField(10);

jp1.add(query\_Text);

query\_List = **new** JComboBox<String>(query\_Option);

jp1.add(query\_List);

query\_Button = **new** JButton("查询");

//注册"查询"按钮事件监听

query\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String str = query\_Text.getText().trim(); //查询内容

String option = query\_List.getSelectedItem().toString(); //查询选项

String sql = CreateSql.getStudent\_Sql(str, option); //获得sql语句

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);//更新Jtable

}

});

jp1.add(query\_Button);

preciseQuery\_Button = **new** JButton("多条件查询");

//注册"多条件查询"按钮事件监听

preciseQuery\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

ConditionsQueryFrame frame = **new** ConditionsQueryFrame(jd, "多条件查询", **true**, jt);

}

});

jp1.add(preciseQuery\_Button);

c.add(jp1,BorderLayout.NORTH); //添加面板

jp2 = **new** JPanel();

jt = **new** JTable();

String sql = CreateSql.getStudent\_Sql(**null**, "全部");//查询全部内容

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);

jsp = **new** JScrollPane(jt);

jp2.add(jsp);

c.add(jp2,BorderLayout.CENTER); //添加面板

jp3 = **new** JPanel();

update\_Button = **new** JButton("修改成绩");

//注册"详细信息"按钮事件监听

update\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**int** rowNum = jt.getSelectedRow();

**if**(rowNum==-1){

JOptionPane.showMessageDialog(jd, "请选择一行！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

UpdateScoreFrame frame = **new** UpdateScoreFrame(jd, "修改成绩", **true**, studentModel, rowNum);

}

});

jp3.add(update\_Button);

c.add(jp3,BorderLayout.SOUTH);

**this**.setSize(600,540);

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);//设置窗体居中。

**this**.setVisible(**true**);

}

}

**package** studentsystem.frame;

**import** studentsystem.model.StudentModel;

**import** studentsystem.util.CreateSql;

**import** studentsystem.util.WindowUtil;

**import** java.awt.BorderLayout;

**import** java.awt.Container;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.util.Vector;

**import** javax.swing.JButton;

**import** javax.swing.JComboBox;

**import** javax.swing.JDialog;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JOptionPane;

**import** javax.swing.JPanel;

**import** javax.swing.JScrollPane;

**import** javax.swing.JTable;

**import** javax.swing.JTextField;

**public** **class** UpdateStudentScoreFrame **extends** JDialog{

**private** JPanel jp1,jp2,jp3; //面板。

**private** JLabel query\_Label; //标签。

**private** JButton query\_Button; //"查询"按钮。

**private** JComboBox query\_List; //"查询"选项。

**private** JButton preciseQuery\_Button; //"精确查询"按钮。

**private** JButton update\_Button; //"修改成绩"按钮。

**private** JTextField query\_Text; //"查询"文本域。

**private** JTable jt; //表格。

**private** JScrollPane jsp; //滚动条。

**private** JDialog jd; //当前窗口。

**private** StudentModel studentModel;//学生数据模型类

**private** **static** Vector<String> query\_Option;

**static** {

query\_Option = **new** Vector<String>();

query\_Option.add("全部");

query\_Option.add("学号");

query\_Option.add("班级");

query\_Option.add("性别");

query\_Option.add("班级");

query\_Option.add("年级");

query\_Option.add("专业");

query\_Option.add("院系");

}

/\*\*

\*

\* **@param** owner 它的父窗口

\* **@param** title 窗口名

\* **@param** modal 指定的模式窗口，还有非模式窗口

\*/

**public** UpdateStudentScoreFrame(JFrame owner, String title, **boolean** modal){

**super**(owner, title, modal);

**this**.jd = **this**;

Container c = **this**.getContentPane();

jp1 = **new** JPanel();

query\_Label = **new** JLabel("请输入查询信息:");

jp1.add(query\_Label);

query\_Text = **new** JTextField(10);

jp1.add(query\_Text);

query\_List = **new** JComboBox<String>(query\_Option);

jp1.add(query\_List);

query\_Button = **new** JButton("查询");

//注册"查询"按钮事件监听

query\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

String str = query\_Text.getText().trim(); //查询内容

String option = query\_List.getSelectedItem().toString(); //查询选项

String sql = CreateSql.getStudent\_Sql(str, option); //获得sql语句

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);//更新Jtable

}

});

jp1.add(query\_Button);

preciseQuery\_Button = **new** JButton("多条件查询");

//注册"多条件查询"按钮事件监听

preciseQuery\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

ConditionsQueryFrame frame = **new** ConditionsQueryFrame(jd, "多条件查询", **true**, jt);

}

});

jp1.add(preciseQuery\_Button);

c.add(jp1,BorderLayout.NORTH); //添加面板

jp2 = **new** JPanel();

jt = **new** JTable();

