**实训报告**

JAVA程序设计

**题 目： 学生成绩管理系统**

**院 系： 信息工程学院**

**专 业： 软件工程**

**姓 名： 胡胜辉**

**学 号： 1951300128**

**指导教师： 邱勋拥**

**日 期： 2021年10月1日**

**桂林电子科技大学信息科技学院**

目录

[1绪论 3](#_Toc84235666)

[1.1课题背景及意义 3](#_Toc84235667)

[1.2国内外发展现状分析 3](#_Toc84235668)

[1.2.1现状 3](#_Toc84235669)

[1.2.2问题分析 4](#_Toc84235670)

[2系统分析 4](#_Toc84235671)

[2.1可行性研究分析 4](#_Toc84235672)

[2.1.1技术可行性 4](#_Toc84235673)

[2.1.2操做可行性 5](#_Toc84235674)

[2.2用户及系统需求说明 5](#_Toc84235675)

[2.2.1系统功能需求 5](#_Toc84235676)

[2.2.2用户权限说明 6](#_Toc84235677)

[2.3数据流图 6](#_Toc84235678)

[2.3.1顶层数据流图 6](#_Toc84235679)

[2.3.2第一层数据流图 7](#_Toc84235680)

[2.4用例图 7](#_Toc84235681)

[2.4.1角色用例图 7](#_Toc84235682)

[3总体设计 8](#_Toc84235683)

[3.1系统功能模块设计目标 8](#_Toc84235684)

[3.2数据库设计 10](#_Toc84235685)

[3.2.1逻辑结构设计 10](#_Toc84235686)

[3.2.2物理结构设计 10](#_Toc84235687)

[4系统实现 14](#_Toc84235688)

[4.1系统功能模块实现 14](#_Toc84235689)

[4.1.1用户登录模块设计 14](#_Toc84235690)

[4.1.2管理员-个人主页模块设计 17](#_Toc84235691)

[4.1.3管理员-分配课程-教师模块设计 18](#_Toc84235692)

[4.1.4管理员-分配课程（教师）-查询已分配模块设计 19](#_Toc84235693)

[4.1.5管理员-分配课程（教师）-未分配教师课程模块设计： 21](#_Toc84235694)

[4.1.6管理员-分配课程（学生）-已分配模块设计： 22](#_Toc84235695)

[4.1.6管理员-分配课程（学生）-未分配模块设计： 24](#_Toc84235696)

[4.1.7管理员-分配（教师）-提交模块设计： 25](#_Toc84235697)

[4.1.8管理员-分配（学生）-提交模块设计： 26](#_Toc84235698)

[4.1.9教师-个人主页模块设计： 27](#_Toc84235699)

[4.1.10教师-已录入模块： 29](#_Toc84235700)

[4.1.11教师－未录入模块设计： 30](#_Toc84235701)

[4.1.12教师提交模块设计： 31](#_Toc84235702)

[4.1.13学生主页模块设计： 33](#_Toc84235703)

[4.1.14学生查询成绩模块： 33](#_Toc84235704)

[5测试 36](#_Toc84235705)

[5.1登录界面测试 36](#_Toc84235706)

[5.2管理员－个人主页模块测试 39](#_Toc84235707)

[5.3管理员分配（教师）模块测试 40](#_Toc84235708)

[5.4管理员已分配教师模块测试 41](#_Toc84235709)

[5.5管理员未分配教师模块测试 42](#_Toc84235710)

[5.6管理员提交分配模块测试 43](#_Toc84235711)

[5.7管理员－分配（学生）模块测试 45](#_Toc84235712)

[5.8管理员－学生未分配模块测试 46](#_Toc84235713)

[5.9管理员提交选课模块测试 47](#_Toc84235714)

[5.10管理员－学生已分配模块测试 49](#_Toc84235715)

[5.11教师个人主页模块测试 50](#_Toc84235716)

[5.12教师-开始录入成绩查询模块 51](#_Toc84235717)

[5.13教师-提交成绩查询模块 52](#_Toc84235718)

[5.14学生个人主页模块测试 54](#_Toc84235719)

[5.15学生-查询成绩模块测试 55](#_Toc84235720)

[5.16整体测试 56](#_Toc84235721)

[6总结 58](#_Toc84235722)

[7参考文献 58](#_Toc84235723)

# 1绪论

## 1.1课题背景及意义

在现如今的时代，数字化的管理系统早就已经替代了原本的纸质化系统，使得对信息的管理更加便捷和精确。提高了管理效率和节省了较多的人力。一块小小的硬盘就能将原本占地一个较大的档案室的信息全部存储起来，极大的提升了信息的安全性和信息的保存周期。

使用计算机软件管理系统来管理信息，能充分发挥它的优越性，将复杂繁琐且重复的管理步骤交给计算机，使用人员只需要简简单单的点击一下按钮就能进行相应的操作，降低了使用人员的学习成本。

## 1.2国内外发展现状分析

### 1.2.1现状

在国内外的全部学校中都采用的计算机学生成绩管理系统来进行学生的信息管理，一个简洁，功能齐全的管理系统就显得尤为的重要，能快速地帮助行政人员管理教师、学生的各种信息。也让教师能清晰地知道自己的任课信息（时间、地点、班级等），以及极大地提高录入成绩地效率。也让学生能够随时查询自己的上课信息（时间、地点、任课老师等）、以及随时查询自己的考核成绩。

### 1.2.2问题分析

学生成绩管理系统顾名思义，就是对学生的成绩进行管理，但是要对得到学生的成绩进行管理就需要管理员先将课程分配给教师、再分配给相应专业的学生，通过考核，然后由教师录入相应的成绩，最后学生才能查询到相应的课程考核的成绩。

