**延安大学数计学院**

**《面向对象程序设计与JAVA》课程设计报告**

( 2021-- 2022年度第1学期)

题 目： 员工工资管理系统

班 级： 软件20

组长学号姓名： 1110520014034马帅

组员学号姓名： 1110520014004和姚晔

组员学号姓名： 1110520014019于纯洋

指 导 教 师： 李富星

设 计 周 数： 3

2021年12月8日

**面向对象程序设计与JAVA课程设计项目任务书**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **组长** | **马帅** | **组员** | **和姚晔 于纯洋** | | | |
| **项目名称** | **员工工资管理系统** | | | | | |
| **专业** | **软件工程** | **班级** | **软件20** | | **指导教师** | **李富星** |
| **项目功能、性能需求（可另附材料）**  设计目的：  （1）掌握JavaSwing的布局设计以及事件处理  （2）掌握MySQL基本操作和利用JDBC技术操作数据库  （3）体会开发流程掌握模块设计方法  设计要求：  利用数据库存储数据，显示可视化窗口，通过JDBC对数据库数据进行增删改查等基本操作  项目功能：  可视化窗口实现存储员工基本信息以及工资，并能对已有数据进行增删改查  性能需求：  数据库内存储各类详细数据，并对已有数据进行批量删减 | | | | | | |
| **团队成员分工情况：**  马帅：程序架构、数据库设计  于纯洋：程序测试、需求分析、PPT  和姚晔：系统功能模块设计、团队日志记录  **团队成员签字：**  **日 期：** | | | | | | |
| **指导教师意见**  **签 字：**  **日 期：** | | | | | | |
| **课程负责人意见**  **签字：**  **日期：** | | | | **专业负责人意见**  **签字： 日期：** | | |

**面向对象程序设计与JAVA课程设计项目完成情况**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **组长** | **马帅** | **组员** | **和姚晔 于纯洋** | | | |
| **项目名称** | **员工工资管理系统** | | | | | |
| **专业** | **软件工程** | **班级** | **软件20** | | **指导教师** | **李富星** |
| **开发环境、进度安排、代码行数等**  开发环境：IDEA  代码行数：1733  开发周期：3周  进度安排：  12月8日-12月9日：进行需求分析和整体规划，确定开发方式，以及基本功能  12月10日-12月17日：确定实现模块以及系统详细功能  12月17日-12月19日：项目收尾，报告撰写，项目交付 | | | | | | |
| **团队成员分工与完成情况**  **成员分工：**  马帅：程序架构、数据库设计  于纯洋：程序测试、需求分析、PPT  和姚晔：系统功能模块设计、团队日志记录  **完成情况：**完成了实验题目的基本要求，在此基础上还增加了附加的功能模块，对Swing显示的窗口进行简单的布局优化，使用数据库通过JDBC进行操作数据，程序整体以面向对象为核心思想，并在此基础上进行实现，体现了基本的封装、继承、多态思想。（具体见实验报告）  **团队成员签字：**  **日期：** | | | | | | |
| **指导教师意见**  **签 字：**  **日 期：** | | | | | | |
| **课程负责人意见**  **签字：**  **日期：** | | | | **专业负责人意见**  **签字： 日期：** | | |

**员工工资管理系统的设计与实现**

**摘要：**设计报告论述了分析，设计，员工工资管理系统的过程。该系统通过数据库存储与JDBC技术操作相结合的方式进行数据的基本操作，通过Swing图形化界面显示详细功能，利用IDEA 开发而成。

设计报告介绍了课程设计的相关内容，并通过功能分析，将系统划分为三个基本数据模型和六个功能模块，最终在面向对象的基础上配合数据库实现数据的增删改查等基本功能。主要通过Swing窗体，内部控件以及监听事件来实现。

**关键词**：IDEA；MySQL；Swing；面向对象。

目录

[1项目开发技术背景 1](#_Toc5190)

[1.1开发技术背景 1](#_Toc32245)

[1.2开发目的和意义 1](#_Toc13222)

[1.3可行性分析 1](#_Toc26950)

[2 系统功能设计 2](#_Toc30589)

[2.1 设计目的 2](#_Toc26114)

[2.2模块功能需求分析 2](#_Toc7074)

[2.3开发流程图 3](#_Toc20777)

[2.4 系统功能模块结构 3](#_Toc24464)

[2.5 用例图 4](#_Toc20697)

[2.6 流程图 5](#_Toc27446)

[3 数据库设计 6](#_Toc18019)

[3.1 数据库E-R图 6](#_Toc5945)

[3.2 数据表设计 6](#_Toc6674)

[4系统功能技术实现 8](#_Toc7206)

[4.1 设计思想 8](#_Toc30206)

[4.2 模块实现描述 8](#_Toc2159)

[5系统运行与测试 15](#_Toc28833)

[5.1测试方法 15](#_Toc6957)

[5.2测试用例 15](#_Toc27923)

[5.3测试环境 16](#_Toc20978)

[5.4测试结论 16](#_Toc26128)

[6 开发常见问题与解决及功能拓展 17](#_Toc10806)

[6.1常见问题 17](#_Toc16946)

[6.2解决办法 17](#_Toc23332)

[6.3功能拓展 17](#_Toc780)

[7 课程设计总结 18](#_Toc13007)

[8 参考文献 19](#_Toc28164)

[附录 20](#_Toc23066)

# 1项目开发技术背景

## **1.1开发技术背景**

随着信息时代的到来，中小型企业的生存和竞争环境发生了根本性的变化。当前中小型企业信息化具有宽广和深刻的内容，其中管理信息化是一个重要方面。如何运用信息技术增强企业的管理，如何制定企业信息化发展战略来提升企业的核心竞争力，如何把信息化系统融入日常的管理工作为企业带来效益，是当前所面临的重要问题。

在这个系统中综合应用了MySQL、Swing等知识。程序界面的结构设计以实用性出发，具有易于操作、简洁、方便等特点。在设计中，首先，运用Swing显示窗体进行基本布局美化。其次，对于Java编程、Swing编程以及MySQL数据库进行努力学习和大量实践，并运用到了程序的建设中。

## **1.2开发目的和意义**

作为一个管理系统，首先布局一定要新颖、有特色，只有这样才能引起用户的关注，最大限度的满足人们的需求，而且要有很强的易用性，易用性差的管理系统会让用户体验大大降低。当然一个好的管理系统还要有很完整的处理信息的功能。通过对用户的调查和对现有的企业财务系统运行的流程分析，企业财务管理系统满足了大多数用户的需求。

## **1.3可行性分析**

（1）经济可行性

本系统对计算机配置的要求不高，低配置电脑完全可以使用，满足需要，所以在经济上具有完全的可行性。

（2）操作可行性

本系统操作简单，输入信息界面大多数都是输入或者下拉框的选择形式，在工号方面，信息可以自动生成，无需输入，操作简便，对操作人员的要求很低，只需对Windows操作熟练即可使用，而且本系统可视性非常好，所以在技术上不会有很大难度。

（3）技术可行性

本系统开发工具是IDEA和MySQL数据库，开发语言是Java，主要使用了JavaSE和Swing的技术，Java是一种面向对象编程语言，简单易学而且灵活方便。员工工资管理系统总体上开发难度不高，数据库的设计和操作是本系统设计的核心。因此，完成系统实现在技术上完全具有可行性。

# **2 系统功能设计**

## **2.1 设计目的**

（1）体验面向对象的编程思想，了解封装，继承，多态的基本思想。

（2）熟练运用JDBC操作MySQL数据库，掌握数据库增删改查基本操作。

（3）体会开发流程，明白各个开发环节的意义以及作用，理解MVC框架模式。

## **2.2模块功能需求分析**

（1）员工工资管理系统解决的是基于Swing的小型企业员工工资管理系统，包括了员工的基本信息的添加和管理以及对部门信息的管理，对员工的工资管理、设置以及查询公司的成员工资比例的基本显示。

## **2.3开发流程图**

本次开发选定开发模型为螺旋模型，进行螺旋式开发，根据所需功能逐步设计完善，依据螺旋模型架构项目最终完成设计。根据每一层次的变化不断修订结构，对必要阶段的功能重复测试确保最终程序的健壮性。

