|  |  |
| --- | --- |
| 成绩 |  |



**课程设计报告**



**题 目 网上拍卖管理系统**

**课 程 名 称 企业级工程项目实战**

**院 部 名 称**  软件工程学院

**专 业** 软件工程（本）

**班 级** 15软件工程（Z）

**学 生 姓 名**  吴跟强

**学 号** 1512001066

**课程设计地点** 北辅311

**指 导 教 师** 王勇

金陵科技学院教务处制

**课程设计报告书写要求**

课程设计报告原则上要求学生手写，要求书写工整。若因课程特点需打印的，要遵照以下字体、字号、间距等的具体要求。纸张一律采用A4的纸张。

**课程设计报告书写说明**

课程设计报告应包含以下七部分内容：1、摘要  2、目录  3、前言/引言  4.正文 5. 结论   6. 参考文献    7. 附录，每部分的书写要求参见具体条目要求。

**填写注意事项**

（1）准确说明，层次清晰。

（2）尽量采用专用术语来说明事物。

（3）外文、符号、公式要准确，应使用统一规定的名词和符号。

（4）应独立完成课程设计报告的书写，严禁抄袭、复印，一经发现，以零分论处。

**课程设计报告批改说明**

课程设计报告的批改要及时、认真、仔细，一律用红色笔批改。课程设计报告的批改成绩采用五级计分制或百分制，具体评分标准由各院部自行制定。

**课程设计报告装订要求**

报告批改完毕后，任课老师应将课程设计报告以自然班为单位、按学号升序排列，并附上一份该课程设计的教学大纲。

网上拍卖管理系统

工具类导入导出系统设计文档

目录

[1引言 6](#_Toc514833717)

[1.1背景 6](#_Toc514833718)

[1.2前景 6](#_Toc514833719)

[1.3概述 6](#_Toc514833720)

[1.4适用范围 6](#_Toc514833721)

[1.5读者对象 6](#_Toc514833722)

[2设计约束 7](#_Toc514833723)

[2.1需求约束 7](#_Toc514833724)

[2.2设计策略 7](#_Toc514833725)

[2.3技术实现 7](#_Toc514833726)

[3系统分析 8](#_Toc514833727)

[3.1总体描述 8](#_Toc514833728)

[3.1.1产品功能 8](#_Toc514833729)

[3.1.2目标 8](#_Toc514833730)

[3.1.3用户特点 9](#_Toc514833731)

[3.1.4运行环境 9](#_Toc514833732)

[4系统设计 10](#_Toc514833733)

[4.1系统总体设计 10](#_Toc514833734)

[4.1.1总体设计目标 10](#_Toc514833735)

[4.1.2总体架构 10](#_Toc514833736)

[4.2业务逻辑分析与设计 10](#_Toc514833737)

[4.2.1功能模块图 10](#_Toc514833738)

[4.2.2数据流图 11](#_Toc514833739)

[4.2.3E-R实体关系图 12](#_Toc514833740)

[5数据库设计 12](#_Toc514833741)

[5.1账户信息导入 12](#_Toc514833742)

[5.2部门信息导入 13](#_Toc514833743)

[5.3职位信息导入 14](#_Toc514833744)

[5.4文件上传 14](#_Toc514833745)

[5.5订单信息显示 15](#_Toc514833746)

[5.6账户信息导出 15](#_Toc514833747)

[5.7部门信息导出 16](#_Toc514833748)

[5.8职位信息导出 17](#_Toc514833749)

[5.9订单信息打印 18](#_Toc514833750)

[5.10日志生成导出 18](#_Toc514833751)

[6模块设计 19](#_Toc514833752)

[6.1账户信息导入 19](#_Toc514833753)

[6.1.1业务概述 19](#_Toc514833754)

[6.1.2处理流程 19](#_Toc514833755)

[6.1.3输入界面 20](#_Toc514833756)

[6.1.4输入界面字段说明 20](#_Toc514833757)

[6.1.5功能说明 21](#_Toc514833758)

[6.2部门信息导入 21](#_Toc514833759)

[6.2.1业务概述 21](#_Toc514833760)

[6.2.2处理流程 21](#_Toc514833761)

[6.2.3输入界面 22](#_Toc514833762)

[6.2.4输入界面字段说明 22](#_Toc514833763)

[6.2.5功能说明 22](#_Toc514833764)

[6.3职位信息导入 23](#_Toc514833765)

[6.3.1业务概述 23](#_Toc514833766)

[6.3.2处理流程 23](#_Toc514833767)

[6.3.3输入界面 24](#_Toc514833768)

[6.3.4输入界面字段说明 24](#_Toc514833769)

[6.3.5功能说明 24](#_Toc514833770)

[6.4文件上传 25](#_Toc514833771)

[6.4.1业务概述 25](#_Toc514833772)

[6.4.2处理流程 25](#_Toc514833773)

[6.4.3输入界面 26](#_Toc514833774)

[6.4.4输入界面字段说明 26](#_Toc514833775)

[6.4.5功能说明 27](#_Toc514833776)

[6.5订单信息显示 27](#_Toc514833777)

[6.5.1业务概述 27](#_Toc514833778)

[6.5.2处理流程 27](#_Toc514833779)

[6.5.3输入界面 28](#_Toc514833780)

[6.5.4输入界面字段说明 28](#_Toc514833781)

[6.5.5功能说明 28](#_Toc514833782)

[6.6日志生成导出 29](#_Toc514833783)

[6.6.1业务概述 29](#_Toc514833784)

[6.6.2处理流程 29](#_Toc514833785)

[6.6.3输出页面 30](#_Toc514833786)

[6.6.4输出界面字段说明 30](#_Toc514833787)

[6.6.5功能说明 31](#_Toc514833788)

[6.7账户信息导出 31](#_Toc514833789)

[6.7.1业务概述 31](#_Toc514833790)

[6.7.2处理流程 32](#_Toc514833791)

[6.7.3输出页面 32](#_Toc514833792)

[6.7.4输出界面字段说明 33](#_Toc514833793)

[6.7.5功能说明 34](#_Toc514833794)

[6.8部门信息导出 34](#_Toc514833795)

[6.8.1业务概述 34](#_Toc514833796)

[6.8.2处理流程 35](#_Toc514833797)

[6.8.3输出页面 35](#_Toc514833798)

[6.8.4输出界面字段说明 36](#_Toc514833799)

[6.8.5功能说明 37](#_Toc514833800)

[6.9职位信息导出 37](#_Toc514833801)

[6.9.1业务概述 37](#_Toc514833802)

[6.9.2处理流程 38](#_Toc514833803)

[6.9.3输出页面 38](#_Toc514833804)

[6.9.4输出界面字段说明 39](#_Toc514833805)

[6.9.5功能说明 40](#_Toc514833806)

[6.10文件下载 40](#_Toc514833807)

[6.10.1业务概述 40](#_Toc514833808)

[6.10.2处理流程 40](#_Toc514833809)

[6.10.3输出页面 41](#_Toc514833810)

[6.10.4输出界面字段说明 42](#_Toc514833811)

[6.10.5功能说明 42](#_Toc514833812)

[6.11订单信息导出 42](#_Toc514833813)

[6.11.1业务概述 42](#_Toc514833814)