管理系统应该具有以下优点例：检索迅速、查找方便、可靠性高、存储量大、保密性好、寿命长、成本低等。能够为用户提供充足的信息和快捷的查询手段，简洁易上手的界面，节约使用人员的学习成本。

这些优点能够极大地提高管理者管理的效率,也是学校走向科学化、正规化管理,与世界接轨的重要条件。 因此，开发设计这样一套学生成绩管理软件成为很有必要的事情。其次，初步掌握软件开发的流程，为以后的工作积累一定的开发经验。

# 2系统分析

## 2.1可行性研究分析

### 2.1.1技术可行性

本次实训所需要开发的学生成绩管理系统，功能简单，界面也简洁明了，没有很复杂的方面。使用Java面向对象编程语言就可以开发出相关的系统部分。因为需要长期保存相应的课程信息、选课信息、成绩信息等。选用MySQL数据库来进行数据的管理，但是MySQL数据库为纯英文界面，且使用的效果没有那么出色。所以使用数据库管理工具Navicat Premium来对MySQl数据库进行管理，简化数据库的各种操作，例如数据库的创建，table表的创建、查询、修改、删除、增加，创建数据库表之间的各种约束条件。保证表与表之间互相制约的关系和对照关系。

本次管理系统采用IDEA开发软件，可以自动填充代码，构建模块等，更大限度的节约开发时间和成本。

现有的Windows10 、IDEA2021.2.1 、Java JDK15.0.2 、MySQL 8.0 、Navicat Premium 15等软件版本互相兼容，在软件开发技术上不存在任何的问题，足以支持学生管理系统的开发。

### 2.1.2操做可行性

1、本系统界面简洁明了、功能一目了然。

2、本系统操作方便，0基础用户可以快速上手。

## 2.2用户及系统需求说明

### 2.2.1系统功能需求

学生成绩管理系统可以来管理规模较大的成绩信息、对学生的成绩信息进行管理。系统的目标用户为管理员 、教师 、学生 三类人员。使用系统可以极大的减少用户的工作量，为用户节约更多的时间。系统为用户提供了简单又实用的功能。