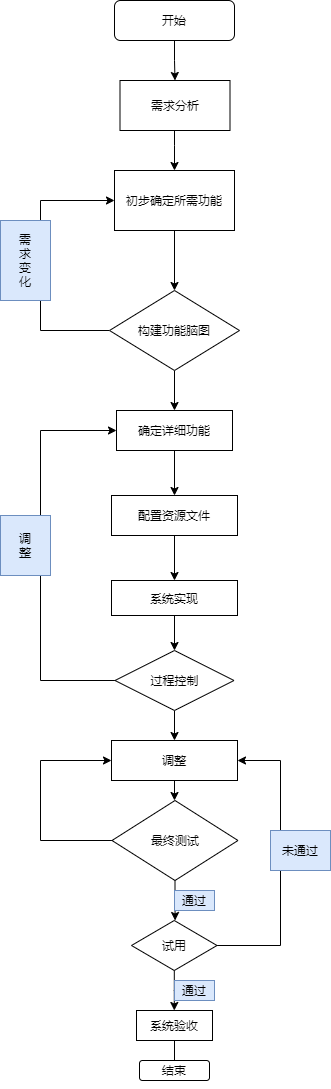


图2.1开发流程图

## **2.4 系统功能模块结构**

根据实际需求和整体设计，可以将系统功能分为以下模块：

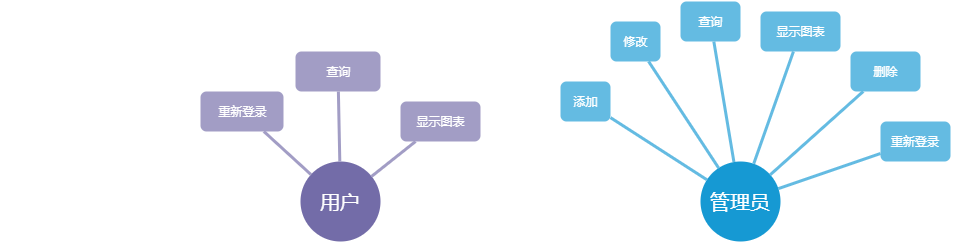
****

图2.2 系统功能结构图

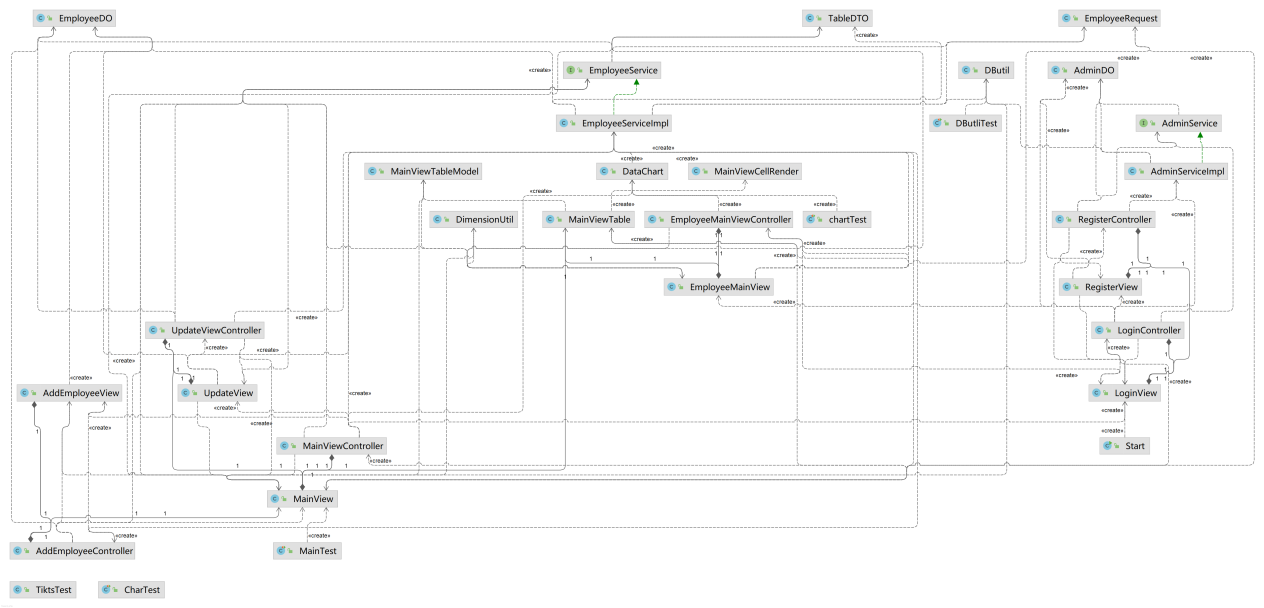
通过在需求阶段对系统的总体功能的要求，管理员包括以下几大全部功能:

1. 添加员工基本信息
2. 删除员工在数据库中的记录
3. 查找员工在数据库中的记录
4. 修改员工在数据库中的记录
5. 显示部门工资比例图表
6. 用户在此基础上删去了对员工信息直接操作的功能

## **2.5 用例图**



## **2.6 流程图**

****

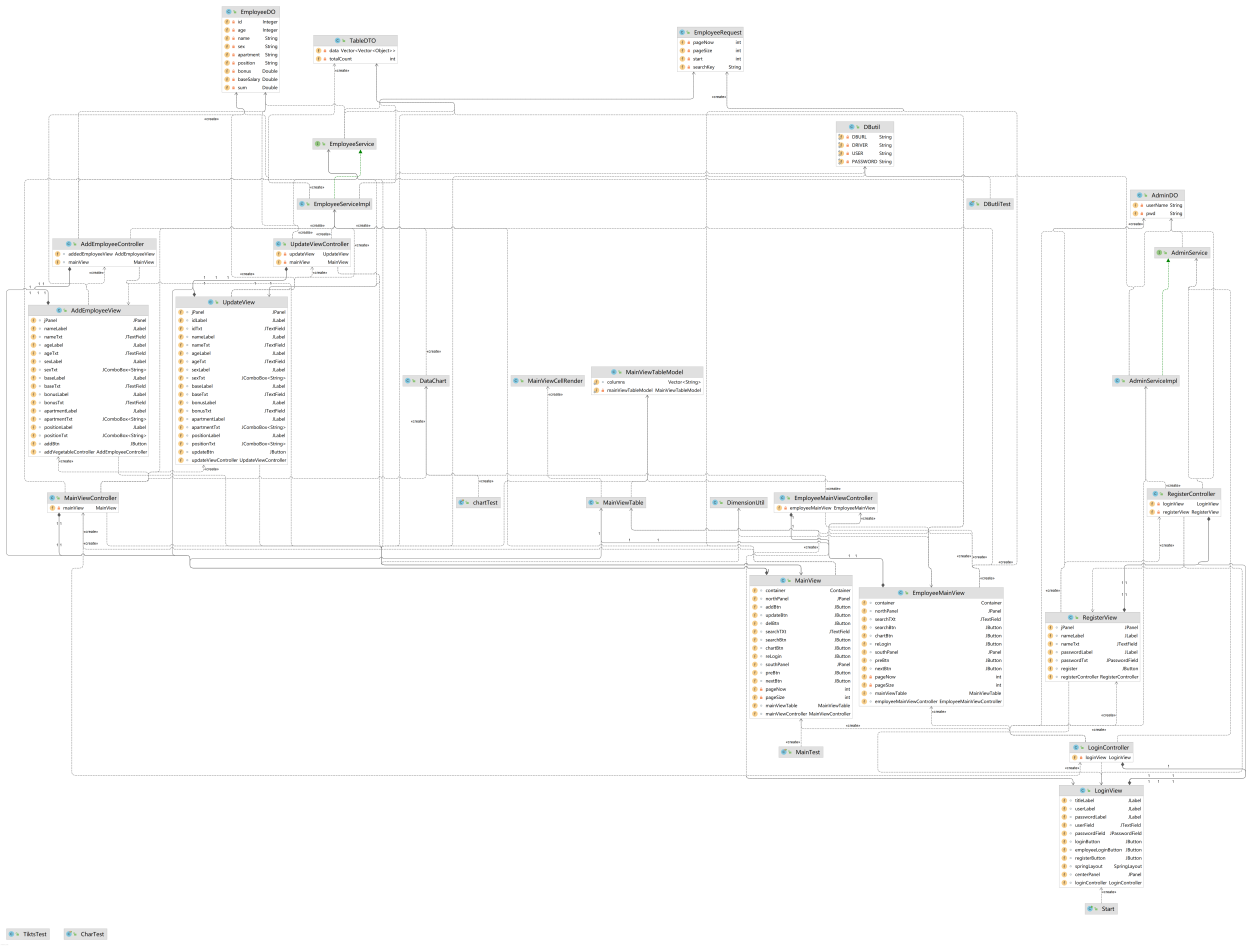
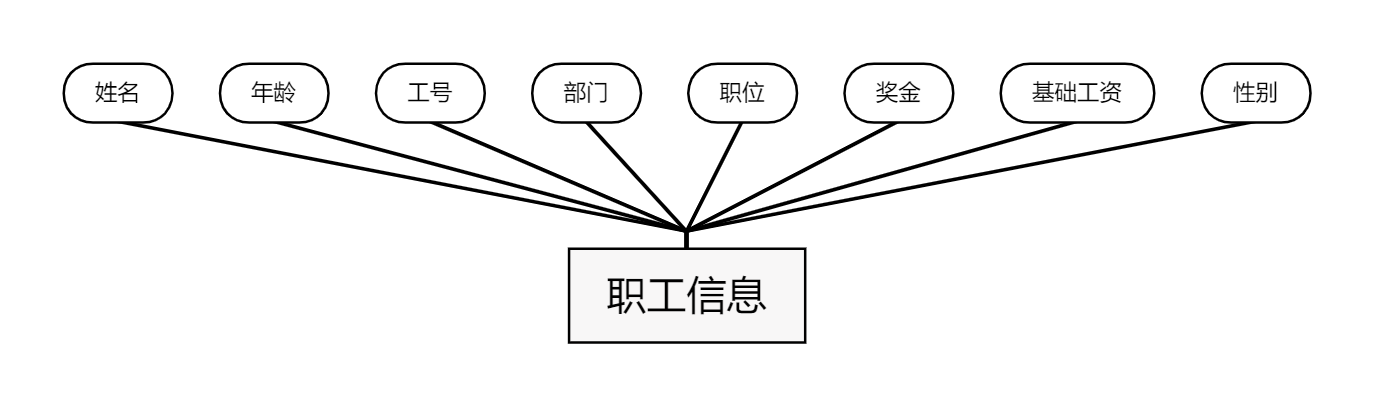


图4.2程序内部结构图

**3 数据库设计**

## **3.1 数据库E-R图**

描述：针对数据库和实体类关系，特设计数据库为一下结构，数据库E-R图如下所示



数据库E-R图

## **3.2 数据表设计**

（1）员工信息表主要记录员工基本信息和工资信息等，如表2.3.1所示

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 是否主键 | 说明 |
| id | int | 11 | 否 | 是 | 编号 |
| name | varchar | 254 | 否 | 否 | 姓名 |
| age | int | 11 | 否 | 否 | 年龄 |
| sex | varchar | 254 | 否 | 否 | 性别 |
| base\_salary | double |  | 否 | 否 | 基本工资 |
| bonus | double |  | 否 | 否 | 奖金 |
| sum | double |  | 否 | 否 | 总工资 |
| apartment | varchar | 254 | 否 | 否 | 部门 |
| position | varchar | 254 | 否 | 否 | 职位 |

表3.3.1员工信息表结构

（2）管理员信息表主要记录管理员登录账号和密码，如表2.3.2示

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 是否主键 | 说明 |
| user | varchar | 254 | 否 | 否 | 账号 |
| password | varchar | 254 | 否 | 否 | 密码 |

表3.3.2管理员信息表结构

（3）用户信息表主要记录用户登录账号和密码，如表2.3.3所示

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 允许空 | 是否主键 | 说明 |
| e\_user | varchar | 254 | 否 | 否 | 账号 |
| e\_password | varchar | 254 | 否 | 否 | 密码 |

表3.3.3用户信息表结构

**4系统功能技术实现**

## **4.1 设计思想**

（1）将员工类，用户类，管理员类封装

（2）通过Swing可视化的方式使用户达到操作数据库的目的

（3）对账户类型进行分类

## **4.2 模块实现描述**

（1）设计注册登录模块

描述：每个用户都需要相应的账号和密码进行登录。这样，不会因为随意登录操作系统而造成系统混乱。在系统设计之初，给出了管理员账号。用户使用时可以自行注册用户账号。

主要核心代码：

public void actionPerformed(ActionEvent e) {

JButton jButton= (JButton) e.getSource();

