

**实 训 报 告**

**课程名称：**Java程序设计实训

**题目**：学生信息管理系统设计

**班 级** \_\_\_\_\_\_\_\_ 软件A1803

**学 号** \_\_\_ 0411180310

**姓 名** \_\_\_\_\_\_\_\_\_\_ 谭茜升

**成 绩** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**指导老师** \_\_\_\_\_\_\_\_\_\_朱儒明\_\_\_\_\_\_\_

**2019年1月11日**

1. 实训日期

2018--2019学年度第1学期第18周（2019.1.7—2020-1.11）

1. 实训地点

实验楼F403

适用班级

软件A1803

1. 实训目的

使用myeclipse-jee-neon-1a-win32 开发工具和mysql数据库,通过本综合实训学生全面掌握Java语言中面向对象编程内容，通过本课程实训，学生能够了解 Java 语言特征、常见的 Java 类库以及面向对象程序设计思想，学会利用 Java 语言编写的简单应用程序。

系统。

1. 实训任务

设计简单应用系统（学生成绩系统，人事，工资管理系统等），让学生全面掌java语言语法， java类与对象，集合，I/O，GUI等API。具体要求如下：

1．掌握安装JDK, 使用myeclipse 编写程序。

2. 掌握Java语言java语言语法。

3. 掌握类，接口的编写，使用。

4. 掌握集合接口List,Set,Map以及它们实现类。

5. 掌握输入输出I/O相关类使用

6. 掌握图形用户界面GUI中AWT包和Swing包常用组件的使用。

8. 掌握数据库编程JDBC（数据的查询、增加，修改，删除）

**五.实训内容**

**实训项目：图书管理系统设计**

**5.1 项目分析**

**1、用户需求**

系统作为用户学生学校信息获取录入修改的平台，系统将为用户提供登录，注册，修改密码，修改学生信息，修改院校信息的功能，能够获取学生信息进行汇总保存。

本系统能够大幅度提高学生信息查询修改的效率，实现学生信息录入删除等功能的系统化。

1. 系统实现登录，注册，修改密码功能。
2. 系统实现学校基础信息维护(学校基本信息的增删改查)。
3. 系统实现学生基础信息维护(学生基本信息的增删改查)。
4. 应用GUI技术实现界面的编写。

**2、业务流程分析**

用户需要使用账号与密码通过登陆页面登陆功能选择管理系统。登陆后进入系统主窗体，可进行功能界面选择，可选择学校密码修改、学生信息修改，院系信息修改和退出功能界面，退出系统时返回登陆页面。

**3、系统功能分析**

本系统主要包括登陆管理、学生信息管理、，院校信息管理，

其功能主要有：

1. 学生账号注册，登录前可进行账号注册，然后进入功能界面。
2. 学生密码信息的修改，用户登陆到系统可进行相应的用户密码修改。
3. 对学生信息进行管理，学生信息的增删改查。
4. 对院校信息进行管理，院校信息的增删改查。

系统流程图如图：

院校学生信息管理系统

用户登陆

功能界面选择

院校信息系统

学生信息系统

注册账号系统

退出界面

学生密码系统

**5.2**

数据库名：books

表1：管理员表(表名：users)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段含义 | 类型 | 长度 | 主键 | 可否为空 |
| id | 编号 | 字符型 | 10 | 是 | 否 |
| uname | 账号 | 字符型 | 12 | 否 | 可 |
| password | 密码 | 字符型 | 10 | 否 | 可 |
| account | 账号 | 字符型 | 20 | 否 | 可 |

表2：图书信息表(表名:booktable)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段含义 | 类型 | 长度 | 主键 | 可否为空 |
| id | 编号 | 字符型 | 255 | 是 | 否 |
| school | 学校 | 字符型 | 255 | 否 | 否 |
| academy | 学院 | 字符型 | 255 | 否 | 可 |
| uname | 姓名 | 字符型 | 255 | 否 | 可 |
| sex | 性别 | 字符型 | 255 | 否 | 否 |
| sid | 学号 | 字符型 | 255 | 否 | 否 |
| personId | 身份证号 | 字符型 | 255 | 否 | 否 |
| Classid | 年级 | 字符型 | 255 | 否 | 否 |
| ClassCode | 班级代码 | 字符型 | 255 | 否 | 否 |
| major | 专业 | 字符型 | 255 | 否 | 否 |
| lang | 主修语言 | 字符型 | 255 | 否 | 否 |

表3：学院信息表(表名：department)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段含 义 | 类型 | 长度 | 主键 | 可否为空 |
| id | 编 号 | 字符型 | 10 | 是 | 否 |
| academy | 学院名 称 | 字符型 | 12 | 否 | 否 |
| academycode | 学院代 码 | 字符型 | 10 | 否 | 否 |
| schoolcode | 学校代 码 | 字符型 | 10 | 否 | 否 |
| compuscode | 校区代 码 | 字符型 | 10 | 否 | 否 |

