



**《Java程序课程设计》**

**（2024/2025学年第一学期）**

**指导教师：任祝、陶灵兵**

**班级：23通信工程2班**

**学号：2023329600143**

**姓名：唐韩宇**

设计报告

# 一、设计题目：

**题目：**个人记账管理系统

设计并实现一个个人记账系统。系统应包含基础的个人基本信息管理、支出信息登记、收入信息登记等功能，实现每月按类别收入、支出统计报表等统计功能。

**基础功能：**

1. 用户操作界面:要求界面逻辑清晰、友好、美观、易用；
2. 可记录多个人的记账信息（唯一标识学号）
3. 支出信息按分类进行登记
4. 实现按月统计收支统计报表功能

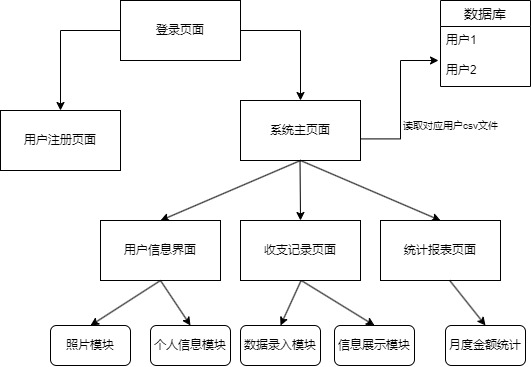
**拓展功能：**

1. 用户注册页面；
2. 用户照片显示、以及默认照片设定；
3. 多账户设置、历史数据导入；
4. 用户修改密码界面；

5、数据导出csv文件，以及UI删除记录。

# 二、系统的整体设计思路：

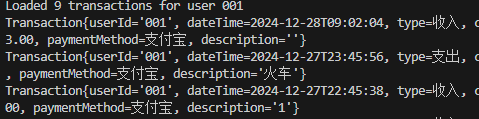
本程序的结构框图如下图展示：



**图 1 程序框架图**

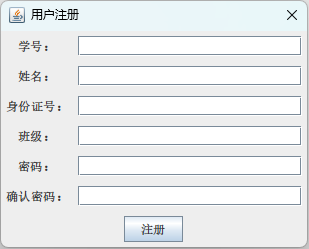
设计思路如下：

1. 程序结构分析，本软件需要设计一个学生记账程序，包含登录页面、用户注册页面、用户信息界面、收支记录页面和统计报表页面。
2. 考虑到**多用户的需求**，为了实现多用户数据的独立，以及操作的方便，本软件选择了在用户注册的时候，自动生成一个transactions\_学号.csv文件，用于存储每个用户的数据。由于DataManager类方法大部分为私有，为了在注册时调用，创建了一个public方法createTransactionFileForNewUser(String userId)，通过调用私有方法createFileIfNotExists(String fileName)实现在注册时自动生成数据文件。在用户登入和退出时会自动读取数据、保存数据。



**图 2数据读取在终端中的显示**

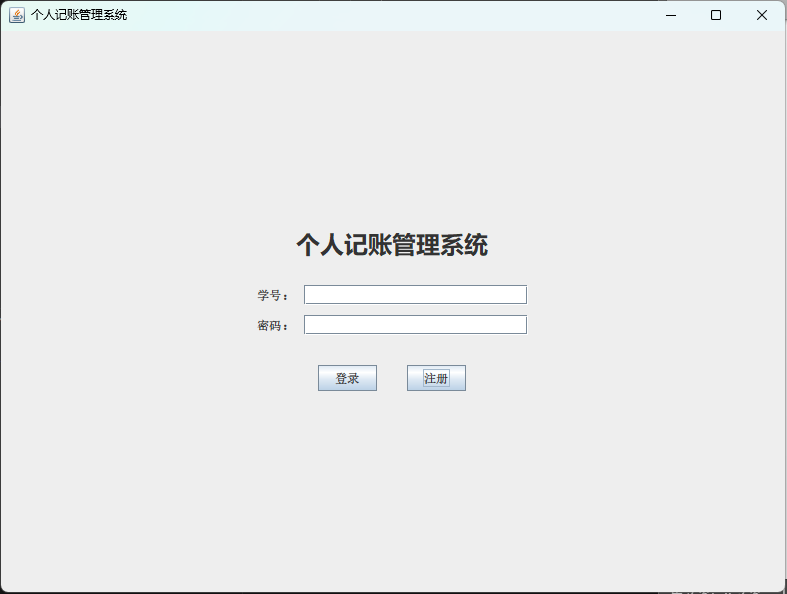
1. 用户注册界面，为了有良好的用户使用，使用JDialog创建一个模态对话框，确保用户在完成注册之前无法与主窗口交互，创建多个JTextField和JPasswordField用于用户输入获取用户信息，并添加一个“注册”按钮，用户点击后触发注册逻辑。