[6.11.2处理流程 43](#_Toc514833815)

[6.11.3输出页面 43](#_Toc514833816)

[6.11.4输出界面字段说明 44](#_Toc514833817)

[6.11.5功能说明 44](#_Toc514833818)

[7代码实现 44](#_Toc514833819)

[8个人总结 55](#_Toc514833820)

# 1引言

## 1.1背景

处于21世纪的我们，身边充斥着各色的电子产品，对于普遍大众来说，网上购物成了日益发展的需要，随着淘宝京东等购物平台的兴起，给老百姓带来的购物的便利与实惠，但是随之而来的问题是，自己不需要的生活用品，丢了可惜，又苦于没有合适的价格出手，于是网上拍卖管理系统应运而生。

## 1.2前景

目前市场上二手市场的前景广阔，普通大众对于生活用品的价格回收率的高追求，决定了这个行业的可挖掘性，前景很好。

## 1.3概述

本文为网上拍卖系统工具类excelUtil模块功能系统设计文档。

根据《OAS-excelUtil-工具类导入导出需求规格说明书》进行功能和体系结构分析设计。

## 1.4适用范围

适用于网上拍卖系统工具类excelUtil相关模块分析和设计过程

## 1.5读者对象

系统开发以及测试人员

# 2设计约束

## 2.1需求约束

本系统应当遵循网上拍卖系统的“开发框架实践手册(Java版)”

软件、硬件环境：Eclipse开发平台、Tomcat、Mysql数据库；

软件质量的约束，如正确性、健壮性、可靠性、效率（性能）、易用性、清晰性、安全性、可扩展性、兼容性、可移植性等等。

## 2.2设计策略

* 扩展策略：为实现将来数据项扩展，各数据表应留有备用字段长度，一面扩展数据造成空间不足以至于数据表重新建立
* 复用策略：本系统使用以服务程序为基本构件的设计方案，每个服务程序完成一个简单功能，在各个功能流程中可重复使用服务程序
* 折衷策略：为实现扩展功能及相关复杂流程，允许牺牲部分空间以获取时间及开发效率

## 2.3技术实现

本系统使用eclipse平台的Java语言

数据库使用MySQL语言

# 3系统分析

## 3.1总体描述

### 3.1.1产品功能

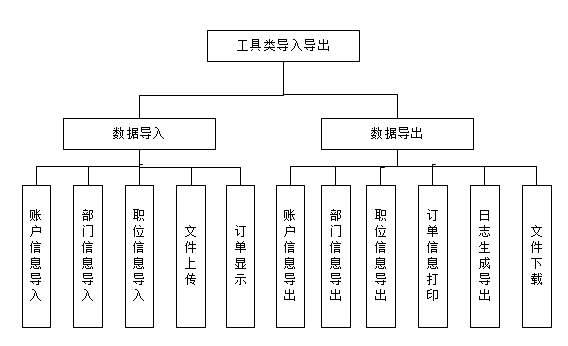


图3.1.1 功能模块图

### 3.1.2目标

|  |  |
| --- | --- |
| 需求名称 | 详细功能实现要求 |
| 数据导入 | 账户信息导入（卖家买家等） |
| 部门信息导入 |
| 职位信息导入 |
| 文件上传 |
| 订单信息显示 |
| 数据导出 | 账户信息导出（卖家买家等） |
| 部门信息导出 |
| 职位信息导出 |
| 订单信息打印 |
| 日志生成导出 |
| 文件下载 |

### 3.1.3用户特点

用户人群有两类：用户和管理员。

用户：注册账号，登录账号进行对活动信息的操作

管理员：系统的维护和对用户数据的管理和操作。

### 3.1.4运行环境

软件平台：

系统：Windows 7及以上

开发工具：Eclipse neon 4.6

数据库：Mysql

服务器：Tomcat9.0

硬件平台：

硬盘： 500G以上；

运行内存： 8G以上；

CPU：133MHz以上

# 4系统设计

## 4.1系统总体设计

### 4.1.1总体设计目标

<将系统分解为若干模块，绘制功能逻辑图，说明各模块的主要功能,各模块如何协调工作，从而实现原系统的功能。可以用功能逻辑图表示。>

<建议分层描述本系统功能模块，功能模块可包含面向用户需求的功能模块，也可包含面向实现的功能模块（如实现数据库的通用访问等），层次建议两到三层。

### 4.1.2总体架构

前端：Bootstrap框架、HTML5+CSS3、JavaScript、Ajax

后台：spring-MVC框架+spring+mybatis

算法：AES加密算法、MD5加密算法、路径规划算法

## 4.2业务逻辑分析与设计

### 4.2.1功能模块图

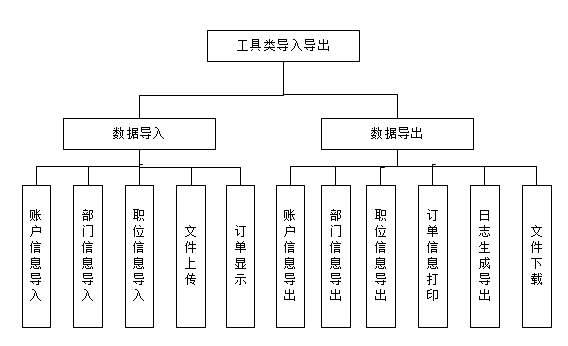


图4-2-1 功能模块图

### 4.2.2数据流图

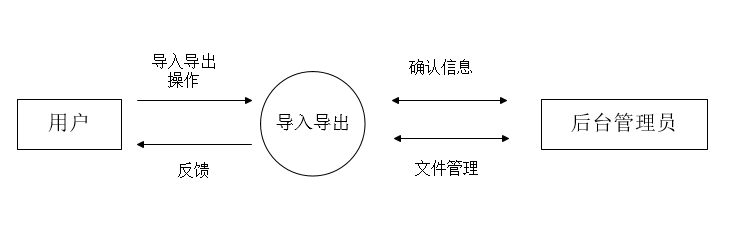


图4-2-2数据流图

### 4.2.3E-R实体关系图

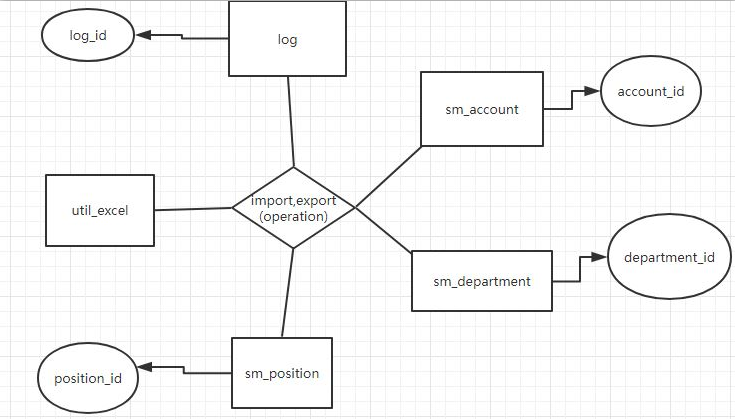


图4-2-3E-R实体关系图

# 5数据库设计

## 5.1账户信息导入

**数据结构：(详细见账户信息表)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 文件字段名称 | 长度 | 类型 | 说明 |
| 编号 | user\_id | 12 | int |  |
| 账号 | user\_account | 60 | varchar |  |
| 姓名 | user\_name | 60 | varchar |  |
| 性别 | user\_gender | 1 | varchar |  |
| 省份 | user\_province | 20 | varchar |  |
| 城市 | user\_city | 20 | varchar |  |
| 地区 | user\_district | 20 | varchar |  |
| 地址 | user\_address | 20 | varchar |  |
| 电话 | user\_phone | 100 | varchar |  |
| 座机 | user\_telephone | 50 | varchar |  |
| 角色 | user\_role\_id | 50 | varchar |  |
| 职位 | user\_position\_id | 12 | int |  |
| 部门 | user\_department\_id | 12 | varchar |  |
| 公司 | user\_company | 12 | varchar |  |
| 创建日期 | User\_created\_time |  | datetime |  |
| 修改日期 | user\_updated\_time |  | datetime |  |
| 导入日期 | user\_import\_time |  | datetime |  |