系统应该提供给管理员对教师任教课程的分配信息的管理、对学生选课信息的管理等功能。提供给教师对学生成绩信息的管理功能。提供给学生查询成绩的功能。

### 2.2.2用户权限说明

管理系统分为管理员 、教师 、学生 三类用户，每种用户有各自相同、不同的操作权限。

管理员作为系统的最高权限者，拥有帮学生选课 并修改、为教师分配课程并修改 等功能。

教师可以查询任课的课程信息 、查询并录入选修了自己任课课程的学生的考核成绩。

学生只能进入系统查询对应学期、相应的课程考核成绩。

## 2.3数据流图

### 2.3.1顶层数据流图

图示

描述已自动生成



### 2.3.2第一层数据流图



## 2.4用例图

### 2.4.1角色用例图

图示

描述已自动生成

# 3总体设计

## 3.1系统功能模块设计目标

用户登录信息模块：此模块用于在登录界面验证用户的用户名是否存在和密码是否正确，根据相应的数据库用户类型进入相应的个人主界面。

管理员-个人主页：此模块用于显示管理员的信息（包括用户名和姓名），以及分配课程-教师和分配课程-学生按钮；

管理员-教师任课信息模块：此模块用于查询数据库中已经分配了教师的课程，并且将数据处理返回为table表。

管理员-未分配教师的课程模块：此模块用于查询数据库未分配了教师的课程，并且将数据处理返回为table表。

管理员-提交任课模块：此模块用于提交table表中修改的任课信息并返回提交结果

管理员-学生选课信息模块：此模块用于查询数据库中已经分配了学生的课程，并且将数据处理返回给table表。

管理员-未分配学生的课程模块：此模块用于查询数据库未分配了学生的课程，并且将数据处理 返回为table表。

管理员-提交选课模块：此模块用于提交table表中修改的选课信息并返回提交结果

教师-个人主页：此模块用于显示教师的信息（包括用户名和姓名），以及查询成绩、录入成绩按钮；

教师-已经录入成绩的显示模块：此模块用于查询数据库中已经录入成绩的学生信息，并且将数据处理 返回为table表。

教师-提交成绩模块：此模块用于提交table表中修改的成绩信息到数据库并返回提交结果

学生-个人主页：此模块用于显示学生的信息（包括学好、姓名），以及查询按钮。

学生-查询模块：此模块用于查询数据库中对应学期的课程考核成绩，并返回查询结果到table表中。

## 3.2数据库设计

### 3.2.1逻辑结构设计

学生成绩管理系统应该包含用户登录验证、任课信息查询和修改、选课信息的查询和修改、成绩录入、成绩查询等对数据库操作的功能。

包含user（登录）表、教师表、管理员表、学生表、课程表、选课表。

User（用户名，密码，类型）；

教师表（工号、姓名）；

管理员表（教师id、姓名）；

学生表（学号、姓名）；

课程表（课程号，课程名，学分，教师id）；

选课表（课程号，学号，学期，成绩）；

### 3.2.2物理结构设计

User表：

图形用户界面, 应用程序, 电子邮件

描述已自动生成

图形用户界面, 表格

描述已自动生成

管理员表：

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 应用程序, Word

描述已自动生成

教师表：

图形用户界面, 应用程序, Excel

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

课程表：

图形用户界面, 应用程序

描述已自动生成图形用户界面, 应用程序

描述已自动生成

选课表：

图形用户界面, 应用程序

描述已自动生成图形用户界面, 应用程序

描述已自动生成

学生表：

图形用户界面, 应用程序

描述已自动生成图形用户界面, 应用程序, Word

描述已自动生成

# 4系统实现

## 4.1系统功能模块实现

### 4.1.1用户登录模块设计

主界面包含提示信息（用户名、密码）以及2个文本输入框，登录按钮和退出按钮。如果用户没有输入用户名和密码就提交，提示用户 请输入用户名和密码。用户输入了用户名和密码后，将输入的用户名和密码封装到user实体中。系统将会使用user对象的用户名和密码进入数据库查询。如果在数据库中未找到输入的用户名提示用户该用户名不存在。如果找到了用户名，倒是密码错误，提示用户密码错误。当用户名和密码与数据库的user表中的数据匹配时，数据库返回相应的用户类型。系统根据返回的类型数据进入switch语句判断并打开相应的个人主页，至此用户才算真正的登录成功。

图形用户界面

描述已自动生成

核心代码：

LoginButton.addActionListener(e -> {  
 User user = new User();  
 user.setUid(uname.getText());  
 user.setUpass(upass.getText());  
 if (user.getUid().equals("") || user.getUpass().equals(""))  
 {  
 JOptionPane.showMessageDialog(null, "请输入用户名或密码");  
 }  
 else{  
 SearchUser search = new SearchUser(user);  
 int result = search.S();  
 System.out.println("身份类型："+result);  
 switch (result){  
 case 1: new Manager(user); frame.dispose();break;  
 case 2: new Teacher(user); frame.dispose();break;  
 case 3: new Student(user); frame.dispose();break;  
 case 10:JOptionPane.showMessageDialog(null,"用户名不存在");break;  
 case 11:JOptionPane.showMessageDialog(null,"密码错误");break;  
 }  
 }  
})**;**

User实体类代码：

package com.sx.User**;**public class User {  
 private String uid**;** private String upass**;** public User(String uid**,** String upass){  
 this.uid = uid**;** this.upass=upass**;** }  
  
 public User() {  
  
 }  
  
 public String getUid() {  
 return uid**;** }  
  
 public void setUid(String uid) {  
 this.uid = uid**;** }  
  
 public String getUpass() {  
 return upass**;** }  
  
 public void setUpass(String upass) {  
 this.upass = upass**;** }}

### 4.1.2管理员-个人主页模块设计

包含管理员的个人信息，以及 分配课程-教师 按钮和 分配课程-学生 按钮，管理员可以根据目前的需求来选择点击相应的按钮来选择是为学生选课还是为教师分配课程。

图形用户界面

描述已自动生成

核心代码：

//---- mno ----  
 mno.setText(Mno)**;** mno.setFont(mno.getFont().deriveFont(mno.getFont().getSize() + **6f**))**;** mno.setHorizontalAlignment(SwingConstants.*CENTER*)**;** contentPane.add(mno)**;** mno.setBounds(**155, 10, 135, 50**)**;** //---- mno2 ----  
 mno2.setText(bundle.getString("mno2.text"))**;** mno2.setFont(mno2.getFont().deriveFont(mno2.getFont().getSize() + **6f**))**;** mno2.setHorizontalAlignment(SwingConstants.*CENTER*)**;** contentPane.add(mno2)**;** mno2.setBounds(**15, 65, 135, 50**)**;** //---- mname ----  
 mname.setText(Mname)**;** mname.setFont(mname.getFont().deriveFont(mname.getFont().getSize() + **6f**))**;** mname.setHorizontalAlignment(SwingConstants.*CENTER*)**;** contentPane.add(mname)**;** mname.setBounds(**155, 65, 135, 50**)**;** //---- button1 ----  
 button1.setText(bundle.getString("button1.text\_2"))**;** button1.setFont(button1.getFont().deriveFont(button1.getFont().getSize() + **5f**))**;** button1.addActionListener(e -> {  
 button1ActionPerformed(e)**;** })**;** contentPane.add(button1)**;** button1.setBounds(**55, 195, 165, 55**)**;** //---- button2 ----  
 button2.setText(bundle.getString("button2.text\_2"))**;** button2.setFont(button2.getFont().deriveFont(button2.getFont().getSize() + **5f**))**;** button2.addActionListener(e -> {  
 button2ActionPerformed(e)**;**})**;**

### 4.1.3管理员-分配课程-教师模块设计

管理员根据当前需求查询教师的任课表或者查询未分配教师的课程表。

上部分为动态的table表，可以根据点击按钮的不同，触发不同的监听事件，重新赋值table来达到刷新课表的效果。

下半部分为固定的表格，只显示目前在职的教师的个人信息（包括工号、姓名、部门），用于点击未分配按钮时的输入信息对照表，若用户输入的教师工号不在教师表中则提示 提交失败，请输入下表中的教师工号。

表格

描述已自动生成

### 4.1.4管理员-分配课程（教师）-查询已分配模块设计

通过SQL语句来进行数据库的查询，并将结果返回给Object[][]数组，重新赋值table1表达到显示效果

核心代码：

//4、定义sql语句  
 String sql = "