**5.3**

**一、登陆模块:**



**登陆界面的的源代码:**

String account = accountText.getText().trim();//获取账号内容

String passwd = passwordText.~~getText~~().trim();//获取密码内容

**if** (ObjUtil.*isEmpty*(account) || ObjUtil.*isEmpty*(passwd)) {

JOptionPane.*showMessageDialog*(**null**, "注意输入框不能为空");

passwordText.setText("");

accountText.setText("");

**return**;

}

User user = **new** User(**null**, **null**, passwd, account);

Connection conn = **null**;

PreparedStatement ps = **null**;

ResultSet rs = **null**;

**try** {

conn = DbUtil.*getConnection*();

String sql = "select \* from users where account=? ";

ps = conn.prepareStatement(sql);

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

rs = ps.executeQuery();

User queryUser = **null**;

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

queryUser = **new** User();

String username = rs.getString("uname");

String acc = rs.getString("account");

String pass = rs.getString("passwd");

queryUser.setUname(username);

queryUser.setAccount(acc);

queryUser.setPasswd(pass);

}

**if** (queryUser != **null**) {

**if** (queryUser.getPasswd().equals(user.getPasswd())) {

//System.out.print("登陆成功");

JOptionPane.*showMessageDialog*(**null**, "登录成功");

turn t = **new** turn();

t.setVisible(**true**);

}

} **else** {

JOptionPane.*showMessageDialog*(**null**, "查无此人！");

}

} **catch** (SQLException e) {

e.printStackTrace();

} **finally** {

DbUtil.*release*(conn, ps, rs);

}

**this**.dispose();

…..

**主方法**

**public** **static** **void** main(String args[]) {

java.awt.EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

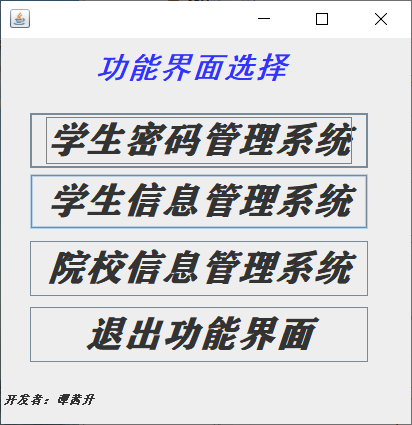
**new** loginUI().setVisible(**true**);

}

});

}

**二、主页面:**



主界面源代码

**private** **void** jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

departmentUI de = **new** departmentUI();

de.setVisible(**true**);

**this**.dispose();

}

**private** **void** jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

loginUI log = **new** loginUI();

log.setVisible(**true**);

**this**.dispose();

}

**private** **void** jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

informationUI inf = **new** informationUI();

inf.setVisible(**true**);

**this**.dispose();

}

**private** **void** jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

changeps change = **new** changeps();

change.setVisible(**true**);

**this**.dispose();

}

主方法

**public** **static** **void** main(String args[]) {

java.awt.EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**new** turn().setVisible(**true**);

}

});

}

**学生密码修改界面**



**修改密码界面代码**

String account = accountText.getText().trim();

String passwd = passwordTxt.~~getText~~().trim();

String passwd2 = passwod2Txt.~~getText~~().trim();

**if** (!(ObjUtil.*isEmpty*(account) || ObjUtil.*isEmpty*(passwd) || ObjUtil

.*isEmpty*(passwd2))) {

**if** (!passwd.equals(passwd2)) {

JOptionPane.*showMessageDialog*(**null**, "两次密码不相同");

accountText.setText("");

passwordTxt.setText("");

passwod2Txt.setText("");

**return**;

}

User user = **new** User(**null**, **null**, passwd, account);

Connection conn = **null**;

PreparedStatement ps = **null**;

String sql = "update users set passwd=? where account=?";

//User user2=new User();

**try** {

conn = DbUtil.*getConnection*();

ps = conn.prepareStatement(sql);

ps.setString(1, passwd);

ps.setString(2, account);

**int** success = ps.executeUpdate();

//判断与原密码是否相同

checkps(account, passwd);

**if** (success > 0) {

JOptionPane.*showMessageDialog*(**null**, "密码修改成功");

turn tu = **new** turn();

tu.setVisible(**true**);

**this**.dispose();

} **else** {

JOptionPane.*showMessageDialog*(**null**, "密码修改失败,请输入正确账号");

**return**;

}

} **catch** (SQLException e) {

e.printStackTrace();

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

e.printStackTrace();

} **finally** {

}

} **else** {

JOptionPane.*showMessageDialog*(**null**, "注意输入框不能为空");

accountText.setText("");

passwordTxt.setText("");

passwod2Txt.setText("");

