

大连理工大学城市学院

Java编程实践报告

**实践题目： 学生学籍管理系统**

**组长姓名： 刘忠坤**

**组员姓名： 刘忠坤**

**完成日期： 2019/4/2**

**目 录**

[1 摘要 1](#_Toc269202533)

[2 系统设计与实现 2](#_Toc269202534)

[2.1 系统分析 2](#_Toc269202535)

[2.1.1 需求分析 2](#_Toc269202536)

[2.1.2 可行性分析 2](#_Toc269202537)

[2.1.3开发及运行环境分析 2](#_Toc269202538)

[2.2 系统总体设计 3](#_Toc269202539)

[2.2.1 系统总体功能描述 3](#_Toc269202540)

[2.2.2 系统总体流程 3](#_Toc269202541)

[2.2.3 系统功能模块图 4](#_Toc269202542)

[2.3 数据库设计 5](#_Toc269202543)

[2.3.1 E-R图 5](#_Toc269202544)

[2.3.2 数据表 6](#_Toc269202545)

[2.3.3 连接数据库模块 7](#_Toc269202546)

[2.4 系统详细设计 9](#_Toc269202547)

[2.4.1 系统功能实现小组分工描述 9](#_Toc269202548)

[2.4.2 管理员登录模块 9](#_Toc269202549)

[2.4.3 系统主界面模块 10](#_Toc269202550)

[2.4.4 学生学籍管理模块 11](#_Toc269202551)

[2.4.5 学生成绩信管理模块 14](#_Toc269202552)

[2.4.7 服务器端处理过程 19](#_Toc269202554)

[2.5 系统测试设计 20](#_Toc269202555)

[2.5.1测试过程描述 20](#_Toc269202556)

[2.5.2 测试结果分析 20](#_Toc269202557)

[3 结 论 21](#_Toc269202558)

# 1 摘要

为了巩固Java语言程序设计课程的主要知识，提高编程实际的开发能力，基于课程的技术要求和作业标准，设计并实现了 学生学籍信息管理系统。

该系统的主要功能是 对学生学籍信息和成绩进行管理

，系统的开发语言是  **java**  ，开发环境是 eclipse ，数据库是 MySql 。

本系统具有的特点是： 用表格进行数据的查询输出 ，优点是： 可以用鼠标对数据的选取操作以进行修改和删除

# 2 系统设计与实现

## 2.1 系统分析

### 2.1.1 需求分析

在一些学校等机构，随着学生数量的不断增加，学生的信息不断增多，人工管理信息的难度也越来越大。而且效率也是很低的。所以如何自动高效地管理信息是这些年来许多人所研究的。随着这些年电脑计算机的速度质的提高，成本的下降，IT互联网大众趋势的发展。我们使用电脑的高效率才处理数据信息成为可能。 学生学籍管理系统的出现，正是管理人员与信息数据，计算机的进入互动时代的体现。友好的人机交互模式，清晰简明的图形界面，高效安全的操作使得我们对成千上万的信息的管理得心应手。 学生学籍管理系统，以SQL数据库作为后台信息存储，Java作为前台系统的语言。提供了对学生信息添加，查询，修改，删除的功能。实现了最基本的信息管理。

### 2.1.2 可行性分析

1.技术可行性：就目前使用的开发技术来说建议系统的功能目标应该能够达到；利用现有的技术在规定的期限内开发工作基本能够完成；而且可以事先建立一个数据库包含所需要的数据资源；有开发和测验所需要的计算机资源。

2.法律可行性：所做的系统都是在法律许可范围内进行的，并未涉及侵犯版权等或其他问题。

3.经济可行性：经济可行性分析中最重要的内容之一是成本－－效益分析。对于一个基于计算机系统的研制项目要在经济方面评价其是否合理，成本－－效益分析要估计出研制开发的花费并与收益（包括有形的可以用货币计数的及无形的利益）衡量比较。

### 2.1.3开发及运行环境分析

1.硬件平台

CPU：INTEL I7-7700HQ 2.8GHZ

内存：8000MB以上

2.软件平台

操作系统：Windows10

数据库：SQL Server2005

开发工具：JDK 1.6.0 Eclipse3.4

## 2.2 系统总体设计

### 2.2.1 系统总体功能描述

学生学籍管理系统需要具备以下几个主要功能：

1. 学生管理：将学生信息存储于系统中，管理员可以管理学生信息，如查询学生信息、添加新学生信息、修改和删除学生信息等；

2. 成绩管理：将成绩信息存储于系统中，管理员可以添加成绩、修改成绩和删除成绩，并可以查阅总体成绩的情况；

3. 信息查询：管理员可以查询特定的学生信息和成绩的大体情况。

### 2.2.2 系统总体流程

系统学生学籍管理流程如图2-1所示。

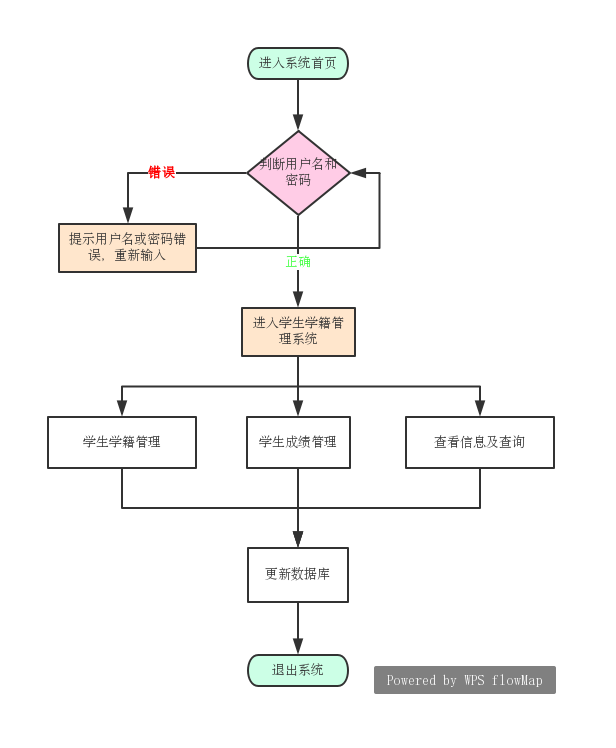
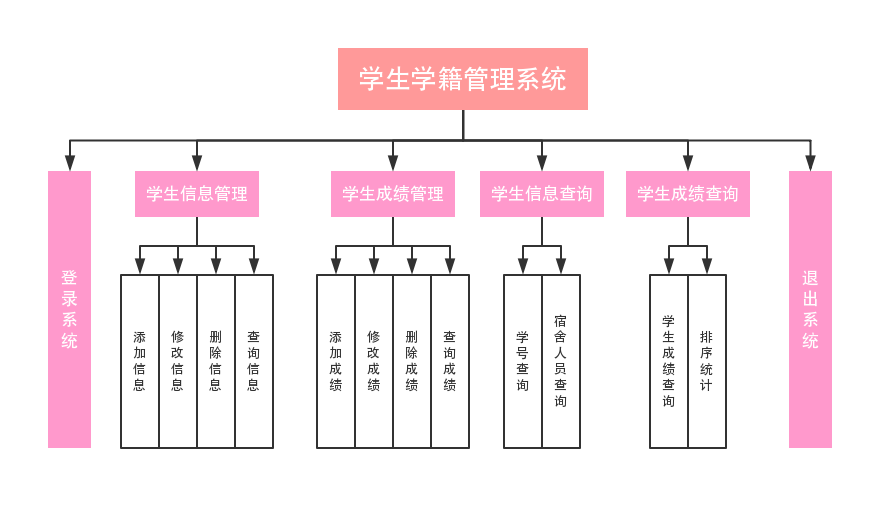


