

桂林电子科技大学信息科技学院

课程名称： **JAVA+SQL综合实训**

题目名称：  **学生选课管理系统**

学院名称： 信息工程系

： 姓名： 肖柳泉

学号： 1651200203

指导老师： 刘芳芳

**桂林电子科技大学信息科技学院**

**信息工程系**

**2018年12月25日**

目录

[前言 4](#_Toc533448534)

[1.引 言 5](#_Toc533448535)

[1.1 目的 5](#_Toc533448536)

[1.2背景 5](#_Toc533448537)

[1.3要求和目标 5](#_Toc533448538)

[1.3.1基本要求 6](#_Toc533448539)

[1.3.2目标 6](#_Toc533448540)

[2. 总体设计 6](#_Toc533448541)

[2.1.可行性报告 6](#_Toc533448542)

[2.1.1法律方面的可行性 6](#_Toc533448543)

[2.1.2使用方面的可行性 6](#_Toc533448544)

[2.1.3 结论 7](#_Toc533448545)

[2.2.需求规定 7](#_Toc533448546)

[2.2.1开发环境和软件 7](#_Toc533448547)

[2.2.2系统功能 7](#_Toc533448548)

[2.2.3系统性能 8](#_Toc533448549)

[2.2.4可靠性 8](#_Toc533448550)

[2.2.5灵活性 8](#_Toc533448551)

[2.2.6输入输出要求 8](#_Toc533448552)

[2.3基本数据流程 9](#_Toc533448553)

[2.4功能说明块 11](#_Toc533448554)

[3.MySql数据库 15](#_Toc533448555)

[3.1课程名称 15](#_Toc533448556)

[3.2学生学号+姓名： 16](#_Toc533448557)

[3.3登陆账号及密码 16](#_Toc533448558)

[3.4学生选择情况 16](#_Toc533448559)

[4. 主要代码 17](#_Toc533448560)

[4.1登录界面整体 17](#_Toc533448561)

[4.2选课界面整体 19](#_Toc533448562)

[4.3显示已选课程状态界面整体 21](#_Toc533448563)

[4.4对数据库进行连接 21](#_Toc533448564)

[5.心得体会 23](#_Toc533448565)

[6.参考资料 24](#_Toc533448566)

**摘 要**

随着计算机技术的飞速发展和高等教育体制改革不断的深入，传统的选课手段及工作效率不能适应新的发展需要，无法更好地完成学校2的选课管理工作。提高教务的选课管理水平的主要途径就是不断更新教学需要的系统，增强对学生选课的管理。基于Java+SQL数据库技术建立一个高校学生选课管理系统，该系统为学校的学生和教师提供方便的选课管理系统，基本上满足了学生选课的需求。

关键字：Java、MySQL、选课

前言

学生选课管理是各大高校的主要日常工作之一，涉及到校、系、师的诸多方面，随着教学体制的不断改革，学生选课管理日常工作及保存管理日益繁重、复杂。迫切需要研发设计一款功能强大、操作简单并且具有人性化的学生选课管理系统。

学生选课管理系统，可用于学校等机构的学生选课管理、查询、更新与维护，使用方便，易用性强，并且图形界面清晰明了。该系统软件采用JAVA高级程序语言编写，并用MySql数据库作为后台数据库对信息进行存储、修改、删除等。用ODBC驱动实现前台JAVA与后台SQL数据库的连接。JAVA语言跨平台性强，可以在windows、linux等系统下使用，方便简单，安全性好。

**Eclipse概述**

  Eclipse是以个集成开发环境（Integrated DevelopmentEnvironment）。其前身是IBM的Visual Agefor Java(VA4J)，IBM投入了3千万美元。Eclipse是可扩展的体系结构，可以集成不同软件开发供应商开发的产品，将他们开发的工具和组件加入到Eclipse平台中。随Java应用的广泛，各大主要软件供应商都参与到Eclipse架构开发中，使得Eclipse插件数量与日增加。其中，IBM的WebSphere Studio Workbench是突出的例子。Eclipse平台的免费，架构的成熟，行业协会Eclipse基金会的支持，使得很多的Java开发采用了Eclipse架构。如今,IBM通过起附属的研发机构Object Technologies International(OTI)，继续引领Eclipse开发。其他的Java集成开发环境有Jbuilder、WebSpher Studio、NetBeans。然而Eclipse以其强大的可扩展性和开放性脱颖而出，拥有为数众多的支持者，也被许多大的企业所青睐。