**return**;

}

**主方法**

**public** **static** **void** main(String args[]) {

java.awt.EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**new** changeps().setVisible(**true**);

}

});

}

**学生信息管理系统**



**学生信息界面源代码**

**private** **void** jButton6ActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

flushtable();

}

**private** **void** jButton5ActionPerformed(java.awt.event.ActionEvent evt) {

**this**.dispose();

turn tu = **new** turn();

tu.setVisible(**true**);

// **TODO** add your handling code here:

}

**private** **void** formMousePressed(java.awt.event.MouseEvent evt) {

// **TODO** add your handling code here:

}

//删除

**private** **void** jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String school = schoolTxt.getText().trim();

String academy = accademyTxt.getText().trim();

String uname = **new** String(unameTxt.getText()).trim();

String sex = **new** String(sexTxt.getText()).trim();

String sid = **new** String(sidTxt.getText()).trim();

String personid = **new** String(personidTxt.getText()).trim();

String classid = **new** String(classidTxt.getText()).trim();

String classcode = **new** String(classcodeTxt.getText()).trim();

String major = **new** String(majorTxt.getText()).trim();

String lang = **new** String(langTxt.getText()).trim();

**if** (ObjUtil.*isEmpty*(sid)) {

JOptionPane.*showMessageDialog*(**null**, "注意sid输入框不能为空");

**return**;

}

Userinfo userInfo = **new** Userinfo(**null**, school, academy, uname, sex,

sid, personid, classid, classcode, major, lang);

Connection conn = **null**;

PreparedStatement ps = **null**;

**try** {

conn = DbUtil.*getConnection*();

String sql = "delete from userinfo where sid=?";

ps = conn.prepareStatement(sql);

ps.setString(1, userInfo.getSid());

**int** success = ps.executeUpdate();

**if** (success > 0)

JOptionPane.*showMessageDialog*(**null**, "删除成功");

//System.out.print("增加成功");

**else**

JOptionPane.*showMessageDialog*(**null**, "删除失败");

} **catch** (Exception e) {

// **TODO**: handle exception

} **finally** {

DbUtil.*release*(conn, ps, **null**);

}

flushtable();

}

//修改

**private** **void** jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String school = schoolTxt.getText().trim();

String academy = accademyTxt.getText().trim();

String uname = **new** String(unameTxt.getText()).trim();

String sex = **new** String(sexTxt.getText()).trim();

String sid = **new** String(sidTxt.getText()).trim();

String personid = **new** String(personidTxt.getText()).trim();

String classid = **new** String(classidTxt.getText()).trim();

String classcode = **new** String(classcodeTxt.getText()).trim();

String major = **new** String(majorTxt.getText()).trim();

String lang = **new** String(langTxt.getText()).trim();

**if** (ObjUtil.*isEmpty*(school) || ObjUtil.*isEmpty*(academy)

|| ObjUtil.*isEmpty*(uname) || ObjUtil.*isEmpty*(sex)

|| ObjUtil.*isEmpty*(sid) || ObjUtil.*isEmpty*(personid)

|| ObjUtil.*isEmpty*(classid) || ObjUtil.*isEmpty*(classcode)

|| ObjUtil.*isEmpty*(major) || ObjUtil.*isEmpty*(lang)) {

JOptionPane.*showMessageDialog*(**null**, "注意输入框不能为空");

**return**;

}

Userinfo userInfo = **new** Userinfo(**null**, school, academy, uname, sex,

sid, personid, classid, classcode, major, lang);

Connection conn = **null**;

PreparedStatement ps = **null**;

**try** {

conn = DbUtil.*getConnection*();

String sql = "update userinfo set school=?, academy=?, uname=?, sex=?, personid=?, classid=?, classcode=?, major=?, lang=? where sid=?";

ps = conn.prepareStatement(sql);

ps.setString(1, userInfo.getSchool());

ps.setString(2, userInfo.getAcademy());

ps.setString(3, userInfo.getUname());

ps.setString(4, userInfo.getSex());

ps.setString(5, userInfo.getPersonid());

ps.setString(6, userInfo.getClassid());

ps.setString(7, userInfo.getClasscode());

ps.setString(8, userInfo.getMajor());

ps.setString(9, userInfo.getLang());

ps.setString(10, userInfo.getSid());

**int** success = ps.executeUpdate();

**if** (success > 0)

JOptionPane.*showMessageDialog*(**null**, "修改成功");

//System.out.print("增加成功");

**else**

JOptionPane.*showMessageDialog*(**null**, "修改失败");

} **catch** (Exception e) {

// **TODO**: handle exception

//e.printStackTrace();

JOptionPane.*showMessageDialog*(**null**, "修改失败,不允许修改学号");