String text=jButton.getText();

if("注册".equals(text)){

//register();

new RegisterView(loginView);

}else if("管理登录".equals(text)){

login();

// new EmployeeMainView();

//System.out.println("登录成功");

}else if("用户登录".equals(text)){

employeelogin();

}

}

private void login() {

String user= loginView.getUserField().getText();

char[]chars=loginView.getPasswordField().getPassword();

if(user==null||"".equals(user.trim())||chars==null||chars.length==0){

JOptionPane.showMessageDialog(loginView,"用户名密码必填");

return;

}

实现截图：



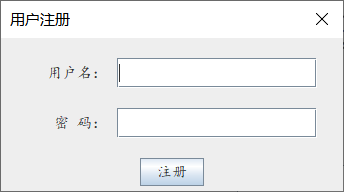


图4.3.1注册登录

（2）设计主界面显示窗口

描述：主界面数据需要与数据库数据进行交互，为了清晰的将数据库表内容展现出来，特利用表格的方式展现数据，可通过鼠标对数据进行选中然后通过上方操作按钮进行控制。

主要核心代码：

public MainView() {

super("员工工资管理系统");

//全屏情况下窗体大小

Rectangle bounds= DimensionUtil.getBounds();

pageSize=Math.floorDiv(bounds.height,35);

//北面布局

layoutNorth();

//中心布局

layoutCenter();

//南面布局

layoutSouth();

setIconImage(new ImageIcon(".\\src\\com\\msskx\\java\\res\\icon.png").getImage());

setBounds(bounds);

setExtendedState(MAXIMIZED\_BOTH);//默认最大化

setLocationRelativeTo(null);//默认居中

setDefaultCloseOperation(EXIT\_ON\_CLOSE);//退出销毁窗体

setResizable(true);

setVisible(true);

}

实现截图：

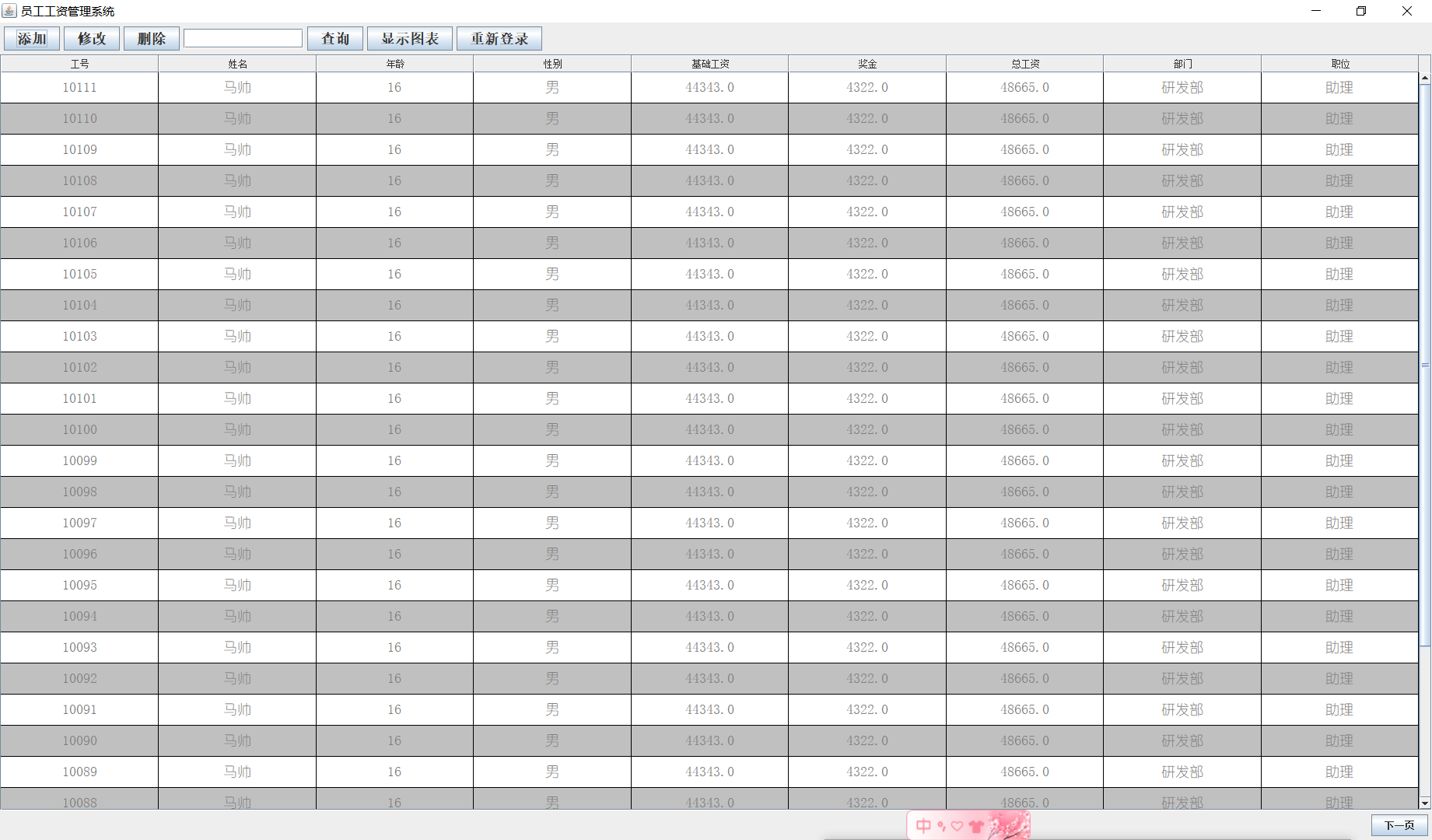


图4.2.2主界面显示窗口

（3）设计主界面功能模块

描述：删除可通过鼠标选中单行或者多行直接删除，添加和修改功能通过弹出对话框的形式，对用户操作进行接收，显示图表将数据库内同类数据的总和进行统计最终绘图显示弹窗。

主要核心代码：

public void actionPerformed(ActionEvent e) {

JButton jButton= (JButton) e.getSource();

String text=jButton.getText();

if("添加".equals(text)){

new AddEmployeeView(mainView);

}else if("修改".equals(text)){

int[]selestedVegetableIds=mainView.getSelectedStudentIds();

if(selestedVegetableIds.length!=1){

JOptionPane.showMessageDialog(mainView,"一次只能修改一行！");

return;

}

new UpdateView(mainView,selestedVegetableIds[0]);

}else if("删除".equals(text)){

int[]selectedVegetableIds=mainView.getSelectedStudentIds();

if(selectedVegetableIds.length==0){

JOptionPane.showMessageDialog(mainView,"请选择要删除的行");

return;

}

int option=JOptionPane.showConfirmDialog(mainView,"你确定要删除选择的"+selectedVegetableIds.length

+"行吗？","确认删除",JOptionPane.YES\_NO\_OPTION);

if(option==JOptionPane.YES\_OPTION){

//执行删除操作

EmployeeService employeeService=new EmployeeServiceImpl();

boolean deleteResult=employeeService.delete(selectedVegetableIds);

if(deleteResult){

mainView.reloadTable();

}else{

JOptionPane.showMessageDialog(mainView,"删除失败");

}

}

}else if("查询".equals(text)){

mainView.setPageNow(1);

mainView.reloadTable();

}else if("上一页".equals(text)){

mainView.setPageNow(mainView.getPageNow()-1);

mainView.reloadTable();

}else if("下一页".equals(text)){

mainView.setPageNow(mainView.getPageNow()+1);

mainView.reloadTable();

}else if("重新登录".equals(text)){

mainView.dispose();

new LoginView();

}else if("显示图表".equals(text)){

new DataChart();

}

}

实现截图：



图4.2.3添加修改功能

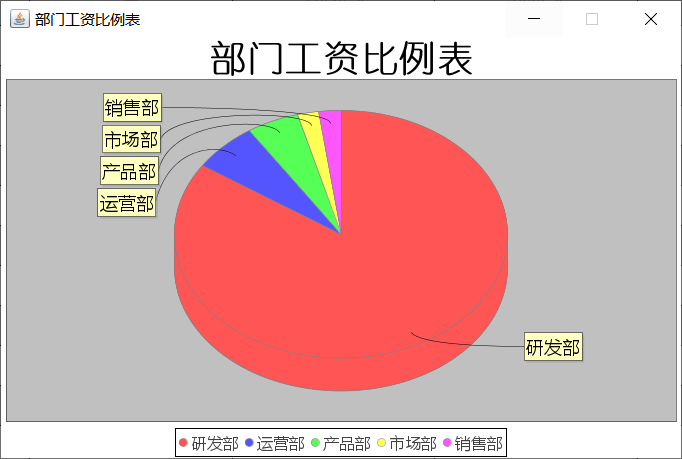


图4.2.4图表显示

**5系统运行与测试**

## **5.1测试方法**

白盒测试：又称结构测试、透明盒测试、[逻辑驱动测试](https://baike.baidu.com/item/%E9%80%BB%E8%BE%91%E9%A9%B1%E5%8A%A8%E6%B5%8B%E8%AF%95/11190159" \t "https://baike.baidu.com/item/%E7%99%BD%E7%9B%92%E6%B5%8B%E8%AF%95/_blank)或基于[代码](https://baike.baidu.com/item/%E4%BB%A3%E7%A0%81" \t "https://baike.baidu.com/item/%E7%99%BD%E7%9B%92%E6%B5%8B%E8%AF%95/_blank)的测试。白盒测试是一种[测试用例设计](https://baike.baidu.com/item/%E6%B5%8B%E8%AF%95%E7%94%A8%E4%BE%8B%E8%AE%BE%E8%AE%A1/9391582" \t "https://baike.baidu.com/item/%E7%99%BD%E7%9B%92%E6%B5%8B%E8%AF%95/_blank)方法，盒子指的是被测试的[软件](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6/12053" \t "https://baike.baidu.com/item/%E7%99%BD%E7%9B%92%E6%B5%8B%E8%AF%95/_blank)，白盒指的是盒子是可视的，即清楚盒子内部的东西以及里面是如何运作的。