**图 3用户注册界面**

在用户注册界面针对输入合法性进行了以下操作：

1. 所有信息都需要填写完整
2. 两次密码需要一样
3. 注册的学号需要没有注册过
4. 主页面设计



**图 4登录界面**

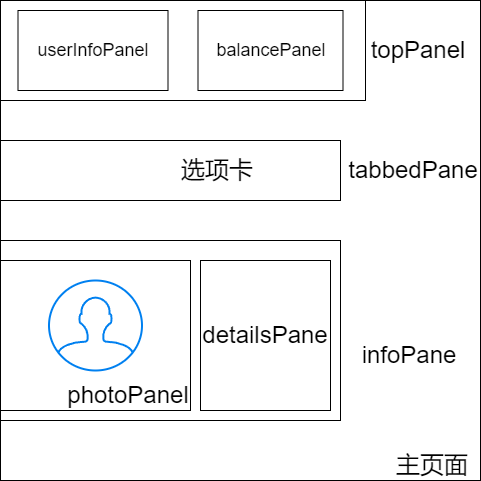
通过将用户输入的数据，与所有用户的信息存储的users.csv文件中的数据进行比较，来判断用户是否有登录权限。当用户输入错误时，也会有登录失败的提示，方便用户交互。



**图 5登录失败提示**

1. 登录后页面的简介

登入后的页面主要由几个部分组成。其中infoPane区域由JTabbedPane中的几个模块展示。



**图 6布局分布图**

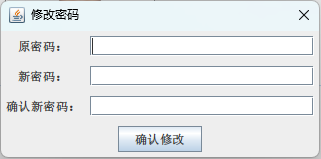
1. 用户信息展示页面



**图 7用户信息界面**

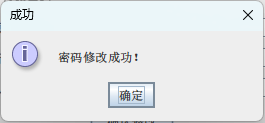
在用户信息页面分为照片模块和个人信息模块，在照片设置中，设定了一个默认照片，然后根据文件名由id转换，实现了不同用户登录展示不同照片的功能。个人信息则是读取库中的数据展现。

1. 修改密码界面



**图 8修改密码界面**

使用JDialog创建一个模态对话框，确保用户在完成密码修改之前无法与主窗口交互，采用GridBagLayout布局管理器，使得组件的排列灵活且可适应不同的屏幕大小。通过userManager.validateUser方法验证用户输入的原密码是否正确。并对新密码进行了一致性检查，确保新密码与确认新密码一致。确保了用户输入的安全。同时修改密码成功后有相应的提示。



**图 9修改密码成功提示界面**

1. 收支纪录页面



**图 10收支记录界面**

本页面使用了下拉选进行数据填写的功能，该功能实现还用到了Type枚举类，实现类别的确定。包含下拉框（JComboBox）用于选择交易类型、类别和支付方式，文本框（JTextField）用于输入金额和描述。此处对输入金额非负性进行了判断。