} **finally** {

DbUtil.*release*(conn, ps, **null**);

}

flushtable();

}

//添加

**private** **void** jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String school = schoolTxt.getText().trim();

String academy = accademyTxt.getText().trim();

String uname = **new** String(unameTxt.getText()).trim();

String sex = **new** String(sexTxt.getText()).trim();

String sid = **new** String(sidTxt.getText()).trim();

String personid = **new** String(personidTxt.getText()).trim();

String classid = **new** String(classidTxt.getText()).trim();

String classcode = **new** String(classcodeTxt.getText()).trim();

String major = **new** String(majorTxt.getText()).trim();

String lang = **new** String(langTxt.getText()).trim();

**if** (ObjUtil.*isEmpty*(school) || ObjUtil.*isEmpty*(academy)

|| ObjUtil.*isEmpty*(uname) || ObjUtil.*isEmpty*(sex)

|| ObjUtil.*isEmpty*(sid) || ObjUtil.*isEmpty*(personid)

|| ObjUtil.*isEmpty*(classid) || ObjUtil.*isEmpty*(classcode)

|| ObjUtil.*isEmpty*(major) || ObjUtil.*isEmpty*(lang)) {

**return**;

}

Userinfo userInfo = **new** Userinfo(**null**, school, academy, uname, sex,

sid, personid, classid, classcode, major, lang);

Connection conn = **null**;

PreparedStatement ps = **null**;

**try** {

conn = DbUtil.*getConnection*();

String sql = "insert into userinfo(school, academy, uname, sex,sid, personid, classid, classcode, major, lang) values (?,?,?,?,?,?,?,?,?,?)";

ps = conn.prepareStatement(sql);

ps.setString(1, userInfo.getSchool());

ps.setString(2, userInfo.getAcademy());

ps.setString(3, userInfo.getUname());

ps.setString(4, userInfo.getSex());

ps.setString(5, userInfo.getSid());

ps.setString(6, userInfo.getPersonid());

ps.setString(7, userInfo.getClassid());

ps.setString(8, userInfo.getClasscode());

ps.setString(9, userInfo.getMajor());

ps.setString(10, userInfo.getLang());

**int** success = ps.executeUpdate();

**if** (success > 0) {

JOptionPane.*showMessageDialog*(**null**, "添加成功");

System.*out*.print("增加成功");

} **else**

JOptionPane.*showMessageDialog*(**null**, "添加失败");

} **catch** (Exception e) {

// **TODO**: handle exception

} **finally** {

DbUtil.*release*(conn, ps, **null**);

}

flushtable();

}

**private** **void** stuTableMousePressed(java.awt.event.MouseEvent evt) {

// **TODO** add your handling code here:

**int** school = 0;

**int** academy = 1;

**int** uname = 2;

**int** sex = 3;

**int** sid = 4;

**int** personid = 5;

**int** classid = 6;

**int** classcode = 7;

**int** major = 8;

**int** lang = 9;

//选中在哪行

**int** row = **this**.stuTable.getSelectedRow();

//一个多少列

**int** cols = **this**.stuTable.getColumnCount();

String[] strs = **new** String[10];

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

strs[i] = (String) **this**.stuTable.getValueAt(row, i);

}

Userinfo userInfo = **new** Userinfo(**null**, strs[school], strs[academy],

strs[uname], strs[sex], strs[sid], strs[personid],

strs[classid], strs[classcode], strs[major], strs[lang]);

schoolTxt.setText(userInfo.getSchool());

accademyTxt.setText(userInfo.getAcademy());

unameTxt.setText(userInfo.getUname());

sexTxt.setText(userInfo.getSex());

sidTxt.setText(userInfo.getSid());

personidTxt.setText(userInfo.getPersonid());

classcodeTxt.setText(userInfo.getClasscode());

majorTxt.setText(userInfo.getMajor());

langTxt.setText(userInfo.getLang());

classidTxt.setText(userInfo.getClassid());

}

//查询

**private** **void** jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String cxnm = **this**.cxTxt.getText();

**if** (!ObjUtil.*isEmpty*(cxnm)) {

flushtablecx(cxnm);

} **else** {

JOptionPane.*showMessageDialog*(**null**, "查询失败");

}

}

**private** **void** flushtablecx(String cxnm) {

Connection conn = **null**;

PreparedStatement ps = **null**;

ResultSet rs = **null**;

**try** {

conn = DbUtil.*getConnection*();

String sql = "select \* from userinfo where uname like ? ";

ps = conn.prepareStatement(sql);

ps.setString(1, "%" + cxnm + "%");

rs = ps.executeQuery();

//school academy uname sex sid

//personid classid classcode major lang

ArrayList<Userinfo> users = **new** ArrayList<Userinfo>();

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

String school = rs.getString("school");

String academy = rs.getString("academy");