对该程序使用代码检查法和逻辑覆盖法，通过覆盖检测部分特殊语句进行测试。

## **5.2测试用例**

### 5.2.1注册功能测试

|  |  |
| --- | --- |
| 软件版本1.0 | 操作系统Windows10 |
| 测试人员姓名：马帅 | 测试日期： |
| 初始设置 |  |
| （1）登录窗口点击注册按钮  （2）注册对话框点击注册按钮 |  |
| 输入 |  |
| （1）在用户名输入user  （2）密码字段输入123 | |
| 预期结果  弹出注册成功消息框 | |
| 实际结果 | ☑通过 □失败 |
|  |  |

表5.2.1注册功能测试

### **5.2.2登录功能测试**

|  |  |
| --- | --- |
| 软件版本1.0 | 操作系统Windows10 |
| 测试人员姓名：马帅 | 测试日期： |
| 初始设置 |  |
| （1）登录窗口点击两个登录按钮  （2）登录对话框点击登录按钮 |  |
| 输入 |  |
| （1）在用户名输入管理员账号admin密码123456 | |
| （2）在用户名输入用户账号user密码123 | |
| 预期结果 |  |
| 跳转到管理员主窗口和用户 |  |
| 实际结果 | ☑通过 □失败 |
|  |  |

表5.2.2登录功能测试

### **5.2.3模块功能测试**

|  |  |
| --- | --- |
| 软件版本1.0 | 操作系统Windows10 |
| 测试人员姓名：马帅 | 测试日期： |
| 初始设置 | |
| （1）确保在登录状态下  （2）确保远程数据库连接正常  （3）点击添加按钮  （4）选中表格一列后点击修改按钮  （5）选中表格一列或者多列后点击删除按钮  （6）点击显示图表按钮  （7）点击重新登录按钮 | |
| 输入 |  |
| （1）输入员工的基本信息   1. 修改员工的姓名为“马帅”，年龄为18，其余不变 2. 鼠标点击选中员工 | |
| 预期结果 |  |
| 1. 添加成功并刷新主页面表格内容 2. 员工姓名被修改为马帅，年龄为18，其余不变 3. 选中行被成功删除 4. 显示图表窗口   （5）主页面销毁，登录页面重新显示 | |
| 实际结果 | ☑通过 □失败 |
|  |  |

表5.2.3模块功能测试

## **5.3测试环境**

Windows 10 64位

## **5.4测试结论**

（1） 经过多轮测试，系统功能已趋于稳定，且项目确定的范围、功能和计划均已实现，项目测试可以结束、员工工资管理系统可以上线。

（2）通过测试觉得产品在用户体验方面有待后续版本进一步改进，不排除用户在使用程序时出现突发状况。

**6 开发常见问题与解决及功能拓展**

## **6.1常见问题**

Swing窗体布局时控件无法精确定位

添加员工ID数据写入数据库空指针异常数据为null报错

控件监听事件操作无效果

## **6.2解决办法**

登录界面布局时采用弹簧布局的方式

设计数据库表ID属性非null,且能自动递增

实现监听接口重写监听执行方法时检查具体操作并作出修改

## **6.3功能拓展**

为实现数据可视化特拓展显示图表功能，为实现在不同账户之间的切换拓展重新登录功能。还可添加登录状态判断，将锁定操作数据，避免数据多线程操作异常。

**7 课程设计总结**

员工工资管理系统的基本功能已经实现。虽然时间很短暂，但却是对一学期学习中理论知识和实践相结合的一次综合检验。通过这次课程设计，收到了比以往理论课程还要大的收益，虽然由于时间仓促及小组的能力有限，系统还有很多不尽人意的地方：比如说界面不够美观；有些功能还不够完善和强大；代码的重用性不够高；一些细节的问题还没有解决。这些都需要平时经验的积累和对技术的熟练掌握，希望在以后的工作学习中能有进一步的提高。

通过小组之间的沟通交流，协同开发通过此次程序设计，小组成员更加深刻地理解的面向对象的编程思想，同时成员的架构能力与编程思维也得到了提高，通过优化编码方式，提高程序的运行效率，同时也提高代码的可读性。此次开发采用螺旋式开发方式，逐步进行，每个功能模块环环相扣，通过一个功能调用另一个功能，功能之间的紧密结合，训练了小组成员的思维能力和团队协作能力。

由于现有水平和经验有限，系统在设计过程中，难免存在不足之处，使得本系统方案设计仍存在许多有待完善和改进的地方，比如说，在程序结构上系统设计得还不够紧密，对一些问题。其次，安全性上考虑不够仔细等等。总的说来，本系统还不成熟，在某些细节方面可能还没有考虑到，功能不是很强大，还不能提供决策性的分析功能，对业务流程的控制不是很灵活。但对它的改善和功能的完备并不会随着毕业设计的结束而终止。后续的工作还需付出更大的努力去完善，将在原来的基础上，着眼未来，追踪计算机新技术发展，并应用到系统持续开发中，不断刻苦钻研，不断提高开发水平，对系统进一步优化。

# **8 参考文献**

1. Java课程设计（第3版） 耿祥义 张跃平 编著 清华大学出版社
2. Java从入门到精通（第6版） 明日科技 编著 清华大学出版社
3. 零基础学Java项目开发 聚慕课教育研发中心 清华大学出版社

# **附录**

Strat

public class Start {  
 public static void main(String[] args) {  
 new LoginView();  
 }  
}

AddEmployeeController

public class AddEmployeeController implements ActionListener {  
 AddEmployeeView addedEmployeeView;  
 MainView mainView;  
  
 public AddEmployeeController(AddEmployeeView addedEmployeeView, MainView mainView) {  
 this.addedEmployeeView = addedEmployeeView;  
 this.mainView = mainView;  
 }  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JButton jButton= (JButton) e.getSource();  
 EmployeeService employeeService=new EmployeeServiceImpl();  
 EmployeeDO employeeDO=addedEmployeeView.buildVegetableDO();  
 boolean addResult=employeeService.add(employeeDO);  
 if(addResult){  
 //刷新表格内容  
 mainView.reloadTable();  
 addedEmployeeView.dispose();  
 }else{  
 JOptionPane.showMessageDialog(addedEmployeeView,"添加失败");  
 }  
 }  
}

EmployeeMainViewController

public class EmployeeMainViewController implements ActionListener {  
 private EmployeeMainView employeeMainView;  
 public EmployeeMainViewController(EmployeeMainView employeeMainView) {  
 this.employeeMainView = employeeMainView;  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JButton jButton= (JButton) e.getSource();  
 String text=jButton.getText();  
 if("查询".equals(text)){  
 employeeMainView.setPageNow(1);  
 employeeMainView.reloadTable();  
 }else if("上一页".equals(text)){  
 employeeMainView.setPageNow(employeeMainView.getPageNow()-1);  
 employeeMainView.reloadTable();  
 }else if("下一页".equals(text)){  
 employeeMainView.setPageNow(employeeMainView.getPageNow()+1);  
 employeeMainView.reloadTable();  
 }else if("重新登录".equals(text)){  
 employeeMainView.dispose();  
 new LoginView();  
 }else if("显示图表".equals(text)){  
 new DataChart();  
 }  
 }  
}

EmployeeRequest

public class EmployeeRequest {  
 private int pageNow;  
 private int pageSize;  
 private int start;  
 private String searchKey;//查询词  
  
 public int getPageNow() {  
 return pageNow;  
 }  
  
 public void setPageNow(int pageNow) {  
 this.pageNow = pageNow;  
 }  
  
 public int getPageSize() {  
 return pageSize;  
 }  
  
 public void setPageSize(int pageSize) {  
 this.pageSize = pageSize;  
 }  
  
 public int getStart() {  
 return (pageNow-1)\*pageSize;  
 }  
  
  
 public String getSearchKey() {  
 return searchKey;  
 }  
  
 public void setSearchKey(String searchKey) {  
 this.searchKey = searchKey;  
 }  
}

LoginController

public class LoginController extends KeyAdapter implements ActionListener {  
 //继承KeyAdapter适配器避免了过多无用方法出现，实现ActionListener接口添加监听事件  
 private LoginView loginView;  
 public LoginController(LoginView loginView) {  
 this.loginView=loginView;  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JButton jButton= (JButton) e.getSource();  
 String text=jButton.getText();  
 if("注册".equals(text)){  
 //register();  
 new RegisterView(loginView);  
  
 }else if("管理登录".equals(text)){  
 login();  
 // new EmployeeMainView();  
 //System.out.println("登录成功");  
 }else if("用户登录".equals(text)){  
 employeelogin();  
  
 }  
 }  
 private void login() {  
 String user= loginView.getUserField().getText();  
 char[]chars=loginView.getPasswordField().getPassword();  
 if(user==null||"".equals(user.trim())||chars==null||chars.length==0){  
 JOptionPane.showMessageDialog(loginView,"用户名密码必填");  
 return;  
 }  
 String passWord=new String(chars);  
 //查询数据库  
 AdminService adminService=new AdminServiceImpl();//创建校验对象  
 AdminDO adminDO=new AdminDO();//创建管理账户对象  
 //将键盘上接受的账号密码赋值给账户对象  
 adminDO.setUserName(user);  
 adminDO.setPwd(passWord);  
  
 //登录成功的标志  
 boolean flag= adminService.validata(adminDO);  
 if (flag){  
 loginView.dispose();  
 new MainView();  
 System.out.println("登录成功");  
 }else{  
 JOptionPane.showMessageDialog(loginView,"用户名或密码错误");  
 }  
 }  
 private void employeelogin() {  
 String user= loginView.getUserField().getText();  
 char[]chars=loginView.getPasswordField().getPassword();  
 if(user==null||"".equals(user.trim())||chars==null||chars.length==0){  
 JOptionPane.showMessageDialog(loginView,"用户名密码必填");  
 return;  
 }  
 String passWord=new String(chars);  
 //查询数据库  
 AdminService adminService=new AdminServiceImpl();//创建校验对象  
 AdminDO adminDO=new AdminDO();//创建管理账户对象  
 //将键盘上接受的账号密码赋值给账户对象  
 adminDO.setUserName(user);  
 adminDO.setPwd(passWord);  
  
 //登录成功的标志  
 boolean flag= adminService.employeevalidata(adminDO);  
 if (flag){  
 loginView.dispose();  
 new EmployeeMainView();  
 System.out.println("登录成功");  
 }else{  
 JOptionPane.showMessageDialog(loginView,"用户名或密码错误");  
 }  
 }