图2-1 系统学生学籍管理流程图

### 2.2.3 系统功能模块图

根据系统的设计要求，图书信息管理系统实现了五个完整的功能。根据这些功能要求，设计的系统功能模块如图2-2所示。

图2-2 系统功能模块



学生学籍管理系统各模块功能要求分析如下：

1. 用户登录要求

实现对管理员用户登录的验证、标识，为用户对系统的操作提供授权依据，在这一模块中，首先要求管理员输入自己的登录用户名和登录密码，然后系统对用户名和密码进行验证，判断用户的身份。

2. 学籍管理要求

由于学生众多，该系统需要存储每个学生的学号，以便修改信息的时候通过学号从数据库中快速调出学籍信息，用户的基本信息包括学号、姓名、性别、班级、宿舍和电话等，管理员可以添加新学生信息、查询学生信息、修改学生信息和删除学生信息。

3. 成绩管理要求

为了方便成绩的管理，需要把每名学生的相关信息添加到数据库中，以便通过学号从数据库中快速调出成绩信息，成绩信息包括学号，各科成绩，总成绩等，管理员可以添加成绩、查询成绩、修改成绩和删除成绩。

4. 查询管理要求

通过本系统，管理员可以进行学生信息和成绩的查询，并可以统计总体成绩的情况。

## 2.3 数据库设计

学生学籍系统不仅要求数据库功能强大，而且要有很好的安全保障机制。SQL Server数据库是当前应用最为广泛的数据库，它可以很好第支持各种信息，而且具有较高的数据安全保障，是一般公司、企业首选的数据库之一，本系统采用SQL Server数据库进行创建。

### 2.3.1 E-R图

本系统的数据库表的E-R图如图2-3至2-6所示：

（1）管理员实体的E-R图

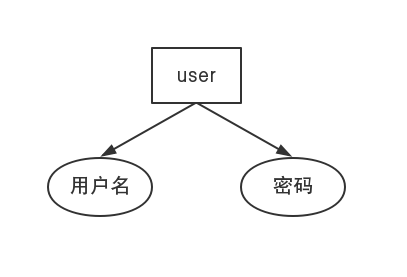


图2-3 管理员实体

（2）学生学籍实体的E-R图

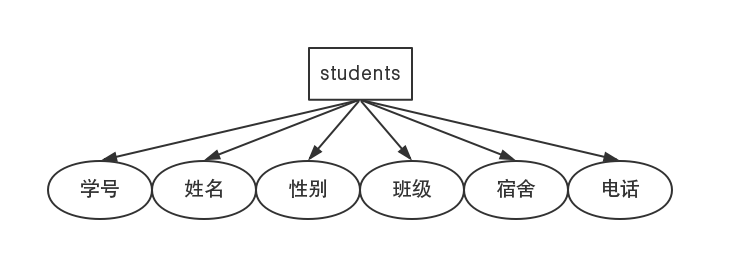


图2-4 学生学籍实体

（3）学生成绩实体的E-R图

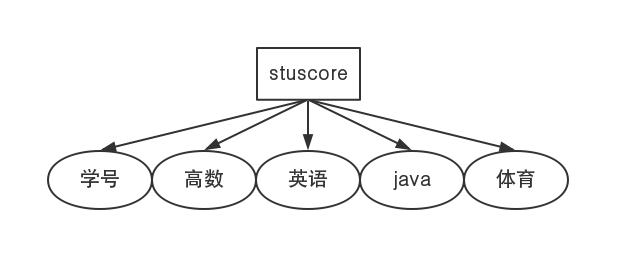


图2-5 学生成绩实体

（4）各实体之间的联系图

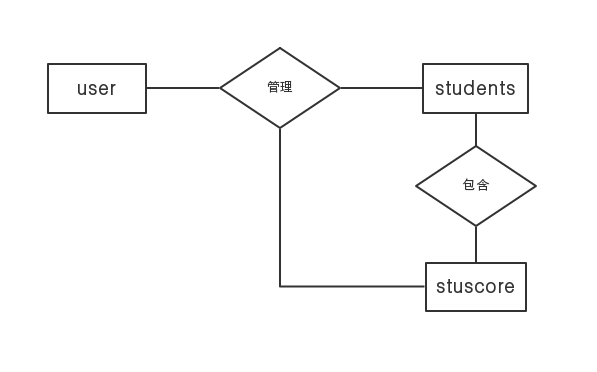


图2-6 实体联系图

### 2.3.2 数据表

根据系统的设计要求和模块功能，进行数据库的分析和设计。由图2-3到图2-6可以知道，系统拥有图书管理员、用户（读者）、图书等实体。在数据库中，可以使用表描述这些实体。另外，还需要存储图书借阅记录，也需要用表来表示，这些表的名称以及表的说明如下：

1. 管理员表Admin

用来保存管理员用户信息，结构如表2-1所示。

表2-1 管理员表Admin

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 说明 |
| name | 文本 | 管理员用户名，主键 |
| pass | 文本 | 管理员密码 |

2. 学生信息表students

用来保存图书信息，结构如表2-2所示。

表2-2 学生信息表students

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 说明 |
| number | 数字 | 学号，主键 |
| name | 文本 | 姓名 |
| gender | 文本 | 性别 |
| class | 文本 | 班级 |
| room | 文本 | 宿舍 |
| phone | 文本 | 电话 |