String uname = rs.getString("uname");

String sex = rs.getString("sex");

String sid = rs.getString("sid");

String personid = rs.getString("personid");

String classid = rs.getString("classid");

String classcode = rs.getString("classcode");

String major = rs.getString("major");

String lang = rs.getString("lang");

Userinfo userInfo = **new** Userinfo(**null**, school, academy, uname,

sex, sid, personid, classid, classcode, major, lang);

users.add(userInfo);

}

//将查询出来的语句,填充到table中

//获得的table对象

DefaultTableModel dtm = (DefaultTableModel) **this**.stuTable

.getModel();

dtm.setRowCount(0);

**if** (users.size() > 0)

**for** (Userinfo userinfo : users) {

Vector vector = **new** Vector();

vector.add(userinfo.getSchool());

vector.add(userinfo.getAcademy());

vector.add(userinfo.getUname());

vector.add(userinfo.getSex());

vector.add(userinfo.getSid());

vector.add(userinfo.getPersonid());

vector.add(userinfo.getClassid());

vector.add(userinfo.getClasscode());

vector.add(userinfo.getMajor());

vector.add(userinfo.getLang());

dtm.addRow(vector);

}

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

DbUtil.*release*(conn, ps, rs);

}

}

//刷新方法

**private** **void** flushtable() {

Connection conn = **null**;

PreparedStatement ps = **null**;

ResultSet rs = **null**;

**try** {

conn = DbUtil.*getConnection*();

String sql = "select \* from userinfo ";

ps = conn.prepareStatement(sql);

rs = ps.executeQuery();

ArrayList<Userinfo> users = **new** ArrayList<Userinfo>();

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

String school = rs.getString("school");

String academy = rs.getString("academy");

String uname = rs.getString("uname");

String sex = rs.getString("sex");

String sid = rs.getString("sid");

String personid = rs.getString("personid");

String classid = rs.getString("classid");

String classcode = rs.getString("classcode");

String major = rs.getString("major");

String lang = rs.getString("lang");

Userinfo userInfo = **new** Userinfo(**null**, school, academy, uname,

sex, sid, personid, classid, classcode, major, lang);

users.add(userInfo);

}

DefaultTableModel dtm = (DefaultTableModel) **this**.stuTable

.getModel();

dtm.setRowCount(0);

**if** (users.size() > 0)

**for** (Userinfo userinfo : users) {

Vector vector = **new** Vector();

vector.add(userinfo.getSchool());

vector.add(userinfo.getAcademy());

vector.add(userinfo.getUname());

vector.add(userinfo.getSex());

vector.add(userinfo.getSid());

vector.add(userinfo.getPersonid());

vector.add(userinfo.getClassid());

vector.add(userinfo.getClasscode());

vector.add(userinfo.getMajor());

vector.add(userinfo.getLang());

dtm.addRow(vector);

}

} **catch** (Exception e) {

e.printStackTrace();

} **finally** {

DbUtil.*release*(conn, ps, rs);

}

}

**主方法**

**public** **static** **void** main(String args[]) {

java.awt.EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**new** informationUI().setVisible(**true**);

}

});

**院校信息管理系统**

**院校信息界面源代码**

//查询

**private** **void** button1ActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String ac = **this**.checkTxt.getText();

**if** (!(ObjUtil.*isEmpty*(ac))) {

cxflushtable(ac);

}

**else**

JOptionPane.*showMessageDialog*(**null**, "查询失败，输入框不能为空");

}

//删除

**private** **void** button4ActionPerformed(java.awt.event.ActionEvent evt) {

String academy = acTxt.getText().trim();

String accode = accodeTxt.getText().trim();

String school = schoolTxt.getText().trim();

String compus = compusTxt.getText().trim();

**if** (ObjUtil.*isEmpty*(academy) || ObjUtil.*isEmpty*(accode)

|| ObjUtil.*isEmpty*(school) || ObjUtil.*isEmpty*(compus)) {

**return**;

}

Connection conn = **null**;

PreparedStatement ps = **null**;

String sql = "delete from department where academy=?";

**try** {

conn = DbUtil.*getConnection*();

ps = conn.prepareStatement(sql);

ps.setString(1, academy);

**int** success = ps.executeUpdate();

**if** (success > 0) {

JOptionPane.*showMessageDialog*(**null**, "删除成功");

} **else**

JOptionPane.*showMessageDialog*(**null**, "删除失败");

} **catch** (SQLException e) {

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

e.printStackTrace();

} **finally** {

DbUtil.*release*(conn, ps, **null**);

}

flushtable();

}

//修改

**private** **void** button3ActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String academy = acTxt.getText().trim();

String accode = accodeTxt.getText().trim();

String school = schoolTxt.getText().trim();

String compus = compusTxt.getText().trim();