SELECT DISTINCT 学期,选课表.课程号,课程名,学分,教师表.教师id FROM 选课表,课程表,教师表

WHERE 选课表.课程号=课程表.课程号

and 课程表.教师id=教师表.教师id

order by 课程表.课程号,学期 desc"**;**

//5、获取执行sql的对象、Statement  
 statement = connection.createStatement()**;** ResultSet rs = statement.executeQuery(sql)**;**

while (rs.next()){  
 i1++**;** }System.*out*.println("行数:"+i1)**;** date3 = new Object[i1][]**;** rs = statement.executeQuery(sql)**;** while (rs.next()) {  
 date3[i2] = new Object[]{  
 rs.getString("学期")**,** rs.getString("课程号")**,** rs.getString("课程名")**,** rs.getDouble("学分")**,** rs.getString("教师id")  
 }**;** System.out.println(Arrays.toString(date3[i2]));  
 i2++;  
  
 }  
  
} catch (SQLException e) {  
 e.printStackTrace();  
} finally {  
 //7、关闭资源  
 if (statement != null) {  
 try {  
 statement.close();  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
 if (connection != null) {  
 try {  
 connection.close();  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
}  
return date3;

刷新table代码：

SelectManager\_Teacher\_all selectManager\_teacher\_all = new SelectManager\_Teacher\_all()**;**date3 = selectManager\_teacher\_all.S()**;**

//刷新table

table1.setModel(newDefaultTableModel(  
 date3**,** newString[]{学期"**,**"\u8bfe\u7a0b\u53f7"

**,**"\u8bfe\u7a0b\u540d"**,**"\u5b66\u5206"**,**"\u4efb\u8bfe\u8001\u5e08id"

}  
))**;**

### 4.1.5管理员-分配课程（教师）-未分配教师课程模块设计：

用于查询数据库中课程表的教师id字段为NULL的课程并返回Object[][]数组刷新并显示在上半部table1表中

核心代码：

//4、定义sql语句  
 String sql = "SELECT \* FROM 课程表 WHERE 教师id is null "**;** //5、获取执行sql的对象、Statement  
 statement = connection.createStatement()**;** ResultSet rs = statement.executeQuery(sql)**;** int i1= **0;** while (rs.next()){  
 i1++**;** }System.*out*.println("行数:"+i1)**;** obj1 = new Object[i1][]**;** rs = statement.executeQuery(sql)**;** int i2=**0;** while (rs.next()) {  
 obj1[i2] = new Object[]{  
 rs.getString("课程号")**,** rs.getString("课程名")**,** rs.getDouble("学分")**,** null  
 }**;** System.*out*.println(Arrays.*toString*(obj1[i2]))**;** i2++**;** }  
  
} catch (SQLException e) {  
 e.printStackTrace()**;**} finally {  
 //7、关闭资源  
 if (statement != null) {  
 try {  
 statement.close()**;** } catch (SQLException throwables) {  
 throwables.printStackTrace()**;** }  
 }  
 if (connection != null) {  
 try {  
 connection.close()**;** } catch (SQLException throwables) {  
 throwables.printStackTrace()**;** }  
 }  
}  
return obj1**;**

刷新代码：

SelectManager\_Teacher selectManager\_teacher = new SelectManager\_Teacher()**;**date1 = selectManager\_teacher.SelectManger\_Student()**;**//刷新table  
table1.setModel(new DefaultTableModel(  
 date1**,** new String[]{  
 "\u8bfe\u7a0b\u53f7"**,** "\u8bfe\u7a0b\u540d"**,** "\u5b66\u5206"**,**"\u4efb\u8bfe\u8001\u5e08id"  
 }  
))**;**

### 4.1.6管理员-分配课程（学生）-已分配模块设计：

大致与分配课程（教师）类似，通过不同按钮触发不同的监听事件，来分别查询数据库中 选课表中的选课数据，将课程号和学号返回给显示模块中调用方法的对象，赋值给数组。使用数组赋值给table1来达到刷新table1表动态显示不同数据

核心代码：

//4、定义sql语句  
 String sql = "SELECT DISTINCT 学期,选课表.课程号,课程名,学分,教师姓名,选课表.学生id FROM 选课表,课程表,教师表 WHERE 选课表.课程号=课程表.课程号 and 课程表.教师id=教师表.教师id and 学期='202101' and 选课表.学生id is not null order by 课程表.课程号";  
 //5、获取执行sql的对象、Statement  
 statement = connection.createStatement();  
 ResultSet rs = statement.executeQuery(sql);  
 while (rs.next()){  
 i1++;  
 }System.out.println("行数:"+i1);  
 obj1 = new Object[i1][];  
 rs = null;  
 rs = statement.executeQuery(sql);  
 while (rs.next()) {  
 obj1[i2] = new Object[]{  
 rs.getString("学期"),  
 rs.getString("课程号"),  
 rs.getString("课程名"),  
 rs.getDouble("学分"),  
 rs.getString("教师姓名"),  
 rs.getString("选课表.学生id")  
 };  
 System.out.println(Arrays.toString(obj1[i2]));  
 i2++;  
 }  
} catch (SQLException e) {  
 e.printStackTrace();  
} finally {  
 //7、关闭资源  
 if (statement != null) {  
 try {  
 statement.close();  
 } catch (SQLException throwables) {  
 throwables.printStackTrace();  
 }  
 }  
 if (connection != null) {  
 try {  
 connection.close();  
 } catch (SQLException throwables) {  
 throwables.printStackTrace()**;** }  
 }  
}return obj1**;**

### 4.1.6管理员-分配课程（学生）-未分配模块设计：

大致与分配课程（教师）-未分配模块类似，通过不同按钮触发不同的监听事件，来分别查询数据库中 课程表中的课程数据，将信息返回给显示模块中调用方法的对象，赋值给数组。使用数组赋值给table1来达到刷新table1表动态显示不同数据