# 1.引 言

## 1.1 目的

通过本次实训程序设计的实践操作，能够让学生懂得Java、SQL的各种相关知识的使用，真正的提高学生独立开发设计Java程序，把课堂上的知识运用在实践上，一门编程语言只有在不断实践操作和练习上才会有进步。

## 1.2背景

在一些学校等机构，随着学生数量的不断增加，学生的课程花样百出，学生选课管理的难度也越来越大。而且效率也是很低的。所以如何自主自动高效地管理学生选课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。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。

## 1.3要求和目标

### 1.3.1基本要求

功能：将该学期的全部课程都整理出来，可实现学生网上选课，方便实现学生选课信息查询，方便学生进行退选课。

## 1.3.2目标

方便学生进行课程的选择和退选不要的课程。

提高选课的工作效率。

降低课程选择的工作量。

# 2. 总体设计

## 2.1.可行性报告

### 2.1.1法律方面的可行性

该系统的开发和研制，将不会侵犯他人、集体和国家的利益，不会违犯国3家政策和法律。

### 2.1.2使用方面的可行性

软件应该保证系统的正常运行，避免系统的漏洞，还要保证数据的准确性，反应速度快，操作简单。

### 2.1.3 结论

综上所述，该项目应立即开始进行研发。从人力资源优化角度来说，可以充分利用人力资源。从经济利益的角度来看，即时开发完成即可用，可以立即赢利。但系统仍旧有些不足，例如修改方面还需人工手动修改数量等。在系统的交付使用中如有不足还望指出，以便改进。

## 2.2.需求规定

### 2.2.1开发环境和软件

2.2.1.1硬件接口

服务器端：

CPU：2.2GHz以上。

内存：1G以上。

客户端：

CPU：P4以上。

内存：1G以上。

2.2.1.2软件接口

（1） 操作系统：Windows 8.1（Windows 7、Windows 10）

（2） 数据库软件：MySQL

（3） Java开发工具：Eclipse

### 2.2.2系统功能

1. 可实现学生网上选课。
2. 方便实现学生选课信息查询。
3. 方便学生进行退课。

### 2.2.3系统性能

使用稳定，操作性能好，操作方法易于掌握。

### 2.2.4可靠性

系统具有较高的可靠性。

### 2.2.5灵活性

系统有良好的可扩展性，允许管理员随时对其进行更新和维护。当用户需求，如操作方式，运行环境，结果精度等发生变化时，设计的软件要做适当调整,灵活性较好。

### 2.2.6输入输出要求

1. 输入

系统在功能区提示输入位置，支持以下两种输入方式：

①鼠标点选：用户点选功能，如该位置是系统允许的有效目标 则记录为用户输入；

②文字输入：用户在指定文本框中输入文字，被系统确认为有 效后记录为用户输入。

1. 输出
2. .基本信息：学生信息包括 学号，姓名，密码。

②.课程信息包括：课号，名称，学分，学时，选择。

## 2.3基本数据流程



图2.1学生选课系统流程图

### 2.4功能说明块

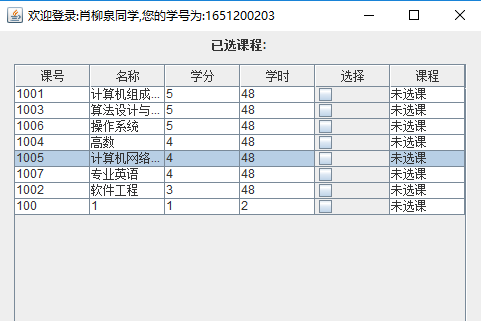
（1）登陆界面

在该界面可以输入学号，登陆密码。用户通过验证后方可进如选课系统，在一定的程度上保证了信息的安全性，防止他人未经允许篡改学生的选课信息。



（2）学生查询选课信息

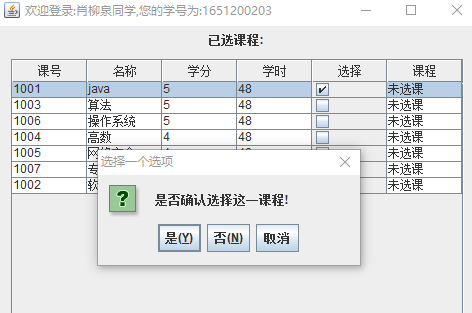
学生登陆选课管理系统后可以查询自己选课的信息，例如已选课程，由于连接了数据库，在点击的时候会弹出数据库里面的可以被选的课程名称