String sql = CreateSql.getStudent\_Sql(**null**, "全部");//查询全部内容

studentModel = **new** StudentModel(sql,jd);//构建新的数据模型类，并更新

jt.setModel(studentModel);

jsp = **new** JScrollPane(jt);

jp2.add(jsp);

c.add(jp2,BorderLayout.CENTER); //添加面板

jp3 = **new** JPanel();

update\_Button = **new** JButton("修改成绩");

//注册"详细信息"按钮事件监听

update\_Button.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

**int** rowNum = jt.getSelectedRow();

**if**(rowNum==-1){

JOptionPane.showMessageDialog(jd, "请选择一行！", "", JOptionPane.WARNING\_MESSAGE);

**return** ;

}

UpdateScoreFrame frame = **new** UpdateScoreFrame(jd, "修改成绩", **true**, studentModel, rowNum);

}

});

jp3.add(update\_Button);

c.add(jp3,BorderLayout.SOUTH);

**this**.setSize(600,540);

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);//设置窗体居中。

**this**.setVisible(**true**);

}

}

**8.6.11 主窗口**

|  |
| --- |
| **StudentSystemMainFrame.java** |

**package** studentsystem.frame;

**import** studentsystem.bean.User;

**import** studentsystem.dao.ManageHelper;

**import** studentsystem.util.WindowUtil;

**import** java.awt.BorderLayout;

**import** java.awt.Container;

**import** java.awt.Graphics;

**import** java.awt.Graphics2D;

**import** java.awt.Image;

**import** java.awt.Toolkit;

**import** java.awt.event.ActionEvent;

**import** java.awt.event.ActionListener;

**import** java.awt.event.WindowEvent;

**import** java.awt.event.WindowListener;

**import** java.io.IOException;

**import** javax.imageio.ImageIO;

**import** javax.swing.JFrame;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenuItem;

**import** javax.swing.JPanel;

**import** javax.swing.JTextField;

//主界面

**public** **class** StudentSystemMainFrame **extends** JFrame {

**private** JMenuBar menuBar; //应用菜单条。

**private** JMenu student\_Management; //"学生管理"菜单。

**private** JMenu score\_Management; //"成绩管理"菜单。

**private** JMenu personal\_Management; //"个人管理"菜单。

**private** JMenu help\_File; //帮助文档菜单

**private** JMenuItem add\_Student; //"添加学生"菜单项。

**private** JMenuItem query\_Student; //"查询学生信息"菜单项。

**private** JMenuItem modify\_Student; //"修改学生信息"菜单项。

**private** JMenuItem delete\_Student; //"删除学生"菜单项。

**private** JMenuItem add\_Score; //"添加学生成绩"菜单项。

**private** JMenuItem modify\_Score; //"修改学生成绩"菜单项。

**private** JMenuItem query\_Score; //"成绩查询"菜单项。

**private** JMenuItem score\_Statistics; //"成绩统计"菜单项。

**private** JMenuItem change\_Password; //"修改密码"菜单项。

**private** JMenuItem logout; //"退出登录"菜单项。

**private** JTextField help\_Blog; //帮助博客

**private** JFrame jf; //当前窗口。

**private** User user;//当前用户。

**public** StudentSystemMainFrame(User user){

**super**("学生成绩管理系统,欢迎你"+user.getUsername());

**this**.user = user;

**this**.jf = **this**;

menuBar = **new** JMenuBar(); //创建菜单条。

**this**.setJMenuBar(menuBar); //添加菜单条。

student\_Management = **new** JMenu("学生管理"); //创建"学生管理"菜单。

menuBar.add(student\_Management); //添加"学生管理"菜单。

add\_Student = **new** JMenuItem("添加学生"); //创建"添加学生"菜单项。

//注册"添加学生"菜单项事件监听

add\_Student.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

AddStudentFrame addStudentFrame = **new** AddStudentFrame(jf,"添加学生",**true**);

}

});

student\_Management.add(add\_Student); //添加"添加学生"菜单项。

query\_Student = **new** JMenuItem("查询学生信息"); //创建"查询学生"菜单项。

//注册"查询学生"菜单项事件监听。

query\_Student.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

QueryStudentFrame queryStudentFrame = **new** QueryStudentFrame(jf, "查询学生信息", **true**);

}

});

student\_Management.add(query\_Student); //添加"查询学生信息"菜单项。

modify\_Student = **new** JMenuItem("修改学生信息"); //创建"修改学生信息"菜单项。

//注册"修改学生信息"菜单项事件监听

modify\_Student.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

ModifyStudentFrame modifyStudentFrame = **new** ModifyStudentFrame(jf, "修改学生信息", **true**);