MainViewController

public class MainViewController implements ActionListener {  
 private MainView mainView;  
 public MainViewController(MainView mainView) {  
 this.mainView=mainView;  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JButton jButton= (JButton) e.getSource();  
 String text=jButton.getText();  
 if("添加".equals(text)){  
 new AddEmployeeView(mainView);  
 }else if("修改".equals(text)){  
 int[]selestedVegetableIds=mainView.getSelectedStudentIds();  
 if(selestedVegetableIds.length!=1){  
 JOptionPane.showMessageDialog(mainView,"一次只能修改一行！");  
 return;  
 }  
 new UpdateView(mainView,selestedVegetableIds[0]);  
  
 }else if("删除".equals(text)){  
 int[]selectedVegetableIds=mainView.getSelectedStudentIds();  
 if(selectedVegetableIds.length==0){  
 JOptionPane.showMessageDialog(mainView,"请选择要删除的行");  
 return;  
 }  
 int option=JOptionPane.showConfirmDialog(mainView,"你确定要删除选择的"+selectedVegetableIds.length  
 +"行吗？","确认删除",JOptionPane.YES\_NO\_OPTION);  
 if(option==JOptionPane.YES\_OPTION){  
 //执行删除操作  
 EmployeeService employeeService=new EmployeeServiceImpl();  
 boolean deleteResult=employeeService.delete(selectedVegetableIds);  
 if(deleteResult){  
 mainView.reloadTable();  
 }else{  
 JOptionPane.showMessageDialog(mainView,"删除失败");  
 }  
 }  
 }else if("查询".equals(text)){  
 mainView.setPageNow(1);  
 mainView.reloadTable();  
 }else if("上一页".equals(text)){  
 mainView.setPageNow(mainView.getPageNow()-1);  
 mainView.reloadTable();  
 }else if("下一页".equals(text)){  
 mainView.setPageNow(mainView.getPageNow()+1);  
 mainView.reloadTable();  
 }else if("重新登录".equals(text)){  
 mainView.dispose();  
 new LoginView();  
 }else if("显示图表".equals(text)){  
 new DataChart();  
 }  
 }  
}

RegisterController

public class RegisterController implements ActionListener {  
 //继承KeyAdapter适配器避免了过多无用方法出现，实现ActionListener接口添加监听事件  
 private LoginView loginView;  
 private RegisterView registerView;  
  
 public RegisterController(LoginView loginView, RegisterView registerView) {  
 this.loginView = loginView;  
 this.registerView = registerView;  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JButton jButton= (JButton) e.getSource();  
 String text=jButton.getText();  
 if("注册".equals(text)){  
 register();  
 }  
 }  
 private void register(){  
  
 AdminDO adminDO=registerView.buildAdminDO();  
 String user= adminDO.getUserName();  
 String password=adminDO.getPwd();  
 if(user==null||"".equals(user.trim())||password==null||"".equals(password.trim())){  
 JOptionPane.showMessageDialog(loginView,"用户名或密码为空");  
 return;  
 }  
 //查询数据库  
 AdminService adminService=new AdminServiceImpl();//创建校验对象  
 //账户存在  
 boolean flag= adminService.existdata(adminDO);  
 if (flag){  
 JOptionPane.showMessageDialog(loginView,"该账户已存在");  
 }else{  
 //注册，在数据库中插入信息  
 boolean registerFlag = adminService.registerdata(adminDO);  
 if(registerFlag){  
 JOptionPane.showMessageDialog(loginView,"注册成功");  
 registerView.dispose();  
 }  
 }  
 }  
}

UpdateViewController

public class UpdateViewController implements ActionListener {  
  
 private UpdateView updateView;  
 private MainView mainView;  
  
 public UpdateViewController(UpdateView updateView, MainView mainView) {  
 this.updateView = updateView;  
 this.mainView = mainView;  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JButton jButton= (JButton) e.getSource();  
 String text=jButton.getText();  
 if("修改".equals(text)){  
 EmployeeService employeeService=new EmployeeServiceImpl();  
 EmployeeDO employeeDO=updateView.buildUpdateEmployeeDO();  
 boolean updateResult=employeeService.update(employeeDO);  
 if(updateResult){  
 mainView.reloadTable();  
 updateView.dispose();  
 }else{  
 JOptionPane.showMessageDialog(updateView,"修改失败");  
 }  
 }  
 }  
}

AdminDO

public class AdminDO {  
  
 private String userName;  
 private String pwd;  
  
  
 public String getUserName() {  
 return userName;  
 }  
  
 public void setUserName(String userName) {  
 this.userName = userName;  
 }  
  
 public String getPwd() {  
 return pwd;  
 }  
  
 public void setPwd(String pwd) {  
 this.pwd = pwd;  
 }  
}

EmployeeDO

public class EmployeeDO {  
 private Integer id;  
 private Integer age;  
 private String name;  
 private String sex;  
 private String apartment;  
 private String position;  
 private Double bonus;  
 private Double baseSalary;  
 private Double sum;  
  
 public Integer getId() {  
 return id;  
 }  
  
 public void setId(Integer id) {  
 this.id = id;  
 }  
  
 public Integer getAge() {  
 return age;  
 }  
  
 public void setAge(Integer age) {  
 this.age = age;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public String getSex() {  
 return sex;  
 }  
  
 public void setSex(String sex) {  
 this.sex = sex;  
 }  
  
 public String getApartment() {  
 return apartment;  
 }  
  
 public void setApartment(String apartment) {  
 this.apartment = apartment;  
 }  
  
 public String getPosition() {  
 return position;  
 }  
  
 public void setPosition(String position) {  
 this.position = position;  
 }  
  
 public Double getBonus() {  
 return bonus;  
 }  
  
 public void setBonus(Double bonus) {  
 this.bonus = bonus;  
 }  
  
 public Double getBaseSalary() {  
 return baseSalary;  
 }  
  
 public void setBaseSalary(Double baseSalary) {  
 this.baseSalary = baseSalary;  
 }  
  
 public Double getSum() {  
 return sum;  
 }  
  
 public void setSum(Double sum) {  
 this.sum = sum;  
 }  
}

TableDTO

public class TableDTO {  
 //表格模型  
 private Vector<Vector<Object>>data;  
 private int totalCount;  
  
 public Vector<Vector<Object>> getData() {  
 return data;  
 }  
  
 public void setData(Vector<Vector<Object>> data) {  
 this.data = data;  
 }  
  
 public int getTotalCount() {  
 return totalCount;  
 }  
  
 public void setTotalCount(int totalCount) {  
 this.totalCount = totalCount;  
 }  
}

AdminService

public interface AdminService {  
 boolean validata(AdminDO adminDO);//校验用户登录信息  
 boolean employeevalidata(AdminDO adminDO);//校验用户登录信息  
 boolean existdata(AdminDO adminDO);//判断用户是否存在  
 boolean registerdata(AdminDO adminDO);//新增用户  
}

EmployeeService

public interface EmployeeService {  
 TableDTO retrieveEmployee(EmployeeRequest employeeRequest);//在数据库取回数据  
 boolean add(EmployeeDO employeeDO);  
 boolean delete(int[]selectEmployeeIds);  
 boolean update(EmployeeDO employeeDO);  
 EmployeeDO getById(int selectedEmployeeId);  
 double []getAllSalary();  
}

AddEmployeeView

public class AddEmployeeView extends JDialog {  
 JPanel jPanel=new JPanel(new FlowLayout(FlowLayout.CENTER,10,20));  
 JLabel nameLabel=new JLabel("姓 名：",JLabel.RIGHT);  
 JTextField nameTxt=new JTextField();  
 JLabel ageLabel =new JLabel("年 龄：",JLabel.RIGHT);  
 JTextField ageTxt =new JTextField();  
  
 JLabel sexLabel=new JLabel("性 别：",JLabel.RIGHT);  
// JTextField sexTxt=new JTextField();  
 JComboBox<String>sexTxt=new JComboBox<>();  
 JLabel baseLabel =new JLabel("基础工资：",JLabel.RIGHT);  
 JTextField baseTxt =new JTextField();  
  
 JLabel bonusLabel =new JLabel("奖 金：",JLabel.RIGHT);  
 JTextField bonusTxt =new JTextField();  
 JLabel apartmentLabel =new JLabel("部 门：",JLabel.RIGHT);  
// JTextField apartmentTxt =new JTextField();//部门改用列表框  
 JComboBox<String>apartmentTxt=new JComboBox<>();  
  
 JLabel positionLabel =new JLabel("职 位：",JLabel.RIGHT);  
 // JTextField positionTxt =new JTextField();//职位改用列表框  
 JComboBox<String>positionTxt=new JComboBox<>();  
 JButton addBtn=new JButton("确认添加");  
 AddEmployeeController addVegetableController;  
  
 public AddEmployeeView(MainView mainView) {  
 super(mainView,"添加",true);  
 addVegetableController=new AddEmployeeController(this,mainView);  
 nameLabel.setPreferredSize(new Dimension(80,30));  
 nameTxt.setPreferredSize(new Dimension(200,30));  
 nameLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 nameTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 jPanel.add(nameLabel);  
 jPanel.add(nameTxt);  
 ageLabel.setPreferredSize(new Dimension(80,30));  
 ageTxt.setPreferredSize(new Dimension(200,30));  
 ageLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 ageTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 jPanel.add(ageLabel);  
 jPanel.add(ageTxt);  
 sexLabel.setPreferredSize(new Dimension(80,30));  
 sexTxt.setPreferredSize(new Dimension(200,30));  
 sexLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 sexTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 sexTxt.addItem("男");  
 sexTxt.addItem("女");  
 jPanel.add(sexLabel);  
 jPanel.add(sexTxt);  
 baseLabel.setPreferredSize(new Dimension(80,30));  
 baseTxt.setPreferredSize(new Dimension(200,30));  
 baseLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 baseTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 jPanel.add(baseLabel);  
 jPanel.add(baseTxt);  
 bonusLabel.setPreferredSize(new Dimension(80,30));  
 bonusTxt.setPreferredSize(new Dimension(200,30));  
 bonusLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 bonusTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 jPanel.add(bonusLabel);  
 jPanel.add(bonusTxt);  
 apartmentLabel.setPreferredSize(new Dimension(80,30));  
 apartmentTxt.setPreferredSize(new Dimension(200,30));  
 apartmentLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 apartmentTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 apartmentTxt.addItem("研发部");  
 apartmentTxt.addItem("运营部");  
 apartmentTxt.addItem("产品部");  
 apartmentTxt.addItem("市场部");  
 apartmentTxt.addItem("销售部");  
 jPanel.add(apartmentLabel);  
 jPanel.add(apartmentTxt);  
 positionLabel.setPreferredSize(new Dimension(80,30));  
 positionTxt.setPreferredSize(new Dimension(200,30));  
 positionLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 positionTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 positionTxt.addItem("助理");  
 positionTxt.addItem("经理");  
 positionTxt.addItem("编辑");  
 positionTxt.addItem("职员");  
 positionTxt.addItem("客服");  
 jPanel.add(positionLabel);  
 jPanel.add(positionTxt);  
  