## 5.2部门信息导入

**数据结构：（详细见部门管理概要设计书）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 文件字段名称 | 长度 | 类型 | 说明 |
| 部门编号 | department\_id |  | int |  |
| 部门名称 | department\_name | 60 | varchar |  |
| 部门描述 | department\_comment | 1000 | varchar |  |
| 创建日期 | department\_created\_datetime |  | datetime |  |
| 创建人名 | department\_created\_name |  | int |  |
| 修改日期 | department\_updated\_datetime |  | datetime |  |
| 修改人名 | department\_updated\_name |  | int |  |
| 默认字段1 | department\_default1 | 20 | varchar |  |
| 默认字段2 | department\_default2 | 20 | varchar |  |
| 是否有效 | department\_isValid |  | boolean |  |

## 5.3职位信息导入

**数据结构：（详细见职位系统概要书）**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 文件字段名称 | 长度 | 类型 | 说明 |
| 职位编号 | postion\_id |  | int |  |
| 职位名称 | postion\_name | 60 | varchar |  |
| 职位描述 | postion\_comment | 1000 | varchar |  |
| 职位等级 | postion\_level | 20 | varchar |  |
| 部门编号 | position\_department\_id |  | int |  |
| 创建日期 | postion\_created\_datetime |  | datetime |  |
| 创建人名 | postion\_created\_name |  | int |  |
| 修改日期 | postion\_updated\_datetime |  | datetime |  |
| 修改人名 | postion\_updated\_name |  | int |  |
| 默认字段1 | postion\_default1 | 20 | varchar |  |
| 默认字段2 | postion\_default2 | 20 | varchar |  |
| 是否有效 | postion\_isValid |  | boolean |  |

## 5.4文件上传

**数据结构：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 文件字段名称 | 长度 | 类型 | 说明 |
| 文件编号 | File\_id |  | int |  |
| 文件名称 | File\_name | 20 | Varchar |  |
| 创建人 | File\_createdName |  | int |  |
| 创建日期 | File\_createdtime |  | date |  |
| 文件大小 | File\_storage |  | int |  |

## 5.5订单信息显示

**数据结构：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 文件字段名称 | 长度 | 类型 | 说明 |
| 订单id | Id |  | Int | 自增非空 |
| 订单编号 | Order\_code | 60 | varchar |  |
| 商品编号 | Goods\_id | 60 | varchar |  |
| 用户编号 | User\_id | 60 | varchar |  |

## 5.6账户信息导出

**数据结构：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 文件字段名称 | 长度 | 类型 | 说明 |
| 编号 | user\_id | 12 | int |  |
| 账号 | user\_account | 60 | varchar |  |
| 姓名 | user\_name | 60 | varchar |  |
| 性别 | user\_gender | 1 | varchar |  |
| 省份 | user\_province | 20 | varchar |  |
| 城市 | user\_city | 20 | varchar |  |
| 地区 | user\_district | 20 | varchar |  |
| 地址 | user\_address | 20 | varchar |  |
| 电话 | user\_phone | 100 | varchar |  |
| 座机 | user\_telephone | 50 | varchar |  |
| 角色 | user\_role\_id | 50 | varchar |  |
| 职位 | user\_position\_id | 12 | int |  |
| 部门 | user\_department\_id | 12 | varchar |  |
| 公司 | user\_company | 12 | varchar |  |
| 创建日期 | User\_created\_time |  | datetime |  |
| 修改日期 | user\_updated\_time |  | datetime |  |
| 导入日期 | user\_import\_time |  | datetime |  |

## 5.7部门信息导出

**数据结构：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 文件字段名称 | 长度 | 类型 | 说明 |
| 部门编号 | department\_id |  | int |  |
| 部门名称 | department\_name | 60 | varchar |  |
| 部门描述 | department\_comment | 1000 | varchar |  |
| 创建日期 | department\_created\_datetime |  | datetime |  |
| 创建人名 | department\_created\_name |  | int |  |
| 修改日期 | department\_updated\_datetime |  | datetime |  |
| 修改人名 | department\_updated\_name |  | int |  |
| 默认字段1 | department\_default1 | 20 | varchar |  |
| 默认字段2 | department\_default2 | 20 | varchar |  |
| 是否有效 | department\_isValid |  | boolean |  |

## 5.8职位信息导出

**数据结构：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 文件字段名称 | 长度 | 类型 | 说明 |
| 职位编号 | postion\_id |  | int |  |
| 职位名称 | postion\_name | 60 | varchar |  |
| 职位描述 | postion\_comment | 1000 | varchar |  |
| 职位等级 | postion\_level | 20 | varchar |  |
| 部门编号 | position\_department\_id |  | int |  |
| 创建日期 | postion\_created\_datetime |  | datetime |  |
| 创建人名 | postion\_created\_name |  | int |  |
| 修改日期 | postion\_updated\_datetime |  | datetime |  |
| 修改人名 | postion\_updated\_name |  | int |  |
| 默认字段1 | postion\_default1 | 20 | varchar |  |
| 默认字段2 | postion\_default2 | 20 | varchar |  |
| 是否有效 | postion\_isValid |  | boolean |  |

## 5.9订单信息打印

**数据结构：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 文件字段名称 | 长度 | 类型 | 说明 |
| 订单id | Id |  | Int | 自增非空 |
| 订单编号 | Order\_code | 60 | varchar |  |
| 商品编号 | Goods\_id | 60 | varchar |  |
| 用户编号 | User\_id | 60 | varchar |  |

## 5.10日志生成导出

**数据结构：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 文件字段名称 | 长度 | 类型 | 说明 |
| 编号 | Log\_Id |  | Int |  |
| 账户姓名 | Account\_name | 20 | Varchar |  |
| 登录ip | Login\_ip | 20 | varchar |  |
| 操作类型 | Type |  | Int |  |
| 操作时间 | Createtime |  | Date |  |
| 备注 | Remark | 500 | Varchar |  |

# 6模块设计

## 6.1账户信息导入

### 6.1.1业务概述

账户建账户信息，如果该账户在本行已开户，则客户的基础信息已存在，否则需新创建账户信息，主要包括编号、账号、姓名、性别、省份、城市、地区、地址、电话、座机、角色、职位、部门、公司、创建日期、修改日期、导入日期等基本信息。