//4、定义sql语句  
 String sql = "SELECT distinct 学期,课程表.课程号,课程名,学分,教师姓名 FROM 选课表,课程表,教师表 WHERE 课程表.课程号=选课表.课程号 and 课程表.教师id=教师表.教师id AND 学期='202101' order by 课程表.课程号";//5、获取执行sql的对象、Statement  
 statement = connection.createStatement()**;** ResultSet rs = statement.executeQuery(sql)**;** while (rs.next()){  
 i1++**;** }System.*out*.println("行数:"+i1)**;** obj1 = new Object[i1][]**;** rs = statement.executeQuery(sql)**;** int i2=**0;** while (rs.next()) {  
 obj1[i2] = new Object[]{  
 rs.getString("学期")**,** rs.getString("课程号")**,** rs.getString("课程名")**,** rs.getDouble("学分")**,** rs.getString("教师姓名")**,** null  
 }**;** System.*out*.println(Arrays.*toString*(obj1[i2]))**;** i2++**;** }  
} catch (SQLException e) {  
 e.printStackTrace()**;**} finally {  
 //7、关闭资源  
 if (statement != null) {  
 try {  
 statement.close()**;** } catch (SQLException throwables) {  
 throwables.printStackTrace()**;** }  
 }  
 if (connection != null) {  
 try {  
 connection.close()**;** } catch (SQLException throwables) {  
 throwables.printStackTrace()**;** }  
 }  
}  
return obj1**;**

### 4.1.7管理员-分配（教师）-提交模块设计：

点击提交按钮后触发监听事件，利用table中的getValueAt（”行数”,”列数”）方法循环获取相应行列的表格值并保存到Object二维数组中并实例化update类对象，调用update类对象进行数据库的信息更新。

调用对象的方法后调用监听事件方法刷新界面的数据，达到不再重复操作某一数据的目的

核心代码：

//获取table的数据  
grade = new Object[table1.getRowCount()][]**;**for (int i = **0;**i<table1.getRowCount()**;**i++){  
 grade[i]=new Object[]{  
 table1.getValueAt(i**,0**)**,** table1.getValueAt(i**,3**)  
 }**;** System.*out*.println("i:"+i)**;** System.*out*.println(Arrays.*toString*(grade[i]))**;**}  
if (grade[**0**][**1**]==null){  
 JOptionPane.*showMessageDialog*(null**,**"请输入选课方案")**;**}  
else{  
 Update\_Manager\_Teacher manager\_teach = new Update\_Manager\_Teacher(grade)**;** manager\_teach.S()**;**}  
button1ActionPerformed()**;**

Update\_manager\_teacher类核心代码：

public void S() {  
 Connection connection = null;  
 Statement statement = null;  
 String sql;  
 int a=0;  
 try {  
 connection = DBManager.getConn();  
 statement = connection.createStatement();  
 //分配课程-学生  
 for (int i= 0;i<date3.length;i++) {  
 sql = "update 课程表 set 教师id=' "+date3[i][1]+"' WHERE 课程号='"+date3[i][0]+"'";  
 int result = statement.executeUpdate(sql);  
 System.out.println(result);  
 a++;  
 }  
 JOptionPane.showMessageDialog(null,"分配方案 提交成功");  
 } catch (Exception e) {  
 if (a!=0){  
 JOptionPane.showMessageDialog(null,"成功提交"+a+"条数据");  
 }  
 JOptionPane.showMessageDialog(null,"更新后续任课信息出现问题\n请输入下表中的教师工号");  
 }  
}

### 4.1.8管理员-分配（学生）-提交模块设计：

点击提交按钮后触发监听事件，利用table中的getValueAt（”行数”,”列数”）方法循环获取相应行列的表格值并保存到Object二维数组中并实例化Insert类对象，调用Insert类对象进行数据库的信息更新。将获取的数据通过while循环循环插入到选课表中，达到模块的功能。

调用对象的方法后调用监听事件方法刷新界面的数据，达到不再重复操作某一数据的目的

核心代码：

public void S(){  
 Connection connection = null**;** Statement statement = null**;** String sql**;** int a =**0;** try {  
 connection = DBManager.*getConn*()**;** statement = connection.createStatement()**;** //分配课程-学生  
 for (int i= **0;**i<obj.length**;**i++) {  
//INSERT INTO `data`.`选课表` (`学期`, `课程号`, `学生id`, `成绩`) VALUES ('202101', '"+obj[i][0]+"', '"+obj[i][1]+"', NULL);  
 sql = "INSERT INTO `data`.`选课表` (`学期`, `课程号`, `学生id`, `成绩`) VALUES ('202101', '"+obj[i][0]+"', '"+obj[i][1]+"', NULL)";//sql = "update 选课表 set 学生id=' "+obj[i][1]+"' WHERE 课程号='"+obj[i][0]+"'";  
 int result = statement.executeUpdate(sql)**;** System.*out*.println("result: "+result)**;** a++**;** }  
 JOptionPane.*showMessageDialog*(null**,**"分配方案 提交成功")**;** } catch (Exception e) {  
 if (a!=**0**){  
 JOptionPane.*showMessageDialog*(null**,**"成功提交"+a+"条数据")**;** }  
 JOptionPane.*showMessageDialog*(null**,**"更新后续选课信息出现问题\n请输入下表中的学生学号")**;** }  
 }

### 4.1.9教师-个人主页模块设计：

包含教师的个人信息，以及 已经录入成绩按钮和开始录入成绩按钮，教师可以通过当前需求来进行操作。通过不同监听事件达到动态刷新table1表的目的，使得界面简洁明了，更加智能。