4. 学生成绩表stuscore

用来保存读者借阅信息，结构如表2-3所示。

表2-3借阅表stuscore

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据类型 | 说明 |
| number | 数字 | 学号，主键 |
| gaoshu | 数字 | 高数 |
| yingyu | 数字 | 英语 |
| java | 数字 | Java |
| tiyu | 数字 | 体育 |
| scores | 数字 | 总成绩 |

### 2.3.3 连接数据库模块

系统所需要的信息都存储在数据库中，例如图书信息、读者信息、借阅信息等，要对这些信息进行操作，就必须连接数据库，为了省去每次操作都要编写连接数据库程序的步骤，把连接数据库操作封装到一个类jdbc.java中，在不同的模块中调用这个类就可以对数据库进行连接，执行相应的数据库操作，这样就使得连接数据库安全高效，程序代码简洁清晰，也符合Java面向对象的思想。

连接数据库类Jdbc.java的代码如下：

**package lzk;**

**import java.sql.\*;**

**import javax.swing.JOptionPane;**

**public class Jdbc {**

**static String url="jdbc:mysql://localhost:3306/mysql?useSSL=false&useUnicode=true&characterEncoding=UTF8&serverTimezone=UTC";**

**static String root="root";**

**static String rootpass="123456789";**

**static int sum;**

**public static boolean Login(String name,String pass) {**

**boolean flag=false;**

**try {**

**Class.forName("com.mysql.cj.jdbc.Driver");**

**Connection conn=DriverManager.getConnection(url,root,rootpass);**

**//Connection conn=DriverManager.getConnection("jdbc:mysql://210.30.108.222:3306/u201712103?useSSL=false&useUnicode=true&characterEncoding=UTF8", "u201712103","u201712103");**

**PreparedStatement ps=conn.prepareStatement("select \* from admin where name=? and pass=?");**

**ps.setString(1,name);**

**ps.setString(2,pass);**

**ResultSet rs=ps.executeQuery();**

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

**flag=true;**

**conn.close();**

**}catch(Exception e) {**

**e.printStackTrace();**

**}**

**return flag;**

**}**

**public static void insert(String sqlstr) {**

**try {**

**Class.forName("com.mysql.cj.jdbc.Driver");**

**Connection conn=DriverManager.getConnection(url,root,rootpass);**

**PreparedStatement ps=conn.prepareStatement(sqlstr);**

**sum=ps.executeUpdate();**

**if(sum>=1){**

**JOptionPane.showMessageDialog(null,sum+"行添加成功！","添加",JOptionPane.INFORMATION\_MESSAGE);**

**}**

**conn.close();**

**}catch(SQLIntegrityConstraintViolationException e) {**

**JOptionPane.showMessageDialog(null,"学号不能重复！","添加",JOptionPane.ERROR\_MESSAGE);**

**}catch(Exception e) {**

**e.printStackTrace();**

**}**

**}**

**public static void delete(String sqlstr) {**

**try {**

**Class.forName("com.mysql.cj.jdbc.Driver");**

**Connection conn=DriverManager.getConnection(url,root,rootpass);**

**PreparedStatement ps=conn.prepareStatement(sqlstr);//"delete from students where number= ?"**

**sum=ps.executeUpdate();**

**if(sum>=1){**

**JOptionPane.showMessageDialog(null,sum+"行删除成功！","删除",JOptionPane.INFORMATION\_MESSAGE);**

**}**

**conn.close();**

**}catch(Exception e) {**

**e.printStackTrace();**

**}**

**}**

**public static void update(String sqlstr) {**

**try {**

**Class.forName("com.mysql.cj.jdbc.Driver");**

**Connection conn=DriverManager.getConnection(url,root,rootpass);**

**PreparedStatement ps=conn.prepareStatement(sqlstr);//"delete from students where number= ?"**

**sum=ps.executeUpdate();**

**System.out.println(sum);**

**if(sum>=1){**

**JOptionPane.showMessageDialog(null,sum+"行修改成功！","修改",JOptionPane.INFORMATION\_MESSAGE);**

**}**

**conn.close();**

**}catch(Exception e) {**

**e.printStackTrace();**

**}**

**}**

**public static ResultSet select(String sqlstr) {**

**ResultSet rs = null;**

**try {**

**Class.forName("com.mysql.cj.jdbc.Driver");**

**Connection conn=DriverManager.getConnection(url,root,rootpass);**

**PreparedStatement ps=conn.prepareStatement(sqlstr);//"select from students where name=? and pass=?"**

**rs=ps.executeQuery();**

**conn.close();**

**}catch(Exception e) {**

**e.printStackTrace();**

**}**

**return rs;**

**}**

**}**

## 2.4 系统详细设计

### 2.4.1 系统功能实现小组分工描述

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **姓名** | **实现模块及功能** | **备注** |
| **1** | **刘忠坤** | **登录模块login.java 进行登录验证** |  |
| **2** | **刘忠坤** | **数据库模块Jdbc.java 连接数据库** |  |
| **3** | **刘忠坤** | **主界面Student.java 为其他模块提供界面** |  |
| **4** | **刘忠坤** | **其他各功能模块** |  |

### 2.4.2 管理员登录模块

为保证系统的完全性，只有管理员才可以对系统进行操作，因此，需要对用户登录信息进行验证，管理员的用户名和密码存放在数据库中，通过文本框获得用户输入的用户名和密码，然后与数据库中的用户名和密码进行比较，如果匹配，则进入系统，否则提示用户名和密码不正确。

1. 管理员登录界面



1. 技术实现主要代码（详见源程序）

3. 开发说明

管理员登录由Login类继承了Jframe类，首先声明了标签、按钮、文本框、密码框和一个Jframe窗口，然后通过构造方法创建标签、按钮、文本框、密码框，并添加到窗口容器中，最后设置窗口的大小和可见行；

然后为按钮增加事件监听，实现AciontListener接口完成用户名、密码的检测并登录。

### 2.4.3 系统主界面模块

系统主界面也就是登录后进入的第一个浏览界面，这也是图书信息管理系统的主要模块，在主页面中应该包括系统葛模块的连接，通过主页面可以打开如何一个功能模块，实现各种不同的功能。