**if** (ObjUtil.*isEmpty*(academy) || ObjUtil.*isEmpty*(accode)

|| ObjUtil.*isEmpty*(school) || ObjUtil.*isEmpty*(compus)) {

**return**;

}

Connection conn = **null**;

PreparedStatement ps = **null**;

String sql = "update department set academycode=?,schoolcode=?,compuscode=? where academy=?";

**try** {

conn = DbUtil.*getConnection*();

ps = conn.prepareStatement(sql);

ps.setString(4, academy);

ps.setString(1, accode);

ps.setString(2, school);

ps.setString(3, compus);

**int** success = ps.executeUpdate();

**if** (success > 0) {

JOptionPane.*showMessageDialog*(**null**, "修改成功");

} **else**

JOptionPane.*showMessageDialog*(**null**, "修改失败");

} **catch** (SQLException e) {

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

e.printStackTrace();

} **finally** {

DbUtil.*release*(conn, ps, **null**);

}

flushtable();

}

//添加

**private** **void** button2ActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String academy = acTxt.getText().trim();

String accode = accodeTxt.getText().trim();

String school = schoolTxt.getText().trim();

String compus = compusTxt.getText().trim();

**if** (ObjUtil.*isEmpty*(academy) || ObjUtil.*isEmpty*(accode)

|| ObjUtil.*isEmpty*(school) || ObjUtil.*isEmpty*(compus)) {

**return**;

}

Connection conn = **null**;

PreparedStatement ps = **null**;

String sql = "insert into department (academy,academycode,schoolcode,compuscode) values(?,?,?,?)";

**try** {

conn = DbUtil.*getConnection*();

ps = conn.prepareStatement(sql);

ps.setString(1, academy);

ps.setString(2, accode);

ps.setString(3, school);

ps.setString(4, compus);

**int** success = ps.executeUpdate();

**if** (success > 0) {

JOptionPane.*showMessageDialog*(**null**, "添加成功");

} **else**

JOptionPane.*showMessageDialog*(**null**, "添加失败");

} **catch** (SQLException e) {

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

e.printStackTrace();

} **finally** {

DbUtil.*release*(conn, ps, **null**);

}

flushtable();

}

**private** **void** compusTxtActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

}

//刷新

**private** **void** yxTableMousePressed(java.awt.event.MouseEvent evt) {

// **TODO** add your handling code here:

**int** academy = 0;

**int** academycode = 1;

**int** schoolcode = 2;

**int** compuscode = 3;

//选中在哪行

**int** row = **this**.yxTable.getSelectedRow();

//一个多少列

**int** cols = **this**.yxTable.getColumnCount();

String[] strs = **new** String[4];

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

strs[i] = (String) **this**.yxTable.getValueAt(row, i);

}

department Department = **new** department(**null**, strs[academy],

strs[academycode], strs[schoolcode], strs[compuscode]);

acTxt.setText(Department.getAcademy());

accodeTxt.setText(Department.getAcademycode());

schoolTxt.setText(Department.getSchoolcode());

compusTxt.setText(Department.getCompuscode());

}

**private** **void** flushtable() {

Connection conn = **null**;

PreparedStatement ps = **null**;

ResultSet rs = **null**;

String sql = "select \* from department";

**try** {

conn = DbUtil.*getConnection*();

ps = conn.prepareStatement(sql);

rs = ps.executeQuery();

ArrayList<department> departments = **new** ArrayList<department>();

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

String academy = rs.getString("academy");

String academycode = rs.getString("academycode");

String schoolcode = rs.getString("schoolcode");

String compuscode = rs.getString("compuscode");

department depa = **new** department(**null**, academy, academycode,

schoolcode, compuscode);

departments.add(depa);

}

DefaultTableModel dtm = (DefaultTableModel) **this**.yxTable.getModel();

dtm.setRowCount(0);

**if** (departments.size() > 0) {

**for** (department department : departments) {

Vector vector = **new** Vector();

vector.add(department.getAcademy());

vector.add(department.getAcademycode());

vector.add(department.getSchoolcode());

vector.add(department.getCompuscode());

dtm.addRow(vector);

}

}

} **catch** (SQLException e) {

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

e.printStackTrace();

} **finally** {

DbUtil.*release*(conn, ps, rs);

}

}

**private** **void** cxflushtable(String ac) {

Connection conn = **null**;

PreparedStatement ps = **null**;

ResultSet rs = **null**;

String sql = "select \* from department where academy like ? ";

**try** {

conn = DbUtil.*getConnection*();

ps = conn.prepareStatement(sql);

ps.setString(1, "%" + ac + "%");

rs = ps.executeQuery();

ArrayList<department> departments = **new** ArrayList<department>();

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

String academy = rs.getString("academy");

String academycode = rs.getString("academycode");