图形用户界面, 应用程序

描述已自动生成

核心代码：

根据主界面的传值获取教师的工号和姓名并显示：

public Teacher(User user) {  
 this.user=user**;** Mno = user.getUid()**;** Connection connection = DBManager.*getConn*()**;** try {  
 Statement statement = connection.createStatement()**;** String sql = "SELECT 教师姓名 FROM 教师表 WHERE 教师id='"+Mno+"'"**;** ResultSet resultSet = statement.executeQuery(sql)**;** while (resultSet.next()){  
 Mname = resultSet.getString("教师姓名")**;** }  
 } catch (SQLException e) {  
 e.printStackTrace()**;** }  
  
 initComponents()**;**}

initComponents()**;中的显示代码：**

//---- Tno ----  
Tno.setText(Mno)**;**contentPane.add(Tno)**;**Tno.setBounds(**130, 10, 120, 40**)**;**//---- Tname ----  
Tname.setText(Mname)**;**contentPane.add(Tname)**;**Tname.setBounds(**130, 55, 120, 40**)**;**

### 4.1.10教师-已录入模块：

点击后触发监听事件，查询数据库选课表中成绩字段IS　ＮＵＬＬ的行并返回到Object［］［］数组，利用table１的setModel方法重新赋值table１达到刷新功能，显示到教师个人主界面。

核心代码：

public Object[][] S() {  
 Connection connection = DBManager.*getConn*()**;** try {  
 Statement statement = connection.createStatement()**;** String sql = "Select 学期,课程表.课程号,课程名,学分,学生id,成绩 from 课程表,选课表 where 课程表.课程号=选课表.课程号 and 教师id='"+user.getUid()+"' and 学期='202101' and 成绩 is not null order by 课程表.课程号 ";ResultSet rs=statement.executeQuery(sql)**;** int i1=**0,**i2=**0;** while (rs.next()){  
 i1++**;** }  
 System.*out*.println("行数:"+i1)**;** rs=statement.executeQuery(sql)**;** obj1=new Object[i1][]**;** while (rs.next()){  
 obj1[i2]=new Object[]{  
 rs.getString("学期")**,** rs.getString("课程表.课程号")**,** rs.getString("课程名")**,** rs.getString("学分")**,** rs.getString("学生id")**,** rs.getString("成绩")  
 }**;** System.*out*.println(Arrays.*toString*(obj1[i2]))**;** i2++**;** }  
 } catch (SQLException e) {  
 e.printStackTrace()**;** }  
 return obj1**;**}

### 4.1.11教师－未录入模块设计：

此模块用于查询数据库中的选课表的成绩字段是否为空。利用while（）循环重复判断下一行是否为空，已经在SQL语句中加入where　教师ｉｄ＝＇＋＂ｉｄ＂＋＇动态判断是发出查询请求的教师的所任课的学生，为空则输出整行到Object数组中，达到遍历数据库的选课表，获取未分配成绩的学生信息的效果。实现该查询功能。

核心代码：

public Object[][] S() {  
 Connection connection = DBManager.*getConn*()**;** try {  
 Statement statement = connection.createStatement();  
 String sql = "Select 学期,课程表.课程号,课程名,学分,学生id,成绩 from 课程表,选课表 where 课程表.课程号=选课表.课程号 and 教师id='"+user.getUid()+"' and 学期='202101' and 学生id is not null and 成绩 is null order by 课程表.课程号";ResultSet rs=statement.executeQuery(sql)**;** int i1=**0,**i2=**0;** while (rs.next()){  
 i1++**;** }  
 System.*out*.println("行数:"+i1)**;** obj1=new Object[i1][]**;** rs=statement.executeQuery(sql)**;** while (rs.next()){  
 obj1[i2]=new Object[]{  
 rs.getString("学期")**,** rs.getString("课程表.课程号")**,** rs.getString("课程名")**,** rs.getString("学分")**,** rs.getString("学生id")**,** rs.getString("成绩")  
 }**;** System.*out*.println(Arrays.*toString*(obj1[i2]))**;** i2++**;** }  
 } catch (SQLException e) {  
 e.printStackTrace()**;** }  
 return obj1**;**}

### 4.1.12教师提交模块设计：

该模块使用table１的ｇｅｔｖａｌｕｅａｔ（）方法获取目标单元格的输入数据。ｆｏｒ循环调用方法，并将获取的值赋值给Object数组。通过实例化update＿Ｔｅａｃｈｅｒ类对象将数组传值到对象中，调用update＿Ｔｅａｃｈｅｒ类对象的方法Ｓ（），达到录入成绩和修改成绩的效果。

调用类对象的方法后调用未分配按钮的监听事件刷新页面，防止对某一数据重复操作。

核心代码：

private void button3ActionPerformed(ActionEvent e) {  
 //获取table的数据  
 date3 = new Object[table1.getRowCount()][]**;** for (int i = **0;**i<table1.getRowCount()**;**i++){  
 date3[i]=new Object[]{  
 table1.getValueAt(i**,1**)**,** table1.getValueAt(i**,4**)**,** table1.getValueAt(i**,5**)  
 }**;** System.*out*.println("i:"+i)**;** System.*out*.println(Arrays.*toString*(date3[i]))**;** }  
 if (date3[**0**][**2**]==null){  
 JOptionPane.*showMessageDialog*(null**,**"请输入选课方案")**;** }  
 else{  
 Update\_Teacher\_Grade TeachGrade = new Update\_Teacher\_Grade(date3)**;** TeachGrade.S()**;** }  
 button2ActionPerformed()**;**}