### 6.1.2处理流程

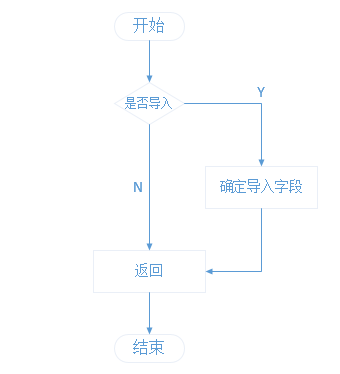


图6-1-2 账户信息导入流程图

### 6.1.3输入界面

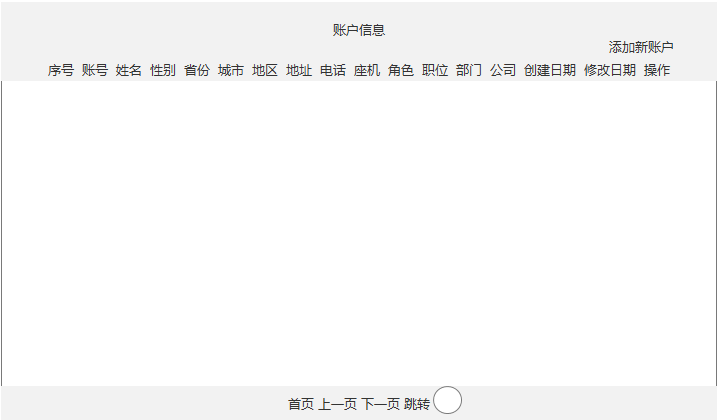


图6-1-3 输入界面图

### 6.1.4输入界面字段说明

【用户账号】：默认值为数据库中数据，无法更改

【姓名】：默认值为数据库中数据，无法更改

【性别】：默认值为数据库中数据，可输入，必输项。

【省份】：默认值为数据库中数据，可输入，必输项,直辖市可为空。

【市区】：默认值为数据库中数据，可输入，必输项。

【区县】：默认值为数据库中数据，可输入，必输项。

【地址】：默认值为数据库中数据，可输入，必输项。

【手机号】：默认值为数据库中数据，可输入，必输项。

【座机电话】：默认值为数据库中数据，可输入，选填项。

【职位】：默认值为数据库中数据，可输入，选填项。

【部门】：默认值为数据库中数据，可输入，选填项。

【公司】：默认值为数据库中数据，可输入，选填项。

### 6.1.5功能说明

点击【账户信息】按钮：即可出现所有账户的相关信息

## 6.2部门信息导入

### 6.2.1业务概述

部门信息主要包括以下字段：部门编号、部门名称、部门描述、创建日期、创建人名、修改日期、修改人名、是否有效等。

### 6.2.2处理流程

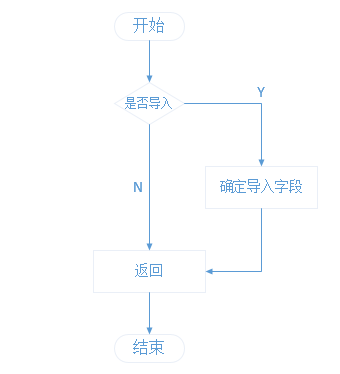


图6-2-2 部门信息导入流程图

### 6.2.3输入界面

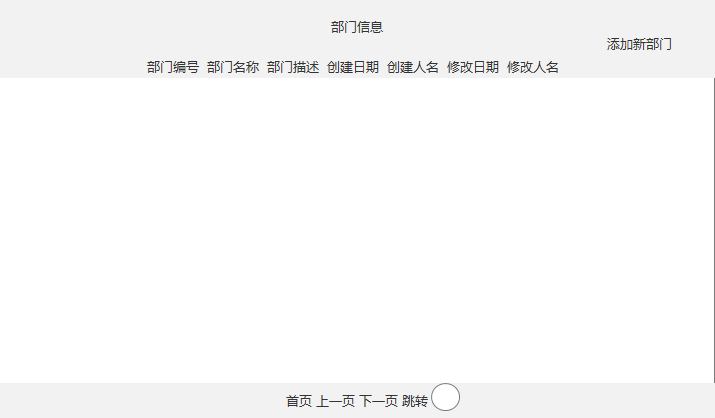


图6-2-3 输入界面图

### 6.2.4输入界面字段说明

【部门编号】：默认值为数据库中数据，无法更改

【部门名称】：默认值为数据库中数据，无法更改

【部门描述】：默认值为数据库中数据，可输入，必输项。

【创建日期】：默认值为数据库中数据，可输入，必输项

【创建人名】：默认值为数据库中数据，可输入，必输项。

【修改日期】：默认值为数据库中数据，可输入，必输项。

【修改人名】：默认值为数据库中数据，可输入，必输项。

### 6.2.5功能说明

点击【部门信息】按钮：即可出现所有部门的相关信息

## 6.3职位信息导入

### 6.3.1业务概述

职位信息包括以下字段：职位名称、职位描述、职位等级、职位编号、创建日期、创建人名、修改日期、修改人名、是否有效等。

### 6.3.2处理流程

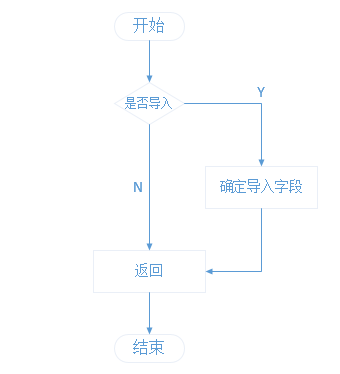


图6-3-2 职位信息导入流程图

### 6.3.3输入界面

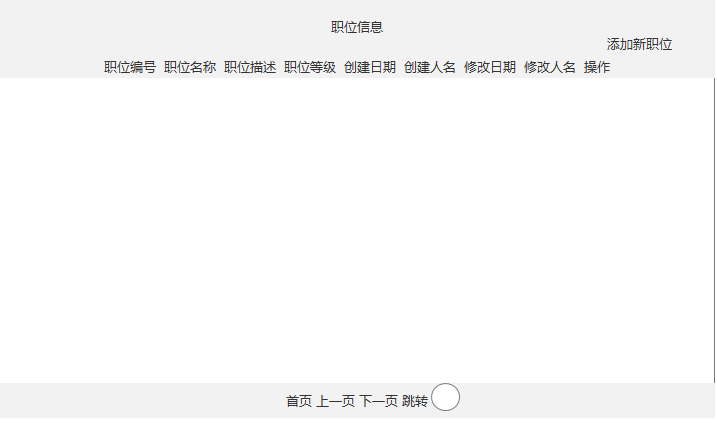


图6-3-3 输入界面图

### 6.3.4输入界面字段说明

【职位编号】：默认值为数据库中数据，无法更改

【职位名称】：默认值为数据库中数据，无法更改

【职位描述】：默认值为数据库中数据，可输入，必输项。

【职位等级】：默认值为数据库中数据，可输入，必输项。

【创建日期】：默认值为数据库中数据，可输入，必输项。

【创建人名】：默认值为数据库中数据，可输入，必输项。

【修改日期】：默认值为数据库中数据，可输入，必输项。