 addBtn.addActionListener(addVegetableController);  
 addBtn.setFont(new Font("楷体",Font.PLAIN,15));  
 jPanel.add(addBtn);  
 Container container=getContentPane();  
 container.add(jPanel);  
 setIconImage(new ImageIcon(".\\src\\com\\msskx\\java\\res\\icon.png").getImage());  
  
 setSize(360,480);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  
 setResizable(false);  
 setVisible(true);  
 }  
 public EmployeeDO buildVegetableDO(){  
 EmployeeDO employeeDO=new EmployeeDO();  
 if(nameTxt.getText().equals("")){  
 JOptionPane.showMessageDialog(this,"姓名不可以为空");  
 return null;  
 }  
 if(Integer.valueOf(ageTxt.getText())<0){  
 JOptionPane.showMessageDialog(this,"请输入正常年龄");  
 return null;  
 }  
 employeeDO.setName(nameTxt.getText());  
 employeeDO.setAge(Integer.valueOf(ageTxt.getText()));  
 employeeDO.setSex((String) sexTxt.getSelectedItem());  
  
 employeeDO.setBaseSalary(Double.valueOf(baseTxt.getText()));  
 employeeDO.setBonus(Double.valueOf(bonusTxt.getText()));  
 employeeDO.setSum(Double.valueOf(baseTxt.getText())+Double.valueOf(bonusTxt.getText()));  
 employeeDO.setApartment((String) apartmentTxt.getSelectedItem());  
 employeeDO.setPosition((String) positionTxt.getSelectedItem());  
  
 return employeeDO;  
 }  
}

DataChart

public class DataChart {  
 public DataChart() {  
 DefaultPieDataset dataSet = new DefaultPieDataset();  
 EmployeeServiceImpl employeeServiceImpl = new EmployeeServiceImpl();  
 double[] sum=employeeServiceImpl.getAllSalary();  
 double development=sum[0];  
 double operation=sum[1];  
 double product=sum[2];  
 double market=sum[3];  
 double sales=sum[4];  
  
  
 dataSet.setValue("研发部", development);  
 dataSet.setValue("运营部", operation);  
 dataSet.setValue("产品部", product);  
 dataSet.setValue("市场部", market);  
 dataSet.setValue("销售部", sales);  
  
 JFreeChart chart = ChartFactory.createPieChart3D("部门工资比例表", (PieDataset)dataSet, true, true, true);  
  
 LegendTitle legend = null;  
 TextTitle txtTitle = null;  
 PiePlot categoryplot = null;  
  
 legend = chart.getLegend();  
 txtTitle = chart.getTitle();  
 categoryplot = (PiePlot) chart.getPlot();  
  
 txtTitle.setFont(new Font("幼圆", Font.BOLD, 36)); // 设置标题字体  
 categoryplot.setLabelFont(new Font("微软雅黑", 30, 18));// 设置图片上固定指示文字字体  
 legend.setItemFont(new Font("微软雅黑", 30, 16));// 设置图例字体  
  
 ChartFrame chartFrame = new ChartFrame("部门工资比例表", chart);  
 chartFrame.pack();  
 chartFrame.setIconImage(new ImageIcon(".\\src\\com\\msskx\\java\\res\\icon.png").getImage());  
 chartFrame.setResizable(false);  
 chartFrame.setLocationRelativeTo(null);  
 chartFrame.setVisible(true);  
  
 }  
}

EmployeeMainView

public class EmployeeMainView extends JFrame {  
 Container container=getContentPane();  
 JPanel northPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));//面板流式布局向左对齐  
  
 JTextField searchTXt=new JTextField(15);  
 JButton searchBtn=new JButton("查询");  
 JButton chartBtn =new JButton("显示图表");  
 JButton reLogin=new JButton("重新登录");  
 JPanel southPanel=new JPanel(new FlowLayout(FlowLayout.RIGHT));//面板流式布局向右对齐  
 JButton preBtn=new JButton("上一页");  
 JButton nextBtn=new JButton("下一页");  
  
  
 private int pageNow=1;//当前页数  
 private int pageSize=10;//一页显示记录条数  
  
 public int getPageNow() {  
 return pageNow;  
 }  
  
 public void setPageNow(int pageNow) {  
 this.pageNow = pageNow;  
 }  
  
 MainViewTable mainViewTable=new MainViewTable();  
 //主页面监听控制  
 EmployeeMainViewController employeeMainViewController=new EmployeeMainViewController(this);  
  
 public EmployeeMainView() {  
 super("员工工资管理系统");  
 //全屏情况下窗体大小  
 Rectangle bounds= DimensionUtil.getBounds();  
 pageSize=Math.floorDiv(bounds.height,35);  
 //北面布局  
 layoutNorth();  
 //中心布局  
 layoutCenter();  
 //南面布局  
 layoutSouth();  
  
 setIconImage(new ImageIcon(".\\src\\com\\msskx\\java\\res\\icon.png").getImage());  
 setBounds(bounds);  
 setExtendedState(MAXIMIZED\_BOTH);//默认最大化  
 setLocationRelativeTo(null);//默认居中  
 setDefaultCloseOperation(EXIT\_ON\_CLOSE);//退出销毁窗体  
 setResizable(true);  
 setVisible(true);  
 }  
  
 private void layoutCenter() {  
 TableDTO dto=getableDTO();  
 //将获取到的表格数据整合到表格模型  
 MainViewTableModel mainViewTableModel=MainViewTableModel.assembleModel(dto.getData());  
 mainViewTable.setModel(mainViewTableModel);//给表格设置模型  
 mainViewTable.renderRule();//渲染  
 JScrollPane jScrollPane=new JScrollPane(mainViewTable);//将表格添加到滚动面板  
 container.add(jScrollPane,BorderLayout.CENTER);  
 showPreNext(dto.getTotalCount());  
 }  
 /\*\*  
 \* 将两个按钮添加到南面  
 \*/  
 private void layoutSouth() {  
 preBtn.addActionListener(employeeMainViewController);  
 nextBtn.addActionListener(employeeMainViewController);  
 southPanel.add(preBtn);  
 southPanel.add(nextBtn);  
 container.add(southPanel,BorderLayout.SOUTH);  
 }  
 /\*\*  
 \* 将四个按钮和一个搜索框加入北面  
 \*/  
 private void layoutNorth() {  
  
 searchBtn.addActionListener(employeeMainViewController);  
 reLogin.addActionListener(employeeMainViewController);  
 chartBtn.addActionListener(employeeMainViewController);  
 searchBtn.setFont(new Font("宋体",Font.BOLD,18));  
 searchTXt.setFont(new Font("宋体",Font.BOLD,18));  
 reLogin.setFont(new Font("宋体",Font.BOLD,18));  
 chartBtn.setFont(new Font("宋体",Font.BOLD,18));  
 northPanel.add(searchTXt);  
 northPanel.add(searchBtn);  
 northPanel.add(chartBtn);  
 northPanel.add(reLogin);  
  
 container.add(northPanel,BorderLayout.NORTH);  
 }  
 /\*\*  
 \* 获取表格对象  
 \* @return  
 \*/  
 private TableDTO getableDTO(){  
 EmployeeService employeeService=new EmployeeServiceImpl();  
 EmployeeRequest request=new EmployeeRequest();  
 request.setPageNow(pageNow);  
 request.setPageSize(pageSize);  
 request.setSearchKey(searchTXt.getText().trim());  
 //将当前页面的基本信息获取到，利用获取到的数据初始化表格模型  
 TableDTO tableDTO=employeeService.retrieveEmployee(request);  
 return tableDTO;  
 }  
 //设置南边布局按钮是否可见  
 private void showPreNext(int totalCount){  
 if(pageNow==1){  
 preBtn.setVisible(false);  
 }else{  
 preBtn.setVisible(true);  
 }  
 int pageCount=0;  
 if(totalCount%pageSize==0){  
 //总页数/每页个数等于页数  
 pageCount=totalCount/pageSize;  
 }else{  
 //如果不能整除说明数据多余，还要另外再取一页  
 pageCount=totalCount/pageSize+1;  
 }  
 if(pageNow==pageCount){  
 nextBtn.setVisible(false);  
 }else{  
 nextBtn.setVisible(true);  
 }  
 }  
  
 /\*\*  
 \* 刷新表格  
 \*/  
 public void reloadTable(){  
 TableDTO dto=getableDTO();//获取表格对象  
 MainViewTableModel.updateModel(dto.getData());//更新表格模型  
 mainViewTable.renderRule();//渲染样式  
 showPreNext(dto.getTotalCount());//是否显示按钮  
 }  
}

LoginView

public class LoginView extends JFrame {  
 JLabel titleLabel=new JLabel("员工工资管理系统",JLabel.CENTER);  
 JLabel userLabel=new JLabel("用户名:",JLabel.CENTER);  
 JLabel passwordLabel=new JLabel("密 码:",JLabel.CENTER);  
 JTextField userField=new JTextField();  
 JPasswordField passwordField=new JPasswordField();  
 JButton loginButton=new JButton("管理登录");  
 JButton employeeLoginButton=new JButton("用户登录");  
 JButton registerButton=new JButton("注册");  
 //定义弹簧布局  
 SpringLayout springLayout=new SpringLayout();  
 //定义根面板并设置为弹簧布局  
 JPanel centerPanel =new JPanel(springLayout);  
 //监听  
 LoginController loginController;  
  