1. 系统主界面



1. 技术实现主要代码（详见源程序）

3. 开发说明

Studnet类继承了Jframe类，定义了一个菜单栏、5个菜单、13个菜单项和一个滚动面板，然后通过构造函数设置窗口的布局、窗口的大小和标题，将菜单项加入到各自的菜单中，把菜单加入到菜单栏中，并将菜单栏加入到窗口中；

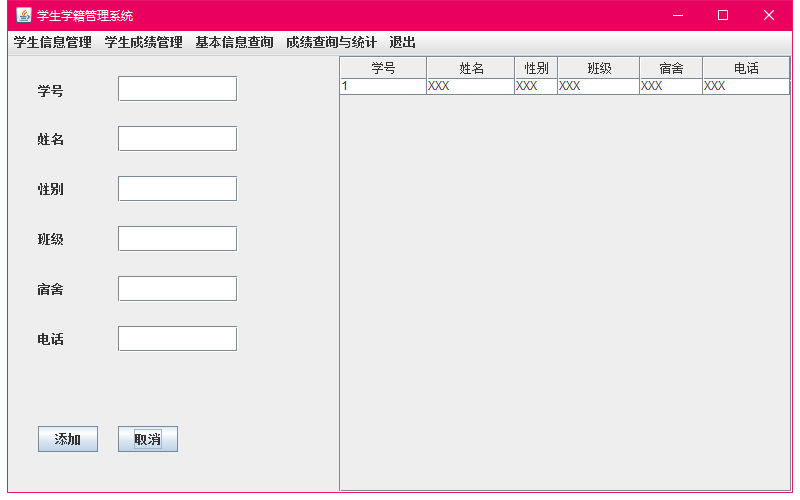
当单击菜单项时候产生单击事件，传递给监听者，实现接口并处理监听到的事件。

### 2.4.4 学生信息管理模块

管理员可以对学生信息进行管理，包括增加新学生、删除学生、修改学生和查询学生信息。

1. 学生信息增加模块

（1）学生信息增加界面



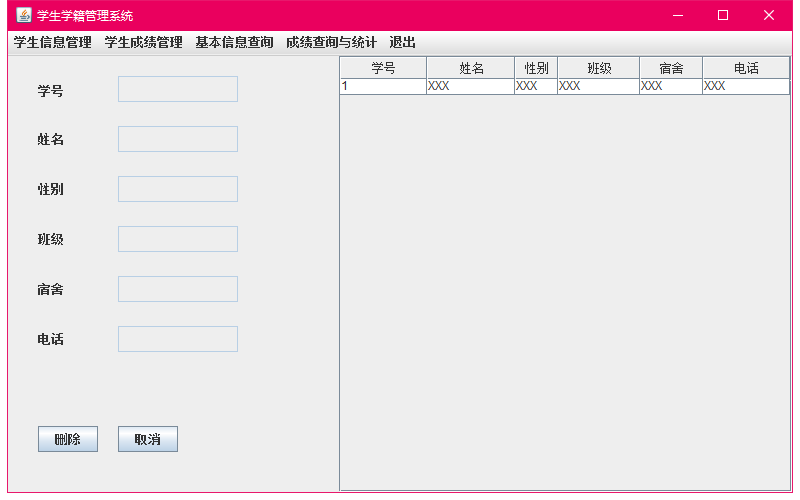
（2）技术实现主要代码（详见源程序）

（3）开发说明

在系统主界面中，单击“学生信息管理”->“添加信息”菜单命令，打开添加信息界面，该界面包含6个标签、6个文本框和两个按钮，在文本框中输入学生信息，单击“添加”按钮就可以把数据信息添加到数据库中。

2. 学生信息删除模块

（1）学生信息删除界面



（2）技术实现主要代码（详见源代码）

（3）开发说明

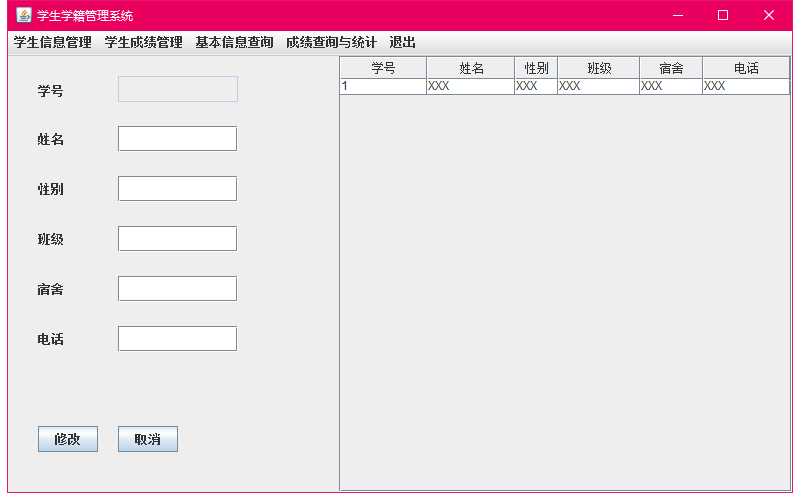
在系统主界面中，单击“学生信息管理”->“删除信息”菜单命令，打开删除用户信息界面，界面包含6个标签、6个文本框和两个按钮，在右侧表格中点击需要删除的信息则把学生信息显示在对应的文本框中，单击“删除”按钮就可以删除数据库中的信息；

单击“取消”按钮就可以清空文本框内容取消删除操作。

为两个按钮分别添加事件监听，并对监听到的事件进行处理。

3. 学生信息修改模块

（1）学生信息修改界面



（2）技术实现主要代码（详见源代码）

（3）开发说明

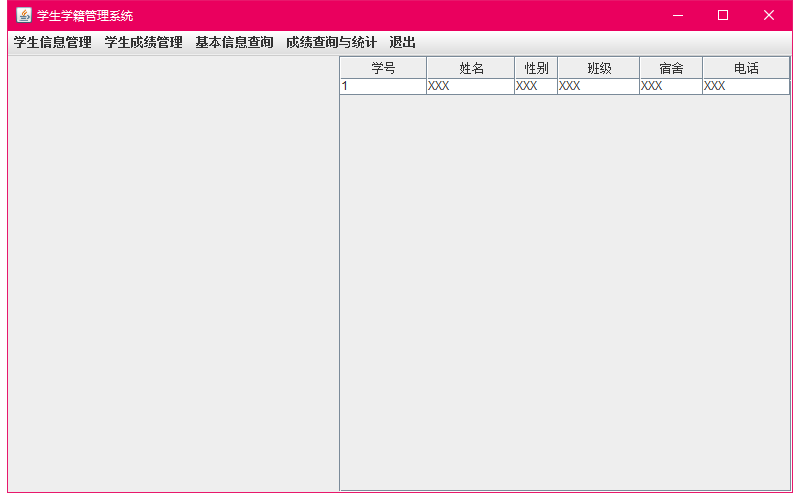
在系统主界面中，单击“学生信息管理”->“修改信息”菜单命令，打开修改学生信息界面，界面包含6个标签、6个文本框和两个按钮，在右侧表格中点击需要修改的信息则把学生信息显示在对应的文本框中，单击“删除”按钮就可以删除数据库中的信息；