【修改人名】：默认值为数据库中数据，可输入，必输项。

### 6.3.5功能说明

点击【职位信息】按钮：即可出现所有职位的相关信息

## 6.4文件上传

### 6.4.1业务概述

文件包括以下字段：文件编号、文件名称、创建人、创建日期、文件大小等。

### 6.4.2处理流程

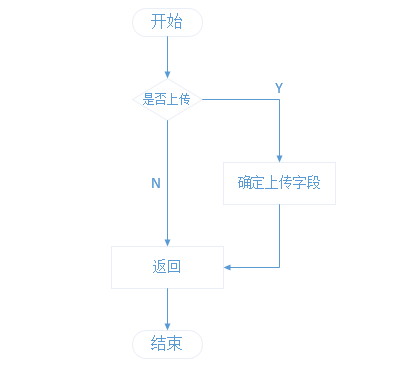


图6-4-2 文件上传流程图

### 6.4.3输入界面

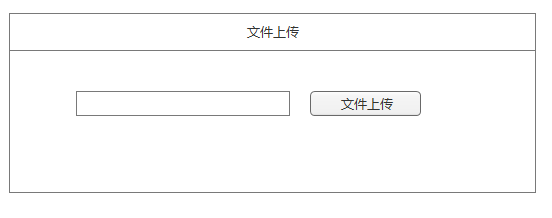


图6-4-3 （1）输入界面

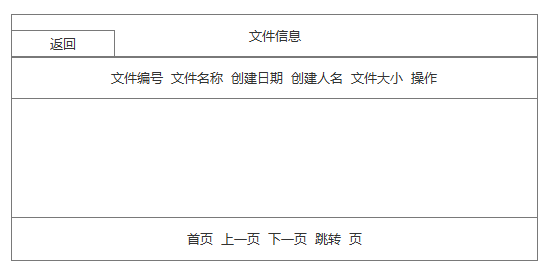


图6-4-3（2）输入界面

### 6.4.4输入界面字段说明

【文件编号】：默认值为数据库中数据，无法更改

【文件名称】：默认值为数据库中数据，无法更改

【创建日期】：默认值为数据库中数据，可输入，必输项。

【创建人名】：默认值为数据库中数据，可输入，必输项。

【文件大小】：默认值为数据库中数据，可输入，必输项。

### 6.4.5功能说明

点击【文件上传】按钮：即可出现上传文件

## 6.5订单信息显示

### 6.5.1业务概述

订单信息包括以下字段：订单id、订单编号、商品编号和用户编号。

### 6.5.2处理流程

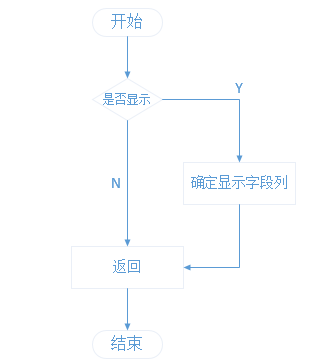


图6-5-2 流程图

### 6.5.3输入界面



图6-5-3 输入界面图

### 6.5.4输入界面字段说明

【订单id】：默认值为数据库中数据，无法更改

【订单编号】：默认值为数据库中数据，无法更改

【商品编号】：默认值为数据库中数据，可输入，必输项。

【用户编号】：默认值为数据库中数据，可输入，必输项。

### 6.5.5功能说明

点击【订单信息】按钮：即可出现订单的相关信息

## 6.6日志生成导出

### 6.6.1业务概述

日志包括以下字段：编号、账户姓名、登录ip、操作类型、操作时间、备注等。

### 6.6.2处理流程

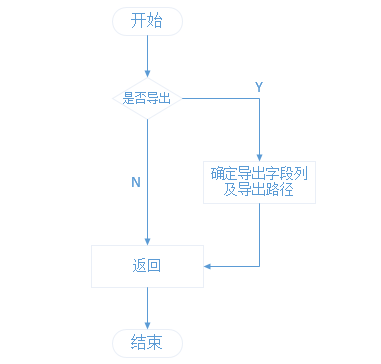


图6-6-2 日志生成导出流程图

### 6.6.3输出页面



图6-6-3 （1）输出页面

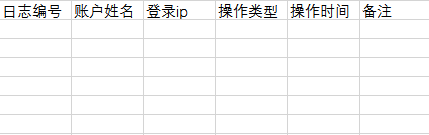


图6-6-3 （2）输出页面

### 6.6.4输出界面字段说明

【日志编号】：默认值为数据库中数据，无法更改

【账户姓名】：默认值为数据库中数据，无法更改

【登录ip】：默认值为数据库中数据，可输入，必输项。

【操作类型】：默认值为数据库中数据，可输入，必输项。

【操作时间】：默认值为数据库中数据，可输入，必输项。

【备注】：默认值为数据库中数据，可输入，必输项。

### 6.6.5功能说明

点击【日志导出】按钮：即可出现日志的相关信息导出到本地

## 6.7账户信息导出

### 6.7.1业务概述

账户建账户信息，如果该账户在本行已开户，则客户的基础信息已存在，否则需新创建账户信息，主要包括编号、账号、姓名、性别、省份、城市、地区、地址、电话、座机、角色、职位、部门、公司、创建日期、修改日期、导入日期等基本信息。