}

});

student\_Management.add(modify\_Student); //添加"修改学生"菜单项。

delete\_Student = **new** JMenuItem("删除学生"); //创建"删除学生"菜单项。

//注册"删除学生"按钮事件监听

delete\_Student.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

DeleteStudentFrame deleteStudentFrame = **new** DeleteStudentFrame(jf, "删除学生", **true**);

}

});

student\_Management.add(delete\_Student); //添加"删除学生"菜单项.

score\_Management = **new** JMenu("成绩管理"); //创建"成绩管理"菜单。

menuBar.add(score\_Management); //添加"成绩管理"菜单。

add\_Score = **new** JMenuItem("添加学生成绩"); //创建"添加学生成绩"菜单项。

//注册"添加学生成绩"菜单项事件监听

add\_Score.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent arg0) {

AddStudentScoreFrame frame = **new** AddStudentScoreFrame(jf, "添加学生成绩", **true**);

}

});

score\_Management.add(add\_Score); //添加"添加学生成绩"菜单项。

modify\_Score = **new** JMenuItem("修改学生成绩"); //创建"修改学生成绩"菜单项。

//注册"修改学生成绩"菜单项事件监听

modify\_Score.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

UpdateStudentScoreFrame frame = **new** UpdateStudentScoreFrame(jf, "修改学生成绩", **true**);

}

});

score\_Management.add(modify\_Score); //添加"修改学生成绩"菜单项。

query\_Score = **new** JMenuItem("成绩查询"); //创建"成绩查询"菜单项。

//注册"成绩查询"菜单项事件监听

query\_Score.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

QueryStudentScoreFrame frame = **new** QueryStudentScoreFrame(jf, "查询学生成绩", **true**);

}

});

score\_Management.add(query\_Score); //添加"成绩查询"菜单项。

score\_Statistics = **new** JMenuItem("成绩统计"); //创建"成绩统计"菜单项。

score\_Statistics.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

// **TODO** Auto-generated method stub

ScoreAnalyzeOption analyzeOption = **new** ScoreAnalyzeOption(jf, "成绩分析选项", **true**);

}

});

score\_Management.add(score\_Statistics); //添加"成绩统计"菜单项。

personal\_Management = **new** JMenu("个人管理"); //创建"个人管理"菜单。

menuBar.add(personal\_Management); //添加"个人管理"菜单。

change\_Password = **new** JMenuItem("修改密码"); //创建"修改密码"菜单项。

//注册"修改密码"菜单项事件监听

change\_Password.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

UpdatePasswordFrame frame = **new** UpdatePasswordFrame(jf, "修改密码", **true**,user);

}

});

personal\_Management.add(change\_Password); //添加"修改密码"菜单项。

logout = **new** JMenuItem("退出登录"); //创建"退出登录"菜单项。

//注册"退出登录"菜单项时间监听

logout.addActionListener(**new** ActionListener() {

@Override

**public** **void** actionPerformed(ActionEvent e) {

jf.dispose(); //关闭当前窗口

//修改登陆状态

ManageHelper helper = **new** ManageHelper();

user.setIsLogin(0);//设置登陆状态为未登录。

helper.Update\_IsLogin(user);

StudentSystemLoginFrame frame = **new** StudentSystemLoginFrame(); //打开登陆界面

}

});

personal\_Management.add(logout); //添加"退出登录"菜单项。

// 添加帮助文档

help\_File = **new** JMenu("帮助");

menuBar.add(help\_File);

help\_Blog = **new** JTextField("https://xiaoyi-s.github.io/");

help\_File.add(help\_Blog);

**this**.setSize(578, 400);

**this**.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

**this**.setResizable(**false**);

WindowUtil.setFrameCenter(**this**);//设置窗体居中。

ImagePanel imagePanel = **new** ImagePanel();

setContentPane(imagePanel);

**try** {

Image img = ImageIO.read(**this**.getClass().getResource("/2.png"));

**this**.setIconImage(img);

} **catch** (IOException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}

**this**.setVisible(**true**);//设置窗体可见。

**this**.addWindowListener(**new** WindowListener() {

@Override

**public** **void** windowOpened(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowIconified(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowDeiconified(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowDeactivated(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowClosing(WindowEvent arg0) {

// **TODO** Auto-generated method stub

//修改登陆状态

ManageHelper helper = **new** ManageHelper();

user.setIsLogin(0);//设置登陆状态为未登录。

helper.Update\_IsLogin(user);

}

@Override

**public** **void** windowClosed(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** windowActivated(WindowEvent arg0) {

// **TODO** Auto-generated method stub

}

});

}

}

**8.7 主程序代码**

|  |
| --- |
| **TestLogin.java** |

**package** studentsystem.run;

**import** studentsystem.frame.StudentSystemLoginFrame;

**public** **class** TestLogin {

**public** **static** **void** main(String[] args) {

StudentSystemLoginFrame frame = **new** StudentSystemLoginFrame();

}

}