单击“取消”按钮就可以清空文本框内容取消修改操作。

为两个按钮分别增加事件监听，并对监听到的事件进行处理。

4. 学生信息查询模块

（1）学生信息查询界面



（2）技术实现主要代码（详见源代码）

（3）开发说明

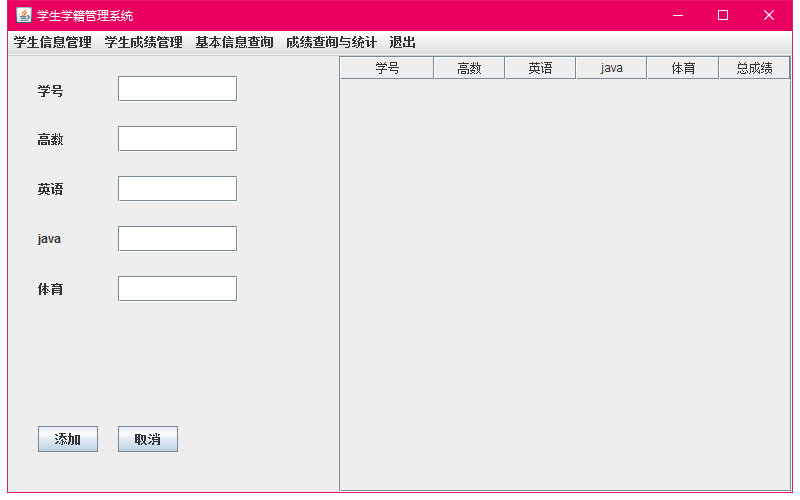
在系统主界面中，单击“学生信息管理”->“查询信息”菜单命令，打开查询学生信息界面，界面包含一个滚动面板、一个表格，点击完查询就可以把全部信息显示出来。

### 2.4.5 学生成绩管理模块

管理员可以对学生成绩进行管理，包括增加新成绩、删除成绩、修改成绩和查询成绩信息。

1. 学生成绩增加模块

（1）学生成绩增加界面



（2）技术实现主要代码（详见源代码）

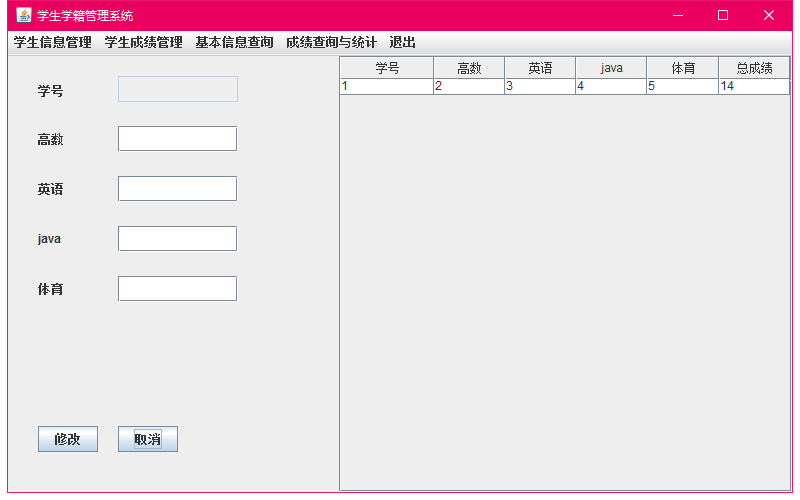
（3）开发说明

在系统主界面中，单击“学生成绩管理”->“添加成绩”菜单命令，打开添加学生成绩界面，界面包含5个标签、5个文本框和两个按钮，在文本框中输入成绩信息；

为按钮添加事件监听，添加图书信息要求每项信息都不能为空，否则弹出消息提示对话框。

2. 学生成绩删除模块

（1）学生成绩删除界面



（2）技术实现主要代码（详见源代码）

（3）开发说明

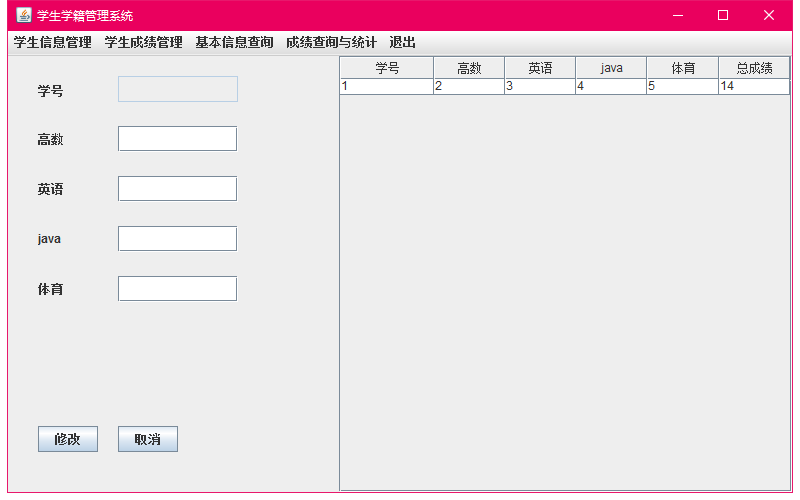
在系统主界面中，单击“学生成绩管理”->“删除成绩”菜单命令，打开删除学生成绩信息界面，界面包含5个标签、5个文本框和两个按钮，在右侧表格中点击需要删除的信息则把学生成绩信息显示在对应的文本框中，单击“删除”按钮就可以删除数据库中的信息；