 //初始化  
 public LoginView(){  
 super("员工工资管理系统");  
 Container container=getContentPane();  
//设置每个控件的字体和大小  
 titleLabel.setFont(new Font("幼圆",Font.BOLD,40));  
 titleLabel.setSize(new Dimension(0,80));  
  
 userLabel.setFont(new Font("楷体",Font.PLAIN,20));  
 userField.setPreferredSize(new Dimension(200,30));  
 passwordLabel.setFont(new Font("楷体",Font.PLAIN,20));  
 passwordField.setPreferredSize(new Dimension(200,30));  
 loginButton.setFont(new Font("楷体",Font.PLAIN,20));  
 employeeLoginButton.setFont(new Font("楷体",Font.PLAIN,20));  
 registerButton.setFont(new Font("楷体",Font.PLAIN,20));  
  
 //添加登录监听  
 loginController =new LoginController(this);  
 centerPanel.add(userLabel);  
 centerPanel.add(userField);  
 centerPanel.add(passwordLabel);  
 centerPanel.add(passwordField);  
 loginButton.addActionListener(loginController);//鼠标点击登录  
 loginButton.addKeyListener(loginController);//按键事件，回车确认登录  
 centerPanel.add(loginButton);  
 registerButton.addActionListener(loginController);//鼠标点击注册  
 centerPanel.add(employeeLoginButton);//用户登录  
 employeeLoginButton.addActionListener(loginController);  
 centerPanel.add(registerButton);  
 //对控件进行弹簧布局  
 //取出两个控件在一行的总宽度和之间预留的间距  
 Spring childWidth=Spring.sum(Spring.sum(Spring.width(userLabel),Spring.width(userField)),Spring.constant(30));  
 int offsetX=childWidth.getValue()/2;//正中心  
 //前面从属物后面是锚点  
 //userLabel的西边相对于面板的中心向左  
 //userLabel的北边相对于面板的北边距离pad  
 springLayout.putConstraint(SpringLayout.WEST,userLabel,-offsetX,SpringLayout.HORIZONTAL\_CENTER,centerPanel);  
 springLayout.putConstraint(SpringLayout.NORTH,userLabel,100,SpringLayout.NORTH,centerPanel);  
 //以userlabel为锚点对userField进行布局  
 //userField北边和userLabel一样,西边靠中心offset  
 springLayout.putConstraint(SpringLayout.EAST,userField,offsetX,SpringLayout.HORIZONTAL\_CENTER,centerPanel);  
 springLayout.putConstraint(SpringLayout.NORTH,userField,0,SpringLayout.NORTH,userLabel);  
 //passwordLabel的西边和userLabel相同，北边还有距离  
 springLayout.putConstraint(SpringLayout.WEST,passwordLabel,0,SpringLayout.WEST,userLabel);  
 springLayout.putConstraint(SpringLayout.NORTH,passwordLabel,30,SpringLayout.SOUTH,userField);  
 //passwordField的东边和userLabel相同，北边还有距离  
 springLayout.putConstraint(SpringLayout.EAST,passwordField,0,SpringLayout.EAST,userField);  
 springLayout.putConstraint(SpringLayout.NORTH,passwordField,30,SpringLayout.SOUTH,userField);  
  
 //LoginButton的西边和passwordLabel有距离，北边还有距离  
 springLayout.putConstraint(SpringLayout.WEST,loginButton,-20,SpringLayout.WEST,passwordLabel);  
 springLayout.putConstraint(SpringLayout.NORTH,loginButton,40,SpringLayout.SOUTH,passwordLabel);  
 //用户登录的西边和passwordLabel有距离，北边还有距离  
 springLayout.putConstraint(SpringLayout.WEST,employeeLoginButton,132,SpringLayout.WEST,loginButton);  
 springLayout.putConstraint(SpringLayout.NORTH,employeeLoginButton,40,SpringLayout.SOUTH,passwordLabel);  
 //registerButton的东边和passwordField有距离，北边还有距离  
 springLayout.putConstraint(SpringLayout.EAST,registerButton,20,SpringLayout.EAST,passwordField);  
 springLayout.putConstraint(SpringLayout.NORTH,registerButton,40,SpringLayout.SOUTH,passwordLabel);  
 //将标题向下移动  
 titleLabel.setBounds(new Rectangle());  
//设置标题标签与JFrame的距离  
 titleLabel.setBorder(new EmptyBorder(20, 10, 10, 10));  
 container.add(titleLabel,BorderLayout.NORTH);  
 container.add(centerPanel, BorderLayout.CENTER);  
  
 getRootPane().setDefaultButton(loginButton);  
  
 setIconImage(new ImageIcon(".\\src\\com\\msskx\\java\\res\\icon.png").getImage());  
  
 setSize(800,500);  
 setLocationRelativeTo(null);//定位与其他无关默认居中避免计算  
 setDefaultCloseOperation(EXIT\_ON\_CLOSE);  
 setResizable(false);  
 setVisible(true);  
 }  
 //返回输入框的内容  
 public JTextField getUserField() {  
 return userField;  
 }  
 public JPasswordField getPasswordField() {  
 return passwordField;  
 }  
}

public class MainView extends JFrame {  
 Container container=getContentPane();  
 JPanel northPanel=new JPanel(new FlowLayout(FlowLayout.LEFT));//面板流式布局向左对齐  
 JButton addBtn=new JButton("添加");  
 JButton updateBtn=new JButton("修改");  
 JButton delBtn=new JButton("删除");  
 JTextField searchTXt=new JTextField(15);  
 JButton searchBtn=new JButton("查询");  
 JButton chartBtn =new JButton("显示图表");  
 JButton reLogin=new JButton("重新登录");  
 JPanel southPanel=new JPanel(new FlowLayout(FlowLayout.RIGHT));//面板流式布局向右对齐  
 JButton preBtn=new JButton("上一页");  
 JButton nextBtn=new JButton("下一页");  
  
 private int pageNow=1;//当前页数  
 private int pageSize=10;//一页显示记录条数  
  
 public int getPageNow() {  
 return pageNow;  
 }  
  
 public void setPageNow(int pageNow) {  
 this.pageNow = pageNow;  
 }  
  
 MainViewTable mainViewTable=new MainViewTable();  
 //主页面监听控制  
 MainViewController mainViewController=new MainViewController(this);  
  
 public MainView() {  
 super("员工工资管理系统");  
 //全屏情况下窗体大小  
 Rectangle bounds= DimensionUtil.getBounds();  
 pageSize=Math.floorDiv(bounds.height,35);  
 //北面布局  
 layoutNorth();  
 //中心布局  
 layoutCenter();  
 //南面布局  
 layoutSouth();  
  
 setIconImage(new ImageIcon(".\\src\\com\\msskx\\java\\res\\icon.png").getImage());  
 setBounds(bounds);  
 setExtendedState(MAXIMIZED\_BOTH);//默认最大化  
 setLocationRelativeTo(null);//默认居中  
 setDefaultCloseOperation(EXIT\_ON\_CLOSE);//退出销毁窗体  
 setResizable(true);  
 setVisible(true);  
 }  
  
 private void layoutCenter() {  
 TableDTO dto=getableDTO();  
 //将获取到的表格数据整合到表格模型  
 MainViewTableModel mainViewTableModel=MainViewTableModel.assembleModel(dto.getData());  
 mainViewTable.setModel(mainViewTableModel);//给表格设置模型  
 mainViewTable.renderRule();//渲染  
 JScrollPane jScrollPane=new JScrollPane(mainViewTable);//将表格添加到滚动面板  
 container.add(jScrollPane,BorderLayout.CENTER);  
 showPreNext(dto.getTotalCount());  
 }  
 /\*\*  
 \* 将两个按钮添加到南面  
 \*/  
 private void layoutSouth() {  
 preBtn.addActionListener(mainViewController);  
 nextBtn.addActionListener(mainViewController);  
 southPanel.add(preBtn);  
 southPanel.add(nextBtn);  
 container.add(southPanel,BorderLayout.SOUTH);  
 }  
   
 private void layoutNorth() {  
 addBtn.addActionListener(mainViewController);  
 updateBtn.addActionListener(mainViewController);  
 delBtn.addActionListener(mainViewController);  
 searchBtn.addActionListener(mainViewController);  
 reLogin.addActionListener(mainViewController);  
 chartBtn.addActionListener(mainViewController);  
  
 addBtn.setFont(new Font("宋体",Font.BOLD,18));  
 updateBtn.setFont(new Font("宋体",Font.BOLD,18));  
 delBtn.setFont(new Font("宋体",Font.BOLD,18));  
 searchBtn.setFont(new Font("宋体",Font.BOLD,18));  
 searchTXt.setFont(new Font("宋体",Font.BOLD,18));  
 reLogin.setFont(new Font("宋体",Font.BOLD,18));  
 chartBtn.setFont(new Font("宋体",Font.BOLD,18));  
 northPanel.add(addBtn);  
 northPanel.add(updateBtn);  
 northPanel.add(delBtn);  
 northPanel.add(searchTXt);  
 northPanel.add(searchBtn);  
 northPanel.add(chartBtn);  
 northPanel.add(reLogin);  
  
 container.add(northPanel,BorderLayout.NORTH);  
 }  
 /\*\*  
 \* 获取表格对象  
 \* @return  
 \*/  
 private TableDTO getableDTO(){  
 EmployeeService employeeService=new EmployeeServiceImpl();  
 EmployeeRequest request=new EmployeeRequest();  
 request.setPageNow(pageNow);  
 request.setPageSize(pageSize);  
 request.setSearchKey(searchTXt.getText().trim());  
 //将当前页面的基本信息获取到，利用获取到的数据初始化表格模型  
 TableDTO tableDTO=employeeService.retrieveEmployee(request);  
 return tableDTO;  
 }  
 //设置南边布局按钮是否可见  
 private void showPreNext(int totalCount){  
 if(pageNow==1){  
 preBtn.setVisible(false);  
 }else{  
 preBtn.setVisible(true);  
 }  
 int pageCount=0;  
 if(totalCount%pageSize==0){  
 //总页数/每页个数等于页数  
 pageCount=totalCount/pageSize;  
 }else{  
 //如果不能整除说明数据多余，还要另外再取一页  
 pageCount=totalCount/pageSize+1;  
 }  
 if(pageNow==pageCount){  
 nextBtn.setVisible(false);  
 }else{  
 nextBtn.setVisible(true);  
 }  
 }  
  
 /\*\*  
 \* 刷新表格  
 \*/  
 public void reloadTable(){  
 TableDTO dto=getableDTO();//获取表格对象  
 MainViewTableModel.updateModel(dto.getData());//更新表格模型  
 mainViewTable.renderRule();//渲染样式  
 showPreNext(dto.getTotalCount());//是否显示按钮  
 }  
  