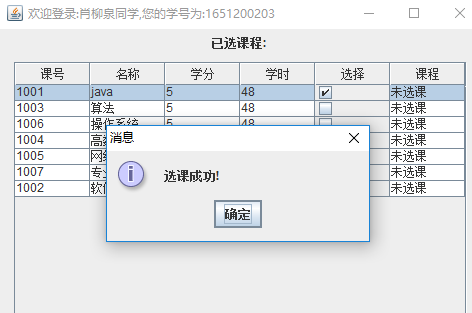




（3）学生课程的选择

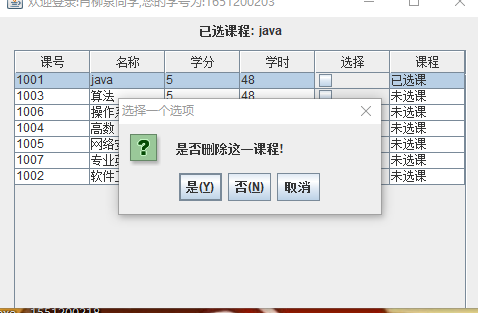
在学生选择一门课程的时候会弹出“是否确认选择这一门课程！”如果确认并且成功的话则会弹出“选课成功！”





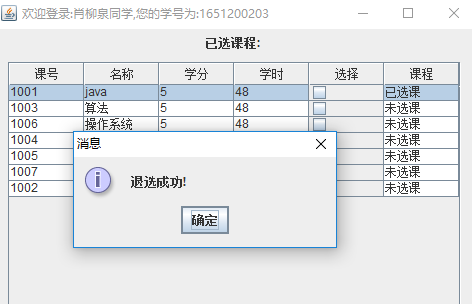
（4）学生删除课程

学生在课程里面还可以删除选择错误的课程，会弹出一句“是否删除这一课程！” 的提示。



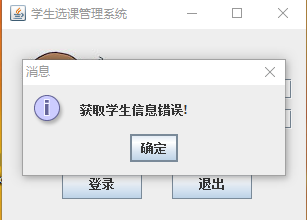
（5）学生退选课程

学生可以在系统中将不要的课程或多选的课程退掉，这样可以让其他还没有选课的同学多一次机会选择。如果学生退课成功的话，则会弹出“退选成功！”的提示。



（6）学生信息

如果出现错误会弹出“获取学生信息错误！”的提示。则说明数据库学生表中没有此学生学号姓名。



（7）课程

学生进入选课界面的时候，会触发控件，连接数据库，这时会将在MySQL数据库提取出数据，将会在界面上跳出在数据库上建立的数据，里面包含了“课号”、“名称”、“学分”、“学时”、“选择”。

# 3.MySql数据库

这个学生选课管理系统一共建立的4个表格。

每一个表格都有各自的作用。

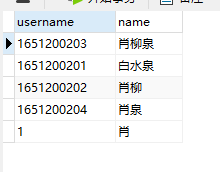
数据库的信息存储是非常有效的。

## 3.1课程名称

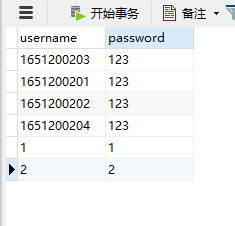
包含了学校的课程和学分和学时：

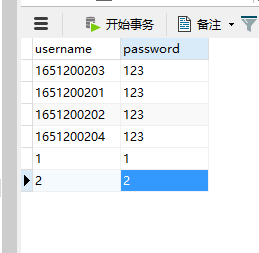


## 3.2学生学号+姓名：



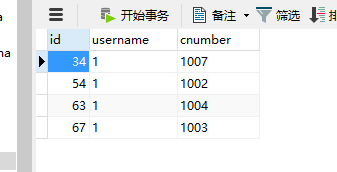
## 3.3登陆账号及密码

包含的学生的账号和密码：



## 3.4学生选择情况

包含了学生选择的课号：



# **4. 主要代码**

## **4.1登录界面整体**

**class** LoginFrame **extends** JFrame {

**private** JLabel userLabel;

**private** JLabel passwordLabel;

**private** JPasswordField passwordField;

**private** JTextField userNameField;

**private** JButton loginButton;

**private** JButton exitButton;

**private** JLabel labelImage;

JFrame frame = **new** JFrame("学生选课管理系统");

// 在窗体中得到一个容器

Container container = frame.getContentPane();