单击“取消”按钮就可以清空文本框内容取消删除操作。

为按钮添加事件监听，并对监听到的事件进行处理。

3. 学生成绩模块

（1）学生成绩修改界面



（2）技术实现主要代码（详见源代码）

（3）开发说明

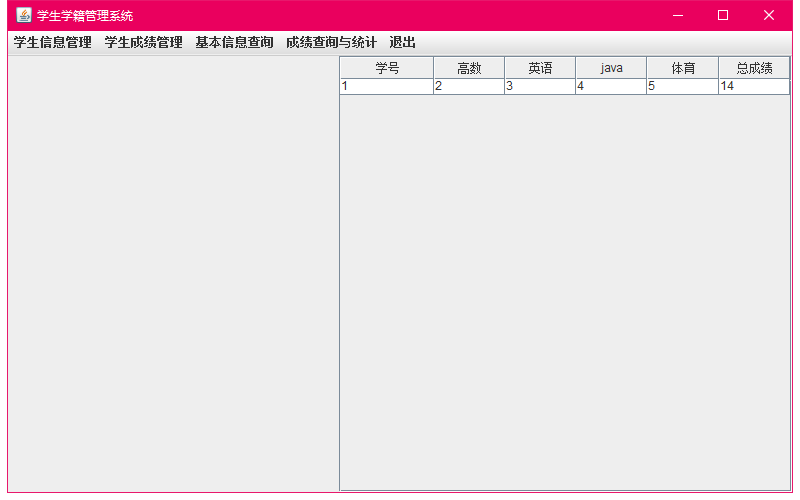
在系统主界面中，单击“学生成绩管理”->“修改成绩”菜单命令，打开修改学生成绩界面，界面包含5个标签、5个文本框和两个按钮，在右侧表格中点击需要修改的信息则把学生成绩信息显示在对应的文本框中，单击“删除”按钮就可以删除数据库中的信息；

单击“取消”按钮就可以清空文本框内容取消修改操作。

为按钮增加事件监听，并对监听到的事件进行处理。

4. 学生成绩查询模块

（1）学生成绩查询界面



（2）技术实现主要代码（详见源代码）

（3）开发说明

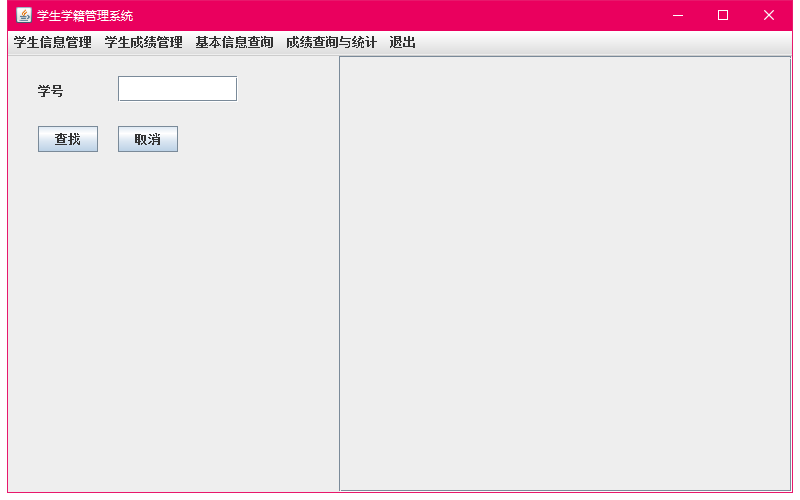
在系统主界面中，单击“学生成绩管理”->“查询成绩”菜单命令，打开查询学生成绩界面，界面包含一个滚动面板、一个表格，点击完查询就可以把全部信息显示出来。

### 2.4.6 特定查询模块

管理员不仅可以对学籍信息和成绩进行管理，还可以查找特定学号的信息。

1. 学号查询信息模块

（1）学号查询信息界面



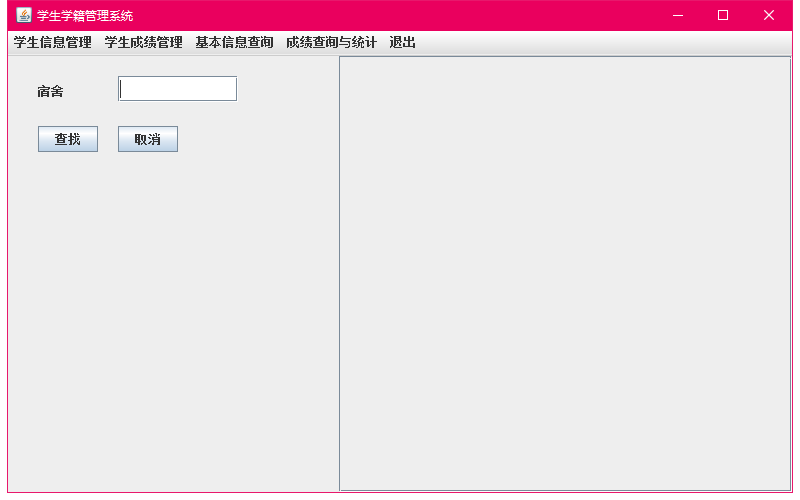
（2）技术实现主要代码（详见源代码）

（3）开发说明

在系统主界面中，单击“基本信息查询”->“学号查询”菜单命令，打开学号查询界面，界面包含1个标签、1个文本框和两个按钮，在文本框中输入学号，单击“查找”按钮就可以查找特定学号的信息。

2. 宿舍人员查询模块模块

（1）宿舍人员查询界面



（2）技术实现主要代码（详见源代码）

（3）开发说明

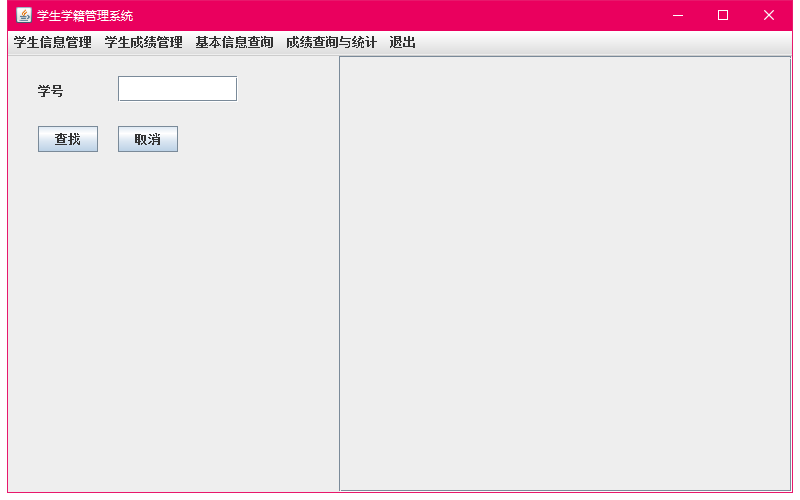
在系统主界面中，单击“基本信息查询”->“宿舍人员查询”菜单命令，打开宿舍人员查询界面，界面包含1个标签、1个文本框和两个按钮，在文本框中输入宿舍号，单击“查找”按钮就可以查找特定宿舍的人员。