 /\*\*  
 \* 获取被选中的ids列  
 \* @return  
 \*/  
 public int[] getSelectedStudentIds(){  
 int[]selectedRows=mainViewTable.getSelectedRows();  
 int[]ids=new int[selectedRows.length];  
 for(int i=0;i<selectedRows.length;i++){  
 int rowIndex=selectedRows[i];  
 Object idObj=mainViewTable.getValueAt(rowIndex,0);  
 ids[i]=Integer.valueOf(idObj.toString());  
 }  
 return ids;  
 }  
   
 public void order() {  
 StringBuilder sql =new StringBuilder(" SELECT \* from employee order by id desc");  
 Connection conn=null;  
 PreparedStatement ps=null;  
  
 try {  
 conn= DButil.getConnection();  
 ps=conn.prepareStatement(sql.toString());  
  
 } catch (SQLException e) {  
 e.printStackTrace();  
 } finally {  
 DButil.closePS(ps);  
 DButil.closeConnection(conn);  
 }  
 }  
  
}

RegisterView

public class RegisterView extends JDialog {  
 JPanel jPanel=new JPanel(new FlowLayout(FlowLayout.CENTER,10,20));  
 JLabel nameLabel=new JLabel("用户名：",JLabel.RIGHT);  
 JTextField nameTxt=new JTextField();  
 JLabel passwordLabel=new JLabel("密 码：",JLabel.RIGHT);  
 JPasswordField passwordTxt=new JPasswordField();  
  
 JButton register =new JButton("注册");  
 RegisterController registerController;  
  
 public RegisterView(LoginView loginView) {  
 super(loginView,"用户注册",true);  
 nameLabel.setPreferredSize(new Dimension(80,30));  
 nameTxt.setPreferredSize(new Dimension(200,30));  
 nameLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 jPanel.add(nameLabel);  
 jPanel.add(nameTxt);  
 passwordLabel.setPreferredSize(new Dimension(80,30));  
 passwordLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 passwordTxt.setPreferredSize(new Dimension(200,30));  
 jPanel.add(passwordLabel);  
 jPanel.add(passwordTxt);  
// confirmLabel.setPreferredSize(new Dimension(80,30));  
// confirmTxt.setPreferredSize(new Dimension(200,30));  
// jPanel.add(confirmLabel);  
// jPanel.add(confirmTxt);  
 registerController=new RegisterController(loginView,this);  
 register.addActionListener(registerController);  
 register.setFont(new Font("楷体",Font.PLAIN,15));  
 jPanel.add(register);  
 Container container=getContentPane();  
 container.add(jPanel);  
 setIconImage(new ImageIcon(".\\src\\com\\msskx\\res\\icon.png").getImage());  
  
 setSize(360,200);  
 setLocationRelativeTo(null);  
 setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  
 setResizable(false);  
 setVisible(true);  
 }  
 //注册得到一个用户对象  
 public AdminDO buildAdminDO(){  
 AdminDO adminDO=new AdminDO();  
 adminDO.setUserName(nameTxt.getText());  
 adminDO.setPwd(String.valueOf(passwordTxt.getPassword()));  
  
 return adminDO;  
 }  
}

UpdateView

public class UpdateView extends JDialog {  
 JPanel jPanel=new JPanel(new FlowLayout(FlowLayout.CENTER,10,20));  
 JLabel idLabel=new JLabel("工 号：",JLabel.RIGHT);  
 JTextField idTxt=new JTextField();  
 JLabel nameLabel=new JLabel("姓 名：",JLabel.RIGHT);  
 JTextField nameTxt=new JTextField();  
 JLabel ageLabel =new JLabel("年 龄：",JLabel.RIGHT);  
 JTextField ageTxt =new JTextField();  
  
 JLabel sexLabel=new JLabel("性 别：",JLabel.RIGHT);  
 //JTextField sexTxt=new JTextField();  
 JComboBox<String>sexTxt=new JComboBox<>();  
 JLabel baseLabel =new JLabel("基础工资：",JLabel.RIGHT);  
 JTextField baseTxt =new JTextField();  
  
 JLabel bonusLabel =new JLabel("奖 金：",JLabel.RIGHT);  
 JTextField bonusTxt =new JTextField();  
 JLabel apartmentLabel =new JLabel("部 门：",JLabel.RIGHT);  
 // JTextField apartmentTxt =new JTextField();  
 JComboBox<String>apartmentTxt=new JComboBox<>();  
 JLabel positionLabel =new JLabel("职 位：",JLabel.RIGHT);  
 //JTextField positionTxt =new JTextField();  
 JComboBox<String>positionTxt=new JComboBox<>();  
 JButton updateBtn =new JButton("修改");  
 UpdateViewController updateViewController;  
  
 public UpdateView(MainView mainView,int selectedvegetableId) {  
 super(mainView,"修改员工",true);  
 updateViewController=new UpdateViewController(this,mainView);  
 //查询selectedStudentId对应的记录并回显  
 EmployeeService employeeService=new EmployeeServiceImpl();  
 EmployeeDO employeeDO=employeeService.getById(selectedvegetableId);  
  
 idLabel.setPreferredSize(new Dimension(80,30));  
 jPanel.add(idLabel);  
 idTxt.setPreferredSize(new Dimension(200,30));  
 idLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 idTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 idTxt.setText(employeeDO.getId()+"");  
 //设置id不可编辑  
 idTxt.setEnabled(false);  
 jPanel.add(idTxt);  
 nameLabel.setPreferredSize(new Dimension(80,30));  
 nameTxt.setPreferredSize(new Dimension(200,30));  
 nameLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 nameTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 nameTxt.setText(employeeDO.getName());  
 jPanel.add(nameLabel);  
 jPanel.add(nameTxt);  
 ageLabel.setPreferredSize(new Dimension(80,30));  
 ageTxt.setPreferredSize(new Dimension(200,30));  
 ageLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 ageTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 ageTxt.setText(String.valueOf(employeeDO.getAge()));  
 jPanel.add(ageLabel);  
 jPanel.add(ageTxt);  
 sexLabel.setPreferredSize(new Dimension(80,30));  
 sexTxt.setPreferredSize(new Dimension(200,30));  
 sexLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 sexTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 //sexTxt.setText(employeeDO.getSex());//改为下拉框  
 sexTxt.addItem("男");  
 sexTxt.addItem("女");  
 sexTxt.setSelectedItem(employeeDO.getSex());  
 jPanel.add(sexLabel);  
 jPanel.add(sexTxt);  
 baseLabel.setPreferredSize(new Dimension(80,30));  
 baseTxt.setPreferredSize(new Dimension(200,30));  
 baseLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 baseTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 baseTxt.setText(String.valueOf(employeeDO.getBaseSalary()));  
 jPanel.add(baseLabel);  
 jPanel.add(baseTxt);  
 bonusLabel.setPreferredSize(new Dimension(80,30));  
 bonusTxt.setPreferredSize(new Dimension(200,30));  
 bonusLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 bonusTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 bonusTxt.setText(String.valueOf(employeeDO.getBonus()));  
  
 jPanel.add(bonusLabel);  
 jPanel.add(bonusTxt);  
 apartmentLabel.setPreferredSize(new Dimension(80,30));  
 apartmentTxt.setPreferredSize(new Dimension(200,30));  
 apartmentLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 apartmentTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 apartmentTxt.addItem("研发部");  
 apartmentTxt.addItem("运营部");  
 apartmentTxt.addItem("产品部");  
 apartmentTxt.addItem("市场部");  
 apartmentTxt.addItem("销售部");  
 //apartmentTxt.setText(employeeDO.getApartment());  
 apartmentTxt.setSelectedItem(employeeDO.getApartment());  
 jPanel.add(apartmentLabel);  
 jPanel.add(apartmentTxt);  
  
 positionLabel.setPreferredSize(new Dimension(80,30));  
 positionTxt.setPreferredSize(new Dimension(200,30));  
 positionLabel.setFont(new Font("楷体",Font.PLAIN,15));  
 positionTxt.setFont(new Font("楷体",Font.PLAIN,15));  
 //positionTxt.setText(employeeDO.getPosition());  
 positionTxt.addItem("助理");  
 positionTxt.addItem("经理");  
 positionTxt.addItem("编辑");  
 positionTxt.addItem("职员");  
 positionTxt.addItem("客服");  
 positionTxt.setSelectedItem(employeeDO.getPosition());  
 jPanel.add(positionLabel);  
 jPanel.add(positionTxt);  
 updateBtn.addActionListener(updateViewController);  
 updateBtn.setFont(new Font("楷体",Font.PLAIN,15));  
 jPanel.add(updateBtn);  
 Container container=getContentPane();  
 container.add(jPanel);  
  
 setIconImage(new ImageIcon(".\\src\\com\\msskx\\java\\res\\icon.png").getImage());  
  
 setSize(360,500);  
 setLocationRelativeTo(null);  
 //只销毁当前窗体  
 setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);  
 setResizable(false);  
 setVisible(true);  
 }  
  
 public EmployeeDO buildUpdateEmployeeDO() {  
 EmployeeDO employeeDO=new EmployeeDO();  
 if(nameTxt.getText().equals("")){  
 JOptionPane.showMessageDialog(this,"姓名不可以为空");  
 return null;  
 }  
 if(Integer.valueOf(ageTxt.getText())<0){  
 JOptionPane.showMessageDialog(this,"请输入正常年龄");  
 return null;  
 }  
 employeeDO.setId(Integer.valueOf(idTxt.getText()));  
 employeeDO.setName(nameTxt.getText());  
 employeeDO.setAge(Integer.valueOf(ageTxt.getText()));  
 //employeeDO.setSex(sexTxt.getText());//性别修改为下拉选择框  
 employeeDO.setSex((String) sexTxt.getSelectedItem());  
 employeeDO.setBaseSalary(Double.valueOf(baseTxt.getText()));  
 employeeDO.setBonus(Double.valueOf(bonusTxt.getText()));  
 employeeDO.setSum(Double.valueOf(baseTxt.getText())+Double.valueOf(bonusTxt.getText()));  
 employeeDO.setApartment((String) apartmentTxt.getSelectedItem());  
 employeeDO.setPosition((String) positionTxt.getSelectedItem());  
 return employeeDO;  
 }  
}