String schoolcode = rs.getString("schoolcode");

String compuscode = rs.getString("compuscode");

department depa = **new** department(**null**, academy, academycode,

schoolcode, compuscode);

departments.add(depa);

}

DefaultTableModel dtm = (DefaultTableModel) **this**.yxTable.getModel();

dtm.setRowCount(0);

**if** (departments.size() > 0) {

**for** (department department : departments) {

Vector vector = **new** Vector();

vector.add(department.getAcademy());

vector.add(department.getAcademycode());

vector.add(department.getSchoolcode());

vector.add(department.getCompuscode());

dtm.addRow(vector);

}

}

} **catch** (SQLException e) {

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

e.printStackTrace();

} **finally** {

DbUtil.*release*(conn, ps, rs);

}

}

**主方法**

**public** **static** **void** main(String args[]) {

java.awt.EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

**new** departmentUI().setVisible(**true**);

}

});

}

**退出界面**

**退出主界面返回到登录界面**



**退出界面源代码**

**private** **void** jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

loginUI log = **new** loginUI();

log.setVisible(**true**);

**this**.dispose();

}

**附：**

**数据库连接方法**

**public** **final** **class** DbUtil {

// uri子类 统一资源定位符

**private** **static** String *url* = "jdbc:mysql://127.0.0.1:3306/stus?characterEncoding=UTF-8";

**private** **static** String *user* = "root";// account

**private** **static** String *password* = "";

**static** {

// 01加载驱动

**try** {

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

} **catch** (ClassNotFoundException e) {

e.printStackTrace();

}

}

//获取链接

**public** **static** Connection getConnection()**throws** SQLException {

**return** DriverManager.*getConnection*(*url*,*user*,*password*);

}

//关闭资源

**public** **static** **void** release(Connection conn,Statement statement,ResultSet rs){

**if**(rs!=**null**){

**try** {

rs.close();

} **catch** (SQLException e) {

e.printStackTrace();

}**finally**{

rs=**null**;

}

}

**if**(statement!=**null**){

**try** {

statement.close();

} **catch** (SQLException e) {

e.printStackTrace();

}**finally**{

statement=**null**;

}

}

**if**(conn!=**null**){

**try** {

conn.close();

} **catch** (SQLException e) {

e.printStackTrace();

}**finally**{

conn=**null**;

}

}

}

}

**改变字体，让字体美观代码**

Font font = **new** Font("Dialog", Font.*PLAIN*, 12);

java.util.Enumeration keys = UIManager.*getDefaults*().keys();

**while** (keys.hasMoreElements()) {

Object key = keys.nextElement();

Object value = UIManager.*get*(key);

**if** (value **instanceof** javax.swing.plaf.FontUIResource)

;

UIManager.*put*(font, value);

}

**加入beautytyeye\_lnf.jar包，更改产品界面代码**

**try** {

org.jb2011.lnf.beautyeye.BeautyEyeLNFHelper.*launchBeautyEyeLNF*();

UIManager.*put*("RootPane.setupButtonVisible", **false**);

BeautyEyeLNFHelper.*frameBorderStyle* = BeautyEyeLNFHelper.*frameBorderStyle*.*osLookAndFeelDecorated*;