### 2.4.7 查询排序模块

管理员不仅可以对学籍信息进行查询，还可以队查找的成绩进行排序。

1. 学号查询成绩模块

（1）学号查询成绩界面



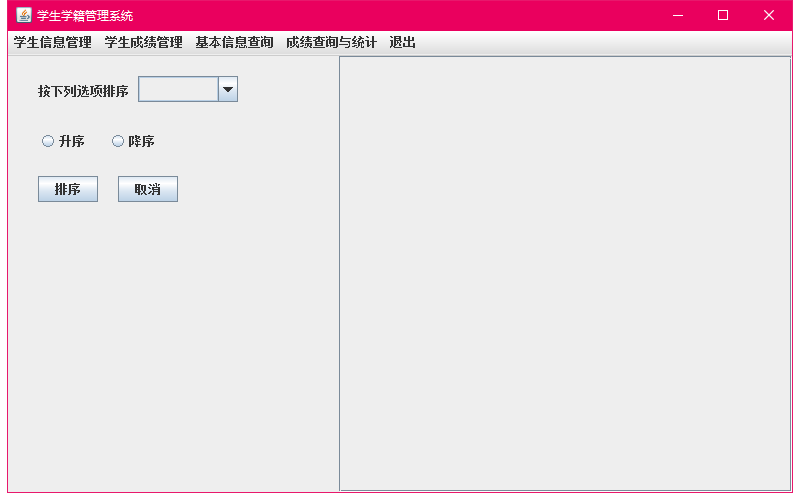
（2）技术实现主要代码（详见源代码）

（3）开发说明

在系统主界面中，单击“成绩查询与统计”->“学生成绩查询”菜单命令，打开查询成绩界面，界面包含1个标签、1个文本框、1个下拉菜单和1个查询按钮，在文本框中输入查询条件，并通过下拉菜单选择按图书或按读者查询，单击“查询”按钮就可以把查询结果显示出来。

2. 成绩查询模块

（1）成绩查询界面



（2）技术实现主要代码（详见源代码）

（3）开发说明

在系统主界面中，单击“成绩查询与统计”->“排序统计”菜单命令，打开查询学生成绩界面，界面包含1个标签、1个下拉菜单、2个单选框和2个按钮，在单选框选中查询条件，并通过下拉菜单选择按各科成绩或总成绩查询，单击“排序”按钮就可以把查询结果显示出来。

## 2.5 系统测试设计

### 2.5.1测试过程描述

各模块运行效果的描述和截图：

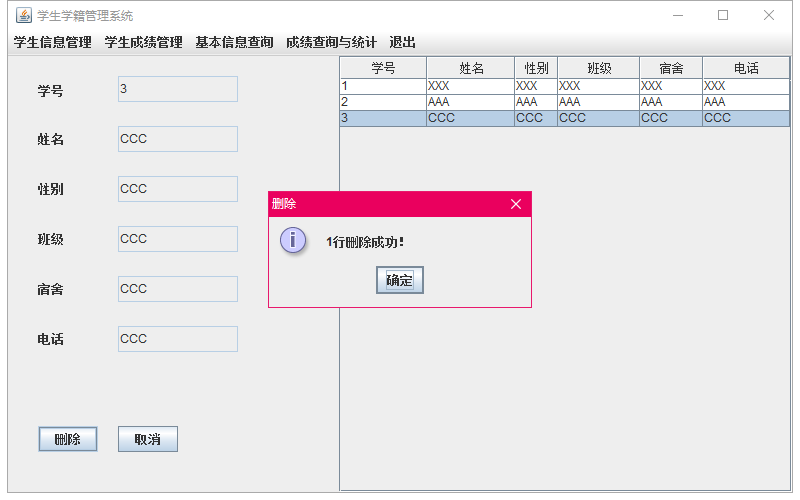
1. 添加信息



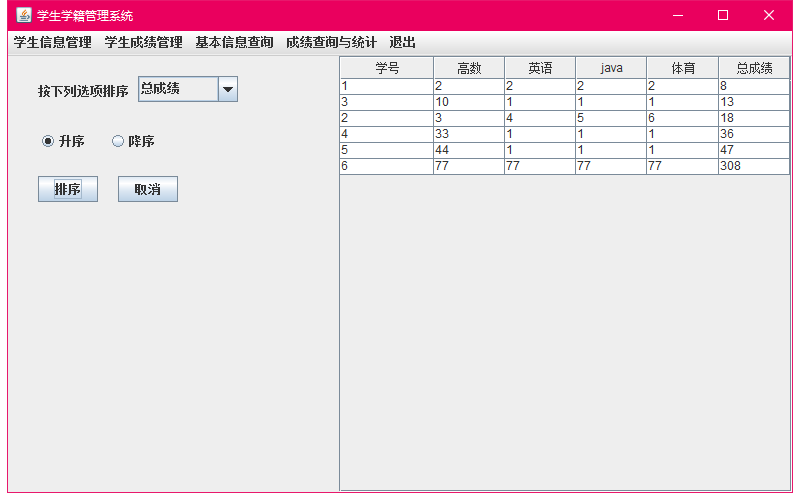
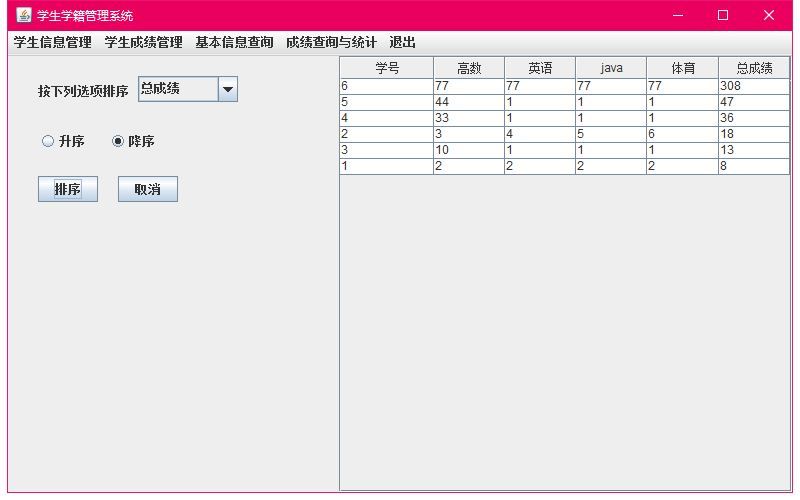
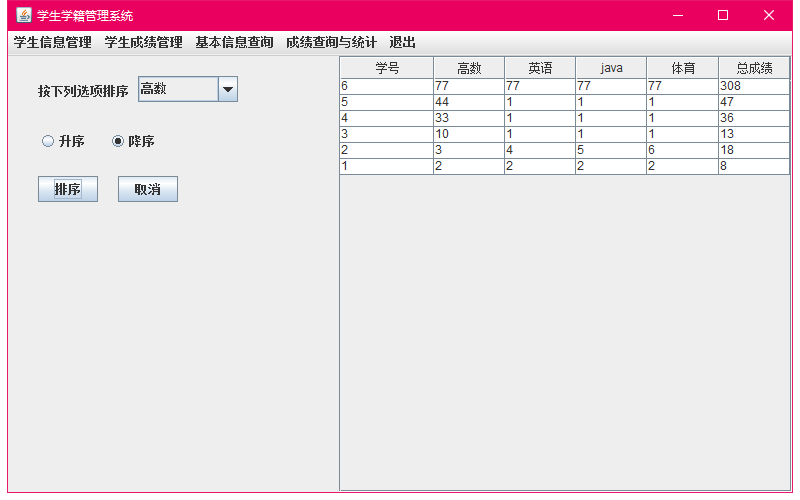
1. 修改信息