update类对象：

public void S() {  
 Connection connection = null**;** Statement statement = null**;** String sql**;** int a=**0;** try {  
 connection = DBManager.*getConn*()**;** statement = connection.createStatement()**;** //分配课程-学生  
 for (int i= **0;**i<date3.length**;**i++) {  
 sql = "update 选课表 set 成绩='"+date3[i][2]+"' WHERE 课程号='"+date3[i][0]+"' and 学生id='"+date3[i][1]+"'";  
int result = statement.executeUpdate(sql)**;** System.*out*.println(result)**;** a++**;** }  
 JOptionPane.*showMessageDialog*(null**,**"成绩方案 提交成功")**;** } catch (Exception e) {  
 if (a!=**0**){  
 JOptionPane.*showMessageDialog*(null**,**"成功提交"+a+"条数据")**;** }  
 JOptionPane.*showMessageDialog*(null**,**"成绩提交失败\n请输入在0到100范围的成绩")**;** }  
}

### 4.1.13学生主页模块设计：

显示出学生的学号，下拉框选择不同的学期，点击查询按钮触发监听事件，根据选择的不同动态获取学期和登录的学生学号，并将查询结果返回数组，调用刷新模块最后显示到界面的table１中。

图形用户界面, 应用程序

描述已自动生成

核心代码：

//---- Sno ----  
Sno.setText(USno)**;**Sno.setFont(new Font("Microsoft YaHei UI"**,** Font.*PLAIN***, 18**))**;**Sno.setHorizontalAlignment(SwingConstants.*CENTER*)**;**contentPane.add(Sno)**;**Sno.setBounds(**95, 0, 80, 45**)**;**

### 4.1.14学生查询成绩模块：

此模块用于查询学生不同的学期的不同课程成绩。

点击下拉框选择对应的学期，然后点击查询按钮，系统将根据所选择的学期在数据库的选课表中进行附带限制条件的查询，并将查询结构通过Select类对象返回给数组，并调用刷新模块刷新，达到学生个人主页的动态查询成绩。

核心代码：

public Object[][] S\_S() {  
 Connection connection = null**;** Statement statement = null**;** Object[][] obj =null**;** System.*out*.println("Sno:"+sno)**;** System.*out*.println("学期:"+xueqi)**;** try {  
 connection = DBManager.*getConn*()**;** //4、定义sql语句  
 String sql = "SELECT \* \n" +  
 "FROM 选课表,课程表,教师表,`学生表`\n" +  
 "WHERE 选课表.课程号=课程表.课程号\n" +  
 "AND 教师表.教师id=课程表.教师id\n" +  
 "AND 选课表.`学生id`=`学生表`.`学生id`\n" +  
 "AND 选课表.学生id="+sno+" and 学期='"+xueqi+"'";  
//5、获取执行sql的对象、Statement  
 statement = connection.createStatement()**;** ResultSet rs = statement.executeQuery(sql)**;** while (rs.next()){  
 if(rs.getInt("成绩")!=**0**) {  
 i1++**;** }  
 }  
 System.*out*.println("SSG\_i1:"+i1)**;** obj = new Object[i1][]**;** rs = statement.executeQuery(sql)**;** while (rs.next()) {  
 if(rs.getInt("成绩")!=**0**){  
 obj[i2] = new Object[]{  
 rs.getString("学期")**,** rs.getString("课程号")**,** rs.getString("课程名")**,** rs.getString("教师姓名")**,** rs.getDouble("学分")**,** rs.getInt("成绩")  
 }**;** i2++**;** }  
  
 }  
 } catch (SQLException e) {  
 e.printStackTrace()**;** } finally {  
 //7、关闭资源  
 if (statement != null) {  
 try {  
 statement.close()**;** } catch (SQLException throwables) {  
 throwables.printStackTrace()**;** }  
 }  
 if (connection != null) {  
 try {  
 connection.close()**;** } catch (SQLException throwables) {  
 throwables.printStackTrace()**;** }  
 }  
 }  
  