} **catch** (Exception e) {

// **TODO**: handle exception

}

**让界面居中代码**

**this**.setLocationRelativeTo(**null**);

**创建标准的构造方法**

**有 users 、information 和department这里只说明information的构造方法public** **class** Userinfo {

**private** Integer id;

**private** String school;

**private** String academy;

**private** String uname;

**private** String sex;

**private** String sid;

**private** String personid;

**private** String classid;

**private** String classcode;

**private** String major;

**private** String lang;

**public** Userinfo(Integer id, String school, String academy, String uname,

String sex, String sid, String personid, String classid,

String classcode, String major, String lang) {

**super**();

**this**.id = id;

**this**.school = school;

**this**.academy = academy;

**this**.uname = uname;

**this**.sex = sex;

**this**.sid = sid;

**this**.personid = personid;

**this**.classid = classid;

**this**.classcode = classcode;

**this**.major = major;

**this**.lang = lang;

}

**public** Integer getId() {

**return** id;

}

**public** **void** setId(Integer id) {

**this**.id = id;

}

**public** String getSchool() {

**return** school;

}

**public** **void** setSchool(String school) {

**this**.school = school;

}

**public** String getAcademy() {

**return** academy;

}

**public** **void** setAcademy(String academy) {

**this**.academy = academy;

}

**public** String getUname() {

**return** uname;

}

**public** **void** setUname(String uname) {

**this**.uname = uname;

}

**public** String getSex() {

**return** sex;

}

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

**this**.sex = sex;

}

**public** String getSid() {

**return** sid;

}

**public** **void** setSid(String sid) {

**this**.sid = sid;

}

**public** String getPersonid() {

**return** personid;

}

**public** **void** setPersonid(String personid) {

**this**.personid = personid;

}

**public** String getClassid() {

**return** classid;

}

**public** **void** setClassid(String classid) {

**this**.classid = classid;

}

**public** String getClasscode() {

**return** classcode;

}

**public** **void** setClasscode(String classcode) {

**this**.classcode = classcode;

}

**public** String getMajor() {

**return** major;

}

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

**this**.major = major;

}

**public** String getLang() {

**return** lang;

}

**public** **void** setLang(String lang) {

**this**.lang = lang;

}

**public** Userinfo() {

**super**();

}

@Override

**public** String toString() {

**return** "Userinfo [academy=" + academy + ", classcode=" + classcode

+ ", classid=" + classid + ", id=" + id + ", lang=" + lang

+ ", major=" + major + ", personid=" + personid + ", school="

+ school + ", sex=" + sex + ", sid=" + sid + ", uname=" + uname

+ "]";

}

}

**创建JCombobox组合下拉框代码**

**private** **void** bjcActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String item = (String) bjc.getSelectedItem();

**if** ("请选择班级代码".equals(item)) {

classcodeTxt.setText("");

} **else** {

classcodeTxt.setText(item);

}

}

**private** **void** zycActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String item = (String) zyc.getSelectedItem();

**if** ("请选择专业".equals(item)) {

majorTxt.setText("");

} **else** {

majorTxt.setText(item);

}

}

**private** **void** njcActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String item = (String) njc.getSelectedItem();

**if** ("请选择年级".equals(item)) {

classidTxt.setText("");

} **else** {

classidTxt.setText(item);

}

}

**void** xxcActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String item = (String) xxc.getSelectedItem();

**if** ("请选择学校".equals(item)) {

schoolTxt.setText("");

} **else** {

schoolTxt.setText(item);

}

}

**private** **void** xbcActionPerformed(java.awt.event.ActionEvent evt) {

// **TODO** add your handling code here:

String item = (String) xbc.getSelectedItem();

**if** ("请选择性别".equals(item)) {

sexTxt.setText("");

} **else** {

sexTxt.setText(item);

}

}

**private** **void** xcActionPerformed(java.awt.event.ActionEvent evt) {

String item = (String) xc.getSelectedItem();

**if** ("请选择学院".equals(item)) {

accademyTxt.setText("");

} **else** {

accademyTxt.setText(item);

}

}

.

.

**六．实训总结**

　　这个星期，我们迎来了java语言实训——一次至为重要的实训。在这个星期里，同学们都很认真的做着那道实训题目，遇到不懂的互相请教，或请问老师。非常感谢老师对我们的帮助，这次实训使我受益匪浅，收获了很多。 通过java语言实训，让我加深了对java语言的了解，而不只是单单的在课本中学到的那些理论，平时乏味的课程，通过自己动手亲自编写，变的生动有趣，而在自己动手的过程中，出现的问题很多，比理论要难的多，当一个程序写完以后，经常会有很多错误而没法解决。不过，通过几天的实训，逐渐积攒了一些经验，有些错误可以很快就看出来。

在这个实现间，终于体会到编程思想和逻辑思维的重要性，同一目的，可能达到的方法很多，是我们更希望代码逻辑清晰、可扩展性强。,并且在在进行条件判断时，逻辑思维一定要清晰，不仅想到要判断什么条件，而且要了解其内部的先后顺序。就如在写修改账号密码期间，这问题看似简单，实则其中的判断逻辑及复杂，判断账号存在、密码与原密码是否相同、两次输入密码是否相同等其他判断，然后扩展到循环，嵌套，条理很清楚，不是一个零散的知识，实际上所有的课程都如此，不过通过实训我也知道了自己的不足，在循环嵌套和条件判断需要有待提高。

判断一个程序的优秀，不仅要看功能齐全完善，而java的三大特性是继承、封装和多态，代码写完要进行封装，这样让代码更有条理，易懂。

还有对变量名的定义，虽然有小驼峰和大驼峰，但设置变量名要定义具有可读性，为后面的调用的对象设置变量名。也具有可读性，后面使用对象一个改变此标识符的每个单词的大小写。

老师在教学对学生信息的增删改后，后面也为我们布置对学生信息的查询，和优化界面添加combox的下拉菜单，自己课下找资源学习，锻炼其自学能力，明白以后步入社会对立自学学习的重要性

　最后，非常感谢老师给我们这次实训的机会。极高的锻炼了我们的逻辑思维能力。人非生而知之，虽然我们现在的知识结构还很差，但是我们知道要学的知识，一靠努力学习，二靠潜心实践。没有实践，学习就是无源之水，无本之木。在剩下的时间里，我们会更加努力的学习。