// 构造函数,初始化对象值

**public** LoginFrame() {

// 设置面板的绝对布局

container.setLayout(**null**);

labelImage = **new** JLabel();

labelImage.setIcon(**new** ImageIcon("res/mowang.jpg"));

labelImage.setBounds(10,20,100,100);

//设置用户标签提示信息

userLabel = **new** JLabel("学号:");

userLabel.setBounds(110, 40, 40, 40);

// 用户名文本框

userNameField = **new** JTextField();

userNameField.setBounds(170, 50, 120, 20);

// 密码标签

passwordLabel = **new** JLabel("密码:");

passwordLabel.setBounds(110, 70, 40, 40);

// 密码文本框

passwordField = **new** JPasswordField();

passwordField.setBounds(170, 80, 120, 20);

// 登录按钮

loginButton = **new** JButton("登录");

loginButton.setBounds(60, 140, 80, 30);

loginButton.addActionListener(**new** buttonListenner()); //用addActionListener方法把buttonListenner注册成loginButton的监视器

// 退出按钮

exitButton = **new** JButton("退出");

exitButton.setBounds(170, 140, 80, 30);

exitButton.addActionListener(**new** buttonListenner());

container.add(labelImage);

// 将退出按钮添加到容器

container.add(exitButton);

// 将登录按钮添加到容器

container.add(loginButton);

// 将密码文本框添加到容器

container.add(passwordField);

// 将密码标签添加到容器

container.add(passwordLabel);

// 将用户名文本框添加到容器

container.add(userNameField);

// 把用户标签放到面板里边

container.add(userLabel);

/\* 设置窗体可见和居中 \*/

frame.setVisible(**true**);

frame.setLocationRelativeTo(**null**);

frame.setLocation(800, 400);

frame.setSize(320, 230);

frame.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***); //关闭窗口有什么用

}

## 4.2选课界面整体

**public** MainFrame(String userName, Statement statement) { //构造方法

**super**();

user = userName;

state = statement;

getInformation(userName, statement);

setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***); //

/\* 1.选课信息 \*/

userPanel = **new** JPanel(); // 设置单击窗体右上角的关闭图标

setSize(500, 130);

content = **new** JPanel();

add(content);

content.setLayout(**new** BorderLayout(5, 5));

/\* 将窗体分割成两个部分1.选课信息,2。课程部分 \*/

userPanel.setLayout(**new** FlowLayout());

add(userPanel, BorderLayout.***NORTH***);

userPanel.add(**new** JLabel("已选课程:"));

course = **new** JLabel();

userPanel.add(course);

/\* 2。课程部分 \*/

JPanel coursePanel = **new** JPanel();

add(coursePanel, BorderLayout.***CENTER***);

informationTable = setTable(userName, statement);

informationTable.addMouseListener(**new** MouseAdapter());

JScrollPane courseScroll = **new** JScrollPane(informationTable);

coursePanel.add(courseScroll);

setVisible(**true**);

setLocationRelativeTo(**null**); //设置窗口相对于指定组件的位置

}

**2.6.3添加课程和删除已选课程界面整体**

private void setData(String userName, Statement statement) {

int nRow = informationTable.getSelectedRow();

if ((boolean) informationTable.getValueAt(nRow, 4)) {

if (JOptionPane.YES\_OPTION == JOptionPane.showConfirmDialog(null,

"是否确认选择这一课程!")) {

try {

String sql = "INSERT INTO studentcourse(username,cnumber) VALUE('"

+ userName + "','" + columsStrings[nRow][0] + "')";

statement.executeUpdate(sql); // 返回值，更新记数量

JOptionPane.showMessageDialog(null, "选课成功!");

getStudentCourse(userName, statement);

informationTable.setValueAt(true, nRow, 4);// 单元格中的值设置为 aValue true为选中

informationTable.setValueAt("已选课", nRow, 5);

}

catch (SQLException e) { // 处理异常块

JOptionPane.showMessageDialog(null, "选课错误!");

}

}

else {

informationTable.setValueAt(false, nRow, 4);

}

}

else {

if (JOptionPane.YES\_OPTION == JOptionPane.showConfirmDialog(null,

"是否删除这一课程!")) {

try {

String sql = "DELETE FROM studentcourse WHERE username="

+ userName + " AND cnumber="

+ columsStrings[nRow][0];

statement.executeUpdate(sql);

getStudentCourse(userName, statement);