### 6.7.2处理流程

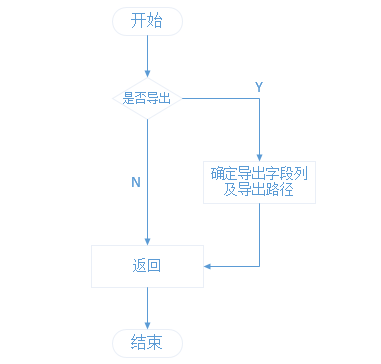


图6-7-2 流程图

### 6.7.3输出页面



图6-7-3 （1）输出页面

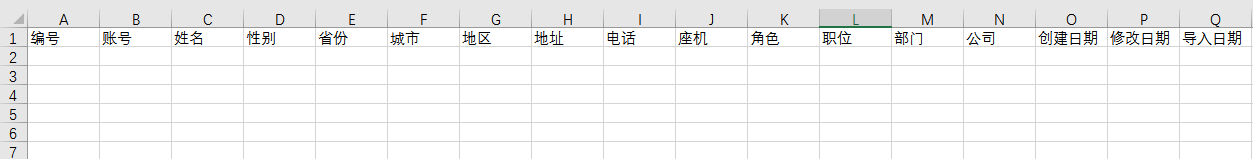


图6-7-3 （2）输出页面

### 6.7.4输出界面字段说明

【用户账号】：默认值为数据库中数据，无法更改

【姓名】：默认值为数据库中数据，无法更改

【性别】：默认值为数据库中数据，可输入，必输项。

【省份】：默认值为数据库中数据，可输入，必输项,直辖市可为空。

【市区】：默认值为数据库中数据，可输入，必输项。

【区县】：默认值为数据库中数据，可输入，必输项。

【地址】：默认值为数据库中数据，可输入，必输项。

【手机号】：默认值为数据库中数据，可输入，必输项。

【座机电话】：默认值为数据库中数据，可输入，选填项。

【职位】：默认值为数据库中数据，可输入，选填项。

【部门】：默认值为数据库中数据，可输入，选填项。

【公司】：默认值为数据库中数据，可输入，选填项。

### 6.7.5功能说明

点击【账户信息导出】按钮：即可导出所有账户的相关信息

## 6.8部门信息导出

### 6.8.1业务概述

部门信息主要包括以下字段：部门编号、部门名称、部门描述、创建日期、创建人名、修改日期、修改人名、是否有效等。

### 6.8.2处理流程

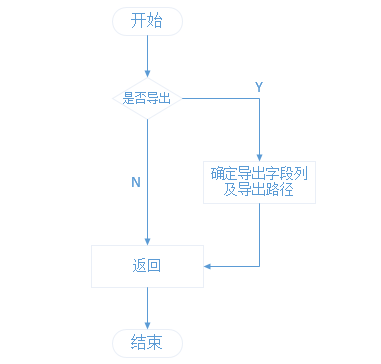


图6-8-2 流程图

### 6.8.3输出页面



图6-8-3 （1）输出页面

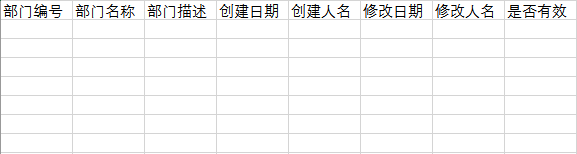


图6-8-3 （2）输出页面

### 6.8.4输出界面字段说明

【部门编号】：默认值为数据库中数据，无法更改

【部门名称】：默认值为数据库中数据，无法更改

【部门描述】：默认值为数据库中数据，可输入，必输项。

【创建日期】：默认值为数据库中数据，可输入，必输项

【创建人名】：默认值为数据库中数据，可输入，必输项。

【修改日期】：默认值为数据库中数据，可输入，必输项。

【修改人名】：默认值为数据库中数据，可输入，必输项。

### 6.8.5功能说明

点击【部门信息导出】按钮：即可导出所有部门的相关信息

## 6.9职位信息导出

### 6.9.1业务概述

职位信息包括以下字段：职位编号、职位名称、职位描述、职位等级、创建日期、创建人名、修改日期、修改人名、是否有效等。

### 6.9.2处理流程

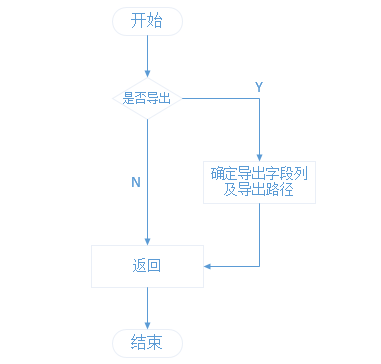


图6-9-2 流程图

### 6.9.3输出页面



图6-9-3 （1）输出页面

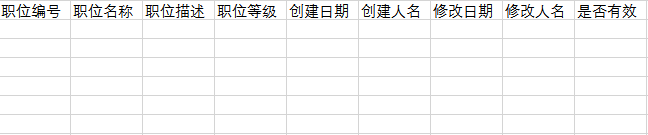


图6-9-3 （2）输出页面

### 6.9.4输出界面字段说明

【职位编号】：默认值为数据库中数据，无法更改

【职位名称】：默认值为数据库中数据，无法更改

【职位描述】：默认值为数据库中数据，可输入，必输项。

【职位等级】：默认值为数据库中数据，可输入，必输项。

【创建日期】：默认值为数据库中数据，可输入，必输项。

【创建人名】：默认值为数据库中数据，可输入，必输项。

【修改日期】：默认值为数据库中数据，可输入，必输项。

【修改人名】：默认值为数据库中数据，可输入，必输项。

### 6.9.5功能说明

点击【职位信息导出】按钮：即可出现所有职位的相关信息

## 6.10文件下载

### 6.10.1业务概述

文件包括以下字段：文件编号、文件名称、创建人、创建日期、文件大小等。

### 6.10.2处理流程

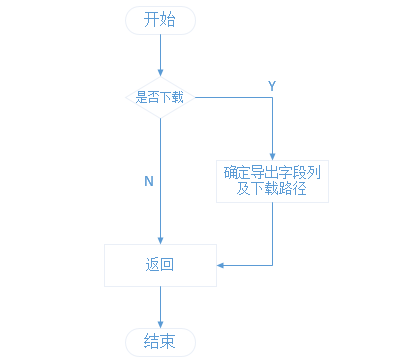


图6-10-2 流程图

### 6.10.3输出页面

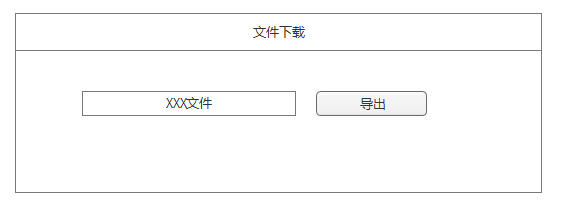


图6-10-3 （1）输出页面

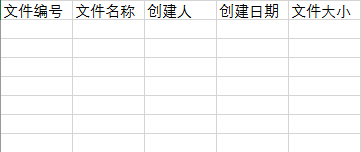


图6-10-3 （2）输出页面

### 6.10.4输出界面字段说明

【文件编号】：默认值为数据库中数据，无法更改

【文件名称】：默认值为数据库中数据，无法更改

【创建日期】：默认值为数据库中数据，可输入，必输项。

【创建人名】：默认值为数据库中数据，可输入，必输项。

【文件大小】：默认值为数据库中数据，可输入，必输项。

### 6.10.5功能说明

点击【文件下载】按钮：即可下载文件信息到本地。

## 6.11订单信息导出

### 6.11.1业务概述

订单信息包括以下字段：订单id、订单编号、商品编号和用户编号。

### 6.11.2处理流程

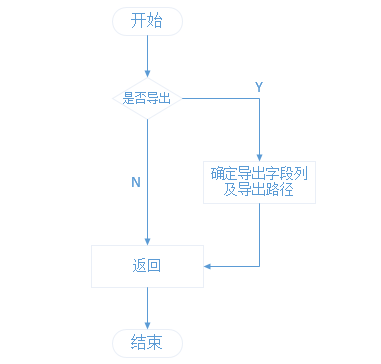


图6-11-2 流程图

### 6.11.3输出页面



图6-11-3 （1）输出页面

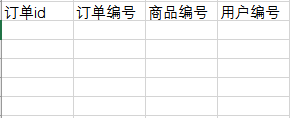


图6-11-3 （2）输出页面

### 6.11.4输出界面字段说明

【订单id】：默认值为数据库中数据，无法更改

【订单编号】：默认值为数据库中数据，无法更改

【商品编号】：默认值为数据库中数据，可输入，必输项。

【用户编号】：默认值为数据库中数据，可输入，必输项。

### 6.11.5功能说明

点击【订单信息打印】按钮：即可导出订单信息到本地。

# 7代码实现

附核心代码：

ExcelUtil.java

/\*\*

\* **@Title**: ExcelUtil.java

\* **@Package** com.clps.oas.util.excel

\* **@Description**: 实现工具类数据导入导出表格

\* **@author** weigion.wu(吴跟强)