 return obj**;**}

# 5测试

## 5.1登录界面测试

启动系统，登录界面显示

图形用户界面

描述已自动生成

未输入用户名或密码时提示用户　请输入用户名和密码

图形用户界面

描述已自动生成

当系统获取到用户名和密码时进入数据库的user表进行比对，若用户名不存在则提示用户　用户名不存在。

图形用户界面

描述已自动生成

当系统获取到用户名和密码时进入数据库的user表进行比对，若用户名存在但是密码错误则提示用户密码错误。

图形用户界面

描述已自动生成

若用户名存在且输入的密码正确，则直接进入相应的个人主界面。

图形用户界面

描述已自动生成

经过测试，登录模块的功能达到设计的目标，可以执行对应的操作。

## 5.2管理员－个人主页模块测试

登录成功后为管理员界面，显示管理员的工号和姓名并显示相应的按钮

图形用户界面

描述已自动生成

经过测试，管理员主页模块的功能达到设计的目标，可以执行对应的操作。

## 5.3管理员分配（教师）模块测试

点击后进入教师的分配课程界面，并显示该页面的主要按钮和表格

表格

描述已自动生成

经过测试，管理员分配（教师）模块的功能达到设计的目标，可以执行对应的操作。

## 5.4管理员已分配教师模块测试

点击查询已经分配教师的课程按钮，显示课程表中教师id字段部位空的行的信息，达到动态刷新上半部的课表的效果。

表格

描述已自动生成

经过测试，管理员已经分配教师模块的功能达到设计的目标，可以执行对应的操作。

## 5.5管理员未分配教师模块测试

点击查询未分配教师的课程按钮，显示课程表中教师id字段部位为空的行的信息，达到动态刷新上半部的课表的效果。此时显示的表中列名为教师id的列整体为空

图形用户界面, 表格

描述已自动生成

经过测试，管理员未分配教师模块的功能达到设计的目标，可以执行对应的操作。

## 5.6管理员提交分配模块测试

点击提交按钮，触发监听事件，将编辑后的单元格的值输入到数据库的课程表中。

如果未输入数据提交提示用户请输入数据。如果输入的数据在不是下半部分的教师表中的数据提示请输入教师表中的工号。

图形用户界面

描述已自动生成

图形用户界面

描述已自动生成

图形用户界面

描述已自动生成

## 5.7管理员－分配（学生）模块测试

点击按钮，跳转到分配学生的界面，在页面中显示下半部分的学生表中的所有学生，以及上半部分的按钮和空表table１。



经过测试，管理员分配（学生）模块的功能达到设计的目标，可以执行对应的操作。

## 5.8管理员－学生未分配模块测试

显示数据库中课程表中的课程信息，已经未在选课表中出现的选课信息。

图形用户界面, 表格

描述已自动生成

经过测试，管理员未分配（学生）模块的功能达到设计的目标，可以执行对应的操作。

## 5.9管理员提交选课模块测试

点击提交按钮，触发监听事件，将编辑后的单元格的值输入到数据库的选课表中。

如果未输入数据提交提示用户请输入数据。如果输入的数据在不是下半部分的学生表中的数据提示请输入学生表中的学号。

图形用户界面

描述已自动生成

图形用户界面

描述已自动生成

图形用户界面

描述已自动生成

经过测试，管理员提交选课模块的功能达到设计的目标，可以执行对应的操作。

## 5.10管理员－学生已分配模块测试

点击查询已分配按钮后上半部分的课表刷新显示在数据库的选课表查询到的已经分配课程的学生的结果。

图形用户界面, 表格

描述已自动生成

经过测试，管理员学生已分配模块的功能达到设计的目标，可以执行对应的操作。

## 5.11教师个人主页模块测试

用于显示教师的个人信息和查询、录入学生的成绩。

图形用户界面, 应用程序

描述已自动生成

经过测试，教师主页模块的功能达到设计的目标，可以执行对应的操作。

## 5.12教师-开始录入成绩查询模块

显示当前任课课程中还没有录入成绩的学生的名单。

表格

描述已自动生成

经过测试，教师-开始录入成绩模块的功能达到设计的目标，可以执行对应的操作。

## 5.13教师-提交成绩查询模块

若提交为空提示用户 请输入数据

提交的成绩需要在0到100范围内，如果输入的范围部队提示用户输入错误

图形用户界面

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

图形用户界面

描述已自动生成

经过测试，教师-提交成绩模块的功能达到设计的目标，可以执行对应的操作。

## 5.14学生个人主页模块测试

显示学生个人信息，已经学期选择下拉框和查询按钮。

图形用户界面, 应用程序

描述已自动生成

经过测试，学生个人主页模块的功能达到设计的目标，可以执行对应的操作。

## 5.15学生-查询成绩模块测试

学生点击查询按钮后，成绩表刷新显示所选学期的全部课程以及成绩。

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 应用程序

描述已自动生成

表格

描述已自动生成

经过测试，学生-查询成绩模块的功能达到设计的目标，可以执行对应的操作。

## 5.16整体测试

分别登录管理员、教师、学生等不同类型的用户账号。进行 管理员为教师分配课程、为学生选课，教师查询任课课程的学生名单、录入任课课程的学生成绩，学生查询不同学期的成绩。

经过测试和调试，学生成绩管理系统的分模块全部测试完毕，每个模块的功能都达到预期水平。

然后进行整体流程测试，管理员为教师分配课程，接着为学生分配课程，被分配到课程的教师登录到个人主页录入分配到的学生的成绩，最后被录入成绩的学生登录主页查询本学期选修课程的考核成绩。

经过测试和调试，学生成绩管理系统的功能模块可以正确无误的显示出预期的数据，系统设计的目标基本实现，测试结束，软件系统达到预期效果。

# 6总结

本次的实训是对《Java程序设计语言》理论课程的补充，也是检验自己真正学会了多少知识、学到了多少知识的一次机会，同时也是一个初探基于Java开发软件的尝试，为后续的软件开发积累经验。

这次的实训也让我受益匪浅，不但了解了Java开发的基本步骤，包括技术可行性分析、需求分析、系统模块的实现、数据库的使用等等，同时也巩固和加强了我对Java软件开发、数据库知识的理论知识，为后续Java开发的框架的学习奠定必要的基础。初步掌握基于Java程序语言和链接数据库进行开发的经验；熟悉和掌握学生管理系统的需求分析、可用性分析、数据库设计、系统功能模块化设计，完成测试。提高资料收集和文档写作、整理能力。也意识到了自己掌握的知识还是太少了，面对数以万计的精美软件或者许许多多的优秀程序员，感到十分的羞愧。

不过也激发了我继续学习Java技术的热情。同样的开发项目，不同的同学就可以达到简洁又美观的样式，犹如魔术一般吸引着我去深入地了解Java语言与数据库的魅力。

# 7参考文献

Java 2实用教程（第五版）——耿祥义，张跃平编著——清华大学出版社  
Java高级程序设计实战教程——戴远泉——人民邮电出版社