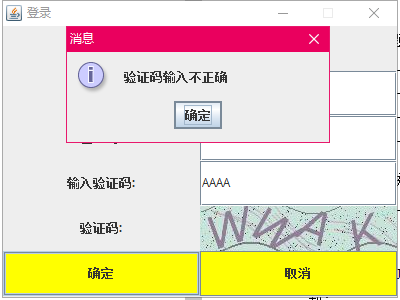
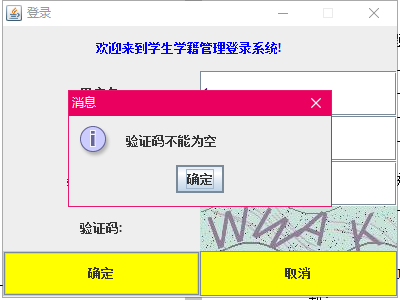
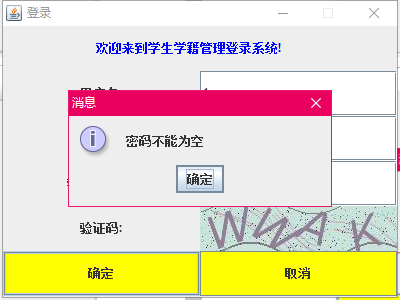
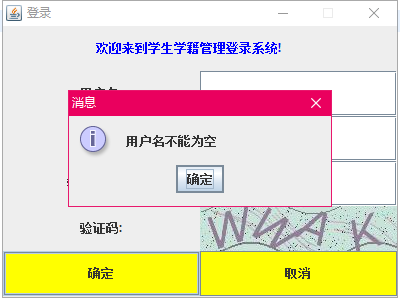
1. 删除信息



1. 排序统计



1. 各种错误的处理



### 2.5.2 测试结果分析

根据以上实际测试结果， 该系统的题目是：学生学籍管理系统，经过测试能够正确地实现了所要求的功能：登陆窗体，学籍的管理，成绩的管理，对成绩的查询与排序。开发过程中采用的工具和环境是：eclipse和sqlserver2000，系统主要功能是：能验证登陆，对学生学籍信息和成绩的增删改查等具有安全性高，功能齐全，操作简单的优点。

# 3 结 论

该系统的题目是： **学生学籍管理系统** ，经过测试能够正确地实现了所要求的功能：对学生学籍和成绩的增删改查及排序 。开发过程中采用的工具和环境是： JDK eclipse ，系统主要功能是： 学生学籍信息的增删改查及学生成绩的增删改查，并可以对成绩进行排序的操作

具有 简易，可视化操作 的优点。

当然此系统仍存在一些不足之处：

对学生学籍信息的查询处理有欠缺

通过本次Java语言程序设计实践项目的设计、开发和实现过程，深深体会到：

课程设计是培养学生综合运用所学知识,发现,提出,分析和解决实际问题,锻炼实践能力的重要环节,是对学生实际工作能力的具体训练和考察过程. 本次课程设计虽然很辛苦，实在是受益匪浅。本来这门课的知识学的就不够扎实，本次课程设计，在设计过程中碰到了很多问题，刚开始要设计的时候，还真不知道从哪里下手但最终在图书、同学和老师的帮助下都得到了解决，让我学会了好多书本上没有的东西，通过本次课程设计我也能将课本上的知识融会贯通，起到了很好的辅助学习的效果，但是我发现我学到的知识比整整一个学期学到的都多。理论和实践的相结合是学习最有效的方法。在设计的过程中发现了自己的不足之处，对以前所学过的知识理解得不够深刻，掌握得不够牢固，通过这次课程设计之后，一定把以前所学过的知识重新温故。

通过这次课程设计使我懂得了理论与实际相结合是很重要的，只有理论知识是远远不够的，只有把所学的理论知识与实践相结合起来，从理论中得出结论，才能真正为社会服务，从而提高自己的实际动手能力和独立思考的能力。

**作业报告成绩评定**

|  |  |
| --- | --- |
| **作业报告成绩：** | |
| **报告格式： □完全符合要求 □符合要求 □基本符合要求 □不符合要求** | |
| **报告内容：□内容很全面、思路很清晰 □内容比较全面、思路较为清晰**  **□内容基本正确、思路基本明晰 □内容不全面、思路不清楚** | |
| **图表应用： □完全正确 □正确 □基本正确 □不正确** | |
| **教师评语** | **教师签字：**  **年 月 日** |

# 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 答辩记录 | | | | |
| 姓名： 学号 | | | | |
| 负责模块： | | | | |
| 运行效果： □流畅、无错误 □正常、无错误  □基本正常、有少量错误 □运行有问题，有错误 | | | | |
| **问题序号** | **问题记录** | **回答记录** | | |
| **完全正确** | **基本正确** | **不**  **正确** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| 备注 |  | | | |
| 答辩分数 |  | | | |