\* **@date** 2018-05-21 20:30:23

\* **@version** V1.0

\*/

**package** com.clps.oas.util.excel;

**import** java.io.File;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** java.io.IOException;

**import** java.io.OutputStream;

**import** java.lang.reflect.Field;

**import** java.lang.reflect.InvocationTargetException;

**import** java.lang.reflect.Method;

**import** java.text.DecimalFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.ArrayList;

**import** java.util.Collection;

**import** java.util.Date;

**import** java.util.Iterator;

**import** java.util.List;

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**import** org.apache.poi.hssf.usermodel.HSSFCell;

**import** org.apache.poi.hssf.usermodel.HSSFCellStyle;

**import** org.apache.poi.hssf.usermodel.HSSFClientAnchor;

**import** org.apache.poi.hssf.usermodel.HSSFComment;

**import** org.apache.poi.hssf.usermodel.HSSFDateUtil;

**import** org.apache.poi.hssf.usermodel.HSSFFont;

**import** org.apache.poi.hssf.usermodel.HSSFPatriarch;

**import** org.apache.poi.hssf.usermodel.HSSFRichTextString;

**import** org.apache.poi.hssf.usermodel.HSSFRow;

**import** org.apache.poi.hssf.usermodel.HSSFSheet;

**import** org.apache.poi.hssf.usermodel.HSSFWorkbook;

**import** org.apache.poi.hssf.util.HSSFColor;

**import** org.apache.poi.xssf.usermodel.XSSFCell;

**import** org.apache.poi.xssf.usermodel.XSSFRow;

**import** org.apache.poi.xssf.usermodel.XSSFSheet;

**import** org.apache.poi.xssf.usermodel.XSSFWorkbook;

/\*\*

\* **@ClassName**: ExcelUtil

\* **@Description**: 实现工具类数据导入导出表格

\* **@author** weigion.wu

\* **@date** 2018-05-21 20:30:23

\*/

**public** **class** ExcelUtil<T> {

**public** **static** **final** String ***FILE\_SEPARATOR*** = System.*getProperties*().getProperty("file.separator");

**public** **void** exportToExcel(String title, String[] headers, Collection<T> dataset, OutputStream out) {

exportToExcel(title, headers, dataset, out, "yyyy-MM-dd");

}

@SuppressWarnings({ "deprecation", "rawtypes", "unchecked" })

**public** **void** exportToExcel(String title, String[] headers, Collection<T> dataset, OutputStream out, String pattern) {

// 声明一个工作薄

HSSFWorkbook workbook = **new** HSSFWorkbook();

// 生成一个表格

HSSFSheet sheet = workbook.createSheet(title);

// 设置表格默认列宽度为15个字节

sheet.~~setDefaultColumnWidth~~((**short**)15);

// 生成一个样式

HSSFCellStyle style = workbook.createCellStyle();

// 设置这些样式

style.setFillForegroundColor(HSSFColor.SKY\_BLUE.***index***);

style.setFillPattern(HSSFCellStyle.***SOLID\_FOREGROUND***);

style.setBorderBottom(HSSFCellStyle.***BORDER\_THIN***);

style.setBorderLeft(HSSFCellStyle.***BORDER\_THIN***);

style.setBorderRight(HSSFCellStyle.***BORDER\_THIN***);

style.setBorderTop(HSSFCellStyle.***BORDER\_THIN***);

style.setAlignment(HSSFCellStyle.***ALIGN\_CENTER***);

// 生成一个字体

HSSFFont font = workbook.createFont();

font.setColor(HSSFColor.VIOLET.***index***);

font.setFontHeightInPoints((**short**) 12);

font.setBoldweight(HSSFFont.***BOLDWEIGHT\_BOLD***);

// 把字体应用到当前的样式

style.setFont(font);

// 生成并设置另一个样式

HSSFCellStyle style2 = workbook.createCellStyle();

style2.setFillForegroundColor(HSSFColor.LIGHT\_YELLOW.***index***);

style2.setFillPattern(HSSFCellStyle.***SOLID\_FOREGROUND***);

style2.setBorderBottom(HSSFCellStyle.***BORDER\_THIN***);

style2.setBorderLeft(HSSFCellStyle.***BORDER\_THIN***);

style2.setBorderRight(HSSFCellStyle.***BORDER\_THIN***);

style2.setBorderTop(HSSFCellStyle.***BORDER\_THIN***);

style2.setAlignment(HSSFCellStyle.***ALIGN\_CENTER***);

style2.setVerticalAlignment(HSSFCellStyle.***VERTICAL\_CENTER***);

// 生成另一个字体

HSSFFont font2 = workbook.createFont();

font2.setBoldweight(HSSFFont.***BOLDWEIGHT\_NORMAL***);

// 把字体应用到当前的样式

style2.setFont(font2);

// 声明一个画图的顶级管理器

HSSFPatriarch patriarch = sheet.createDrawingPatriarch();

// 定义注释的大小和位置,详见文档

HSSFComment comment = patriarch.createComment(**new** HSSFClientAnchor(0, 0, 0, 0, (**short**) 4, 2, (**short**) 6, 5));

// 设置注释作者，当鼠标移动到单元格上是可以在状态栏中看到该内容.

comment.setAuthor("leno");

// 产生表格标题行

HSSFRow row = sheet.createRow(0);

**for** (**short** i = 0; i < headers.length; i++) {

HSSFCell cell = row.~~createCell~~(i);

cell.setCellStyle(style);

HSSFRichTextString text = **new** HSSFRichTextString(headers[i]);

cell.setCellValue(text);

}

// 遍历集合数据，产生数据行

Iterator<T> it = dataset.iterator();

**int** index = 0;

**while** (it.hasNext()) {

index++;

row = sheet.createRow(index);

T t = (T) it.next();

// 利用反射，根据javabean属性的先后顺序，动态调用getXxx()方法得到属性值

Field[] fields = t.getClass().getDeclaredFields();

**for** (**short** i = 0; i < fields.length; i++) {

HSSFCell cell = row.~~createCell~~(i);

cell.setCellStyle(style2);

Field field = fields[i];

String fieldName = field.getName();

String getMethodName = "get" + fieldName.substring(0, 1).toUpperCase() + fieldName.substring(1);

**try** {

Class tCls = t.getClass();

Method getMethod = tCls.getMethod(getMethodName, **new** Class[] {});

Object value = getMethod.invoke(t, **new** Object[] {});

// 判断值的类型后进行强制类型转换

String textValue = **null**;

**if** (value **instanceof** Boolean) {

**boolean** bValue = (Boolean) value;

textValue = "男";

**if** (!bValue) {

textValue = "女";

}

} **else** **if** (value **instanceof** Date) {

Date date = (Date) value;

SimpleDateFormat sdf = **new** SimpleDateFormat(pattern);

textValue = sdf.format(date);

} **else** **if** (value **instanceof** **byte**[]) {

// 有图片时，设置行高为60px;

row.setHeightInPoints(60);

// 设置图片所在列宽度为80px,注意这里单位的一个换算

sheet.~~setColumnWidth~~(i, (**short**) (35.7 \* 80));

// sheet.autoSizeColumn(i);

**byte**[] bsValue = (**byte**[]) value;

HSSFClientAnchor anchor = **new** HSSFClientAnchor(0, 0, 1023, 255, (**short**) 6, index, (**short**) 6,

index);

anchor.setAnchorType(2);

patriarch.createPicture(anchor, workbook.addPicture(bsValue, HSSFWorkbook.***PICTURE\_TYPE\_JPEG***));