JOptionPane.showMessageDialog(null, "退选成功!");

informationTable.setValueAt(false, nRow, 4);

informationTable.setValueAt("未选课", nRow, 5);

}

catch (SQLException e) {

JOptionPane.showMessageDialog(null, "退选错误!");

}

}

else {

informationTable.setValueAt(true, nRow, 4);

}

}

}

# 4.3显示已选课程状态界面整体

**if** (flag) {

StringBuffer buffer = **new** StringBuffer();

resultSet.first();

buffer.append(resultSet.getString(1));

**while** (resultSet.next()) {

buffer.append(",");

buffer.append(resultSet.getString(1));

}

course.setText(buffer.toString());

}

**else** {

course.setText("");

}

resultSet.close();

}

}

## 4.4对数据库进行连接

class buttonListenner implements ActionListener {

@Override

public void actionPerformed(ActionEvent e) {

if (exitButton == e.getSource()) {

System.exit(0);

}

if (loginButton == e.getSource()) {

/\* 对数据库的提取验证操作; \*/

try {

Class.forName("com.mysql.jdbc.Driver"); // 注册驱动

String url = "jdbc:mysql://localhost/student?useUnicode=true&characterEncoding=utf8&useSSL=true";//建立li

Connection conection = DriverManager.getConnection(url,"root","123456");

Statement statement = conection.createStatement(); // 发起请求增删改查

String sql = "SELECT \* FROM user"; // 选择用户表

/\* 获取结果集 \*/

ResultSet resultSet = statement.executeQuery(sql); // 执行返回多个结果集以resultset类型返回ex做传递sql信息

while(resultSet.next()){ // 每次一行 成功true

String userName = null;

String password = null;

userName = resultSet.getString(1); // 得到id

password = resultSet.getString(2);// 得到密码

if (userName.equals(userNameField.getText())) {

if (password.equals(new String(passwordField.getPassword()))) {

frame.dispose();

new MainFrame(userName,statement);

}

}

else {

}

}

JOptionPane.showMessageDialog(null, "账号或密码错误!");

resultSet.close();

}catch (SQLException e1) {

}catch(ClassNotFoundException e2){

JOptionPane.showMessageDialog(null, "类错误");

}

}

}

}

}

# 5.心得体会

这次实训是Java+SQL数据库课程的程序设计，Java高级程序应用是是计算机技术专业重要的实践课程之一，SQL数据库也是计算机技术专业重要的实践课程之一。SQL数据库是在掌握程序设计语言的基础上，学习《数据库原理》课程后的一次综合实训练习。

通过本次的程序设计，将在课堂上了解和掌握的Java程序设计语言原理以及设计阶段的方法与技术，直接运用到实际系统的开发工作中。对程序的编写及修改还有数据库的建立和连接。在程序设计初期定义系统的需求，列出了相应的模块，并进行了结构化设计。然后根据初步研究结果，确定编程方向，开始任务。主要通过借阅相关书籍，上网查阅资料、上网看视频教学等方式，完成了我的系统编程初稿。但在系统测试阶段，出现了很多的问题和错误。 比如数据库的应用还不是很熟练，创建数据库的时候会出点错误；还有在数据库文件的导入路径出错，导致数据库连接错误等。由于在MySQL数据库的平台上运行，还需要加载很多插件，所以在安装的时候都花了很多的时间，但是所花的时间都是值得的，特别是看着自己在电脑上创建的数据库一步一步的实现自己要的功能。

这次的实训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. 本次实训的程序设计虽然很辛苦但收获颇多。

通过这次程序设计，我不仅清楚地知道了自己对数据库原理知识和java编程语言方面的不足，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，从而提高自己的实际动手能力和独立思考的能力。

通过这次实训熟悉了软件工具与环境，把课堂学习的理论知识很好的在实践中运用了起来。奋斗虽是艰苦的，但收获是快乐的。这次实训对知道了如何将java和mysql连接，对java编程有了更深的认识

# 6.参考资料

[1] 赵杰，李涛，朱慧 SQL Server数据库管理、设计与实现教程 [M].清华大学出版社，2004

[2] 王晟，马里杰 SQL SERVER数据库开发经典案例解析 [M].清华大学出版社，2006

[3] 肖海蓉、任民宏 数据库原理与应用 [M].北京：清华大学出版社，2016年

[4] 耿祥义，《JAVA课程设计》 [M].北京;清华大学出版社，2016年

[5] 黄晓东，《JAVA课程设计案例精编》 [M]. 北京：中国水利水电出版社，2016年