} **else** {

// 其它数据类型都当作字符串简单处理

textValue = value.toString();

}

// 如果不是图片数据，就利用正则表达式判断textValue是否全部由数字组成

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

Pattern p = Pattern.*compile*("^//d+(//.//d+)?$");

Matcher matcher = p.matcher(textValue);

**if** (matcher.matches()) {

// 是数字当作double处理

cell.setCellValue(Double.*parseDouble*(textValue));

} **else** {

HSSFRichTextString richString = **new** HSSFRichTextString(textValue);

HSSFFont font3 = workbook.createFont();

font3.setColor(HSSFColor.BLUE.***index***);

richString.applyFont(font3);

cell.setCellValue(richString);

}

}

} **catch** (SecurityException e) {

e.printStackTrace();

} **catch** (NoSuchMethodException e) {

e.printStackTrace();

} **catch** (IllegalArgumentException e) {

e.printStackTrace();

} **catch** (IllegalAccessException e) {

e.printStackTrace();

} **catch** (InvocationTargetException e) {

e.printStackTrace();

} **finally** {

// 清理资源

}

}

}

**try** {

workbook.write(out);

} **catch** (IOException e) {

e.printStackTrace();

}

}

/\*\*

\* 对外提供读取excel 的方法

\*/

**public** List<Object> ImportFromExcel(File file, **int** sheet) **throws** IOException {

String fileName = file.getName();

String extension = fileName.lastIndexOf(".") == -1 ? "" : fileName.substring(fileName.lastIndexOf(".") + 1);

**if** ("xls".equals(extension)) {

**return** read2003Excel(file, sheet);

} **else** **if** ("xlsx".equals(extension)) {

**return** read2007Excel(file, sheet);

} **else** {

**throw** **new** IOException("不支持的文件类型");

}

}

/\*\*

\* 读取 office 2003 excel

\*

\* **@throws** IOException

\* **@throws** FileNotFoundException

\*/

**private** List<Object> read2003Excel(File file, **int** sheetIndex) **throws** IOException {

List<Object> list = **new** ArrayList<Object>();

HSSFWorkbook hwb = **new** HSSFWorkbook(**new** FileInputStream(file));

HSSFSheet sheet = hwb.getSheetAt(sheetIndex);

Object value = **null**;

HSSFRow row = **null**;

HSSFCell cell = **null**;

**int** counter = 0;

**for** (**int** i = sheet.getFirstRowNum(); counter < sheet.getPhysicalNumberOfRows(); i++) {

row = sheet.getRow(i);

**if** (row == **null**) {

**continue**;

} **else** {

counter++;

}

List<Object> linked = **new** ArrayList<Object>();

**for** (**int** j = row.getFirstCellNum(); j <= row.getLastCellNum(); j++) {

cell = row.getCell(j);

**if** (cell == **null**) {

**continue**;

}

DecimalFormat df = **new** DecimalFormat("0");// 格式化 number String

// 字符

SimpleDateFormat sdf = **new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss");// 格式化日期字符串

DecimalFormat nf = **new** DecimalFormat("0.00");// 格式化数字

**switch** (cell.getCellType()) {

**case** XSSFCell.***CELL\_TYPE\_STRING***:

value = cell.getStringCellValue();

**break**;

**case** XSSFCell.***CELL\_TYPE\_NUMERIC***:

**if** ("@".equals(cell.getCellStyle().getDataFormatString())) {

value = df.format(cell.getNumericCellValue());

} **else** **if** ("General".equals(cell.getCellStyle().getDataFormatString())) {

value = nf.format(cell.getNumericCellValue());

} **else** {

value = sdf.format(HSSFDateUtil.*getJavaDate*(cell.getNumericCellValue()));

}

**break**;

**case** XSSFCell.***CELL\_TYPE\_BOOLEAN***:

value = cell.getBooleanCellValue();

**break**;

**case** XSSFCell.***CELL\_TYPE\_BLANK***:

value = "";

**break**;

**default**:

value = cell.toString();

}

**if** (value == **null** || "".equals(value)) {

**continue**;

}

linked.add(value);

}

list.add(linked);

}

**return** list;

}

/\*\*

\* 读取Office 2007 excel

\*/

**private** List<Object> read2007Excel(File file, **int** sheetIndex) **throws** IOException {

List<Object> list = **new** ArrayList<Object>();

// 构造 XSSFWorkbook 对象，strPath 传入文件路径

XSSFWorkbook xwb = **new** XSSFWorkbook(**new** FileInputStream(file));

// 读取第一章表格内容

XSSFSheet sheet = xwb.getSheetAt(sheetIndex);

Object value = **null**;

XSSFRow row = **null**;

XSSFCell cell = **null**;

**int** counter = 0;

**for** (**int** i = sheet.getFirstRowNum(); counter < sheet.getPhysicalNumberOfRows(); i++) {

row = sheet.getRow(i);

**if** (row == **null**) {

**continue**;

} **else** {

counter++;

}

List<Object> linked = **new** ArrayList<Object>();

**for** (**int** j = row.getFirstCellNum(); j <= row.getLastCellNum(); j++) {

cell = row.getCell(j);

**if** (cell == **null**) {

value = "";

linked.add(value);

**continue**;

}

DecimalFormat df = **new** DecimalFormat("0");// 格式化 number String

// 字符

SimpleDateFormat sdf = **new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss");// 格式化日期字符串

DecimalFormat nf = **new** DecimalFormat("0");// 格式化数字

**switch** (cell.getCellType()) {

**case** XSSFCell.***CELL\_TYPE\_STRING***:

value = cell.getStringCellValue();

**break**;

**case** XSSFCell.***CELL\_TYPE\_NUMERIC***:

**if** ("@".equals(cell.getCellStyle().getDataFormatString())) {

value = df.format(cell.getNumericCellValue());

} **else** **if** ("General".equals(cell.getCellStyle().getDataFormatString())) {

value = nf.format(cell.getNumericCellValue());

} **else** {

value = sdf.format(HSSFDateUtil.*getJavaDate*(cell.getNumericCellValue()));

}

**break**;

**case** XSSFCell.***CELL\_TYPE\_BOOLEAN***:

value = cell.getBooleanCellValue();

**break**;

**case** XSSFCell.***CELL\_TYPE\_BLANK***:

value = "";

**break**;

**default**:

value = cell.toString();

}

linked.add(value);

}

list.add(linked);

}

**return** list;

}

}

# 8个人总结

通过这段时间的项目实战学习，我们组终于完成了网上拍卖系统的设计，系统在设计方面具有可优化功能，这个系统设计不是很复杂，但是要优化其实并不容易，在我们的项目中，我被分到的模块是实现工具类导入导出功能，在这个过程中，我学习到了很多东西，同时也在老师身边学习到了开发系统的一些技巧和方法，对我以后的学习给予了很大帮助，以前只是喜欢程序，学习了不少语言，但是，我还真没有独立设计过大型数据库，还没有系统的团队协作做过项目，这个过程则是理论和实践的有机结合，为我以后的发展提供了很大的帮助。

希望以后有更多这样的机会来提升自己，加油！