使用 JTable 显示交易记录，表格不可编辑。底部按钮面板包含“导出记录”和“删除记录”按钮。

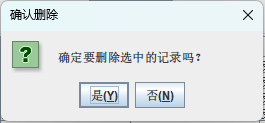
1. 导出记录: 允许用户选择文件位置导出交易记录到 CSV 文件。

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

**图 11数据导出**

2）删除记录: 提供删除选中交易记录的功能，并在删除前进行确认。



**图 12删除记录确认界面**

其具体实现为：

1）BufferedWriter: 用于写入文本文件，提供高效的字符输出。

2）FileWriter: 用于创建和写入文件。

3）表头: 首先写入 CSV 文件的表头，列出每个字段的名称。记录循环: 遍历用户的交易记录，将每条交易的信息格式化为 CSV 格式并写入文件。

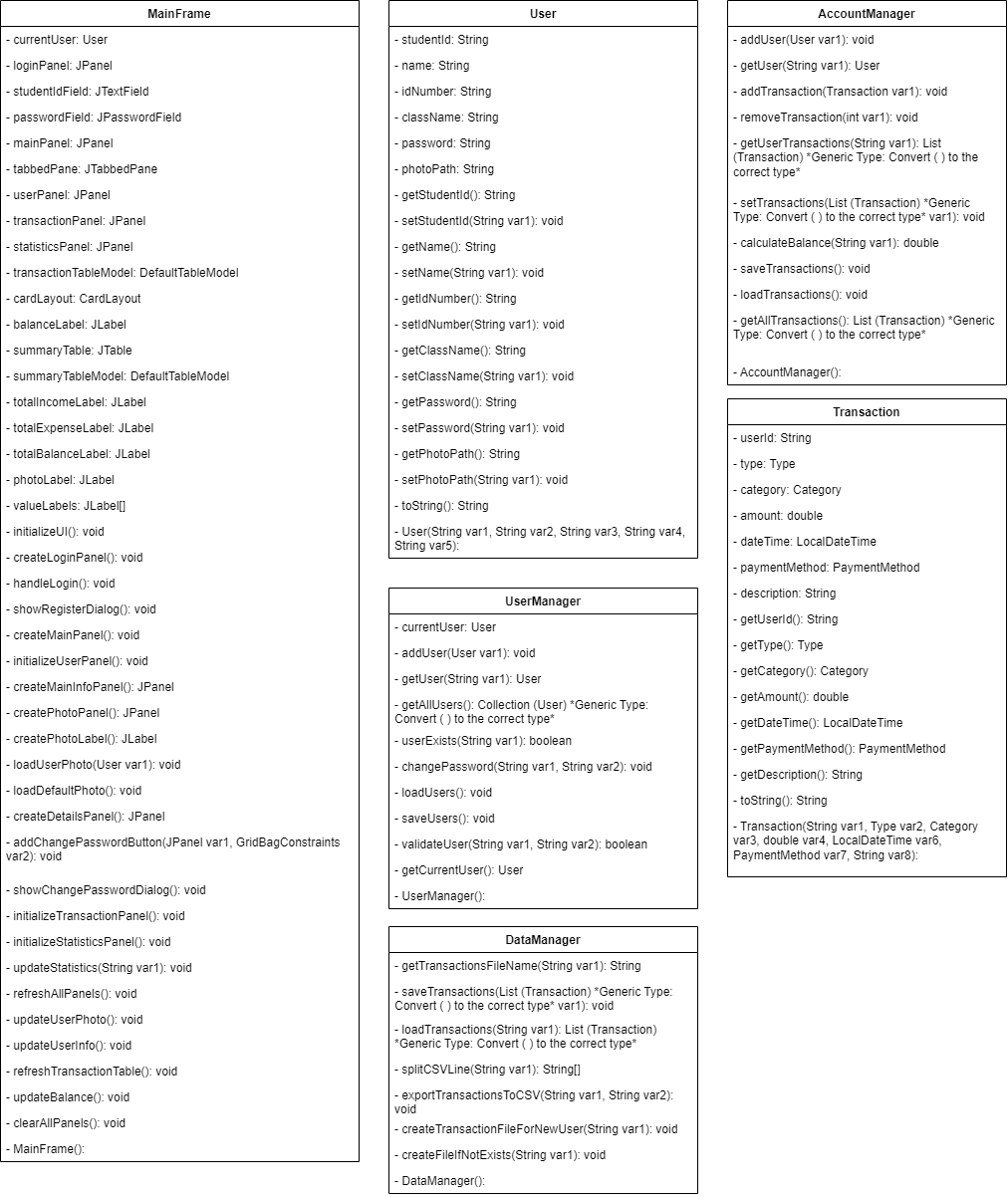
1. 数据图表界面



**图 13统计报表展示**

通过对数据数据的筛选，实现了对指定时间段的数据进行展示。然后再次展示这个Table。并对表格下方的Lable进行更新。

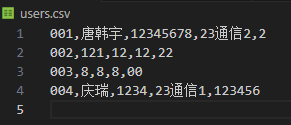
# 三、Java类图结构的设计：



**图 14 UML图结构**

# 四、测试数据与结果分析

1、程序测试过程（多用户）



**图 15用户信息存储**

通过登录不同账号能够看见不同的用户照片以及历史余额以及存储记录。



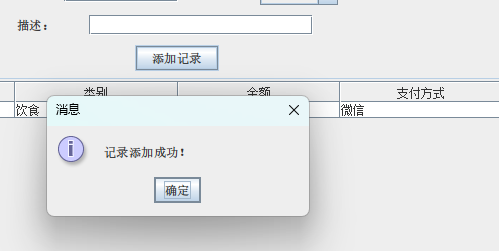
**图 16用户1数据**

图形用户界面, 应用程序

描述已自动生成

**图 17用户2数据**

2、添加数据测试



**图 18添加数据后图表中数据更新**

可以直接根据下拉菜单实现数据的填写。



**图 19输入非法金额后的提示**

当输入数据为负数时，系统会提示报错，但本系统对余额正负并不进行操作设置。

删除数据操作在前文提到过。

3、开发中遇到的主要问题

1）**照片的导入**是在开发中遇到的比较大的问题，由于用户的注册时不一定按照学号的名称进行注册，导致photoPanel中无法添加正确的图片，从而导致报错，本系统采用了默认图片（采用相对路径寻找图片）的方式来实现无法找到正确图片导致的报错。



**图 20处理无图片用户操作**

1. 对于添加各种Lable后更新无法及时更新，以及在布局时会乱跑无法达到理想的效果。本项目采用了GridBagLayout一种灵活的布局管理器，用于创建复杂的用户界面。允许组件在一个可调整大小的网格中放置，并支持跨多个单元格的布局，并在设计前先构思了布局，然后进行开发。
2. 对于存入文件的信息格式不一样，读取不方便。本项目在开发中采用格式化时间格式来完成这一问题。



**图 21格式化时间操作**

这样存到文件中的数据就格式一样然后方便操作了。

# 五、课程设计总结

1、设计特点

本课程设计主要围绕个人记账管理系统展开，旨在帮助用户进行日常收支记录和统计分析。其主要特点包括：

用户友好的界面：采用简洁直观的界面设计，使用户能够轻松进行操作。

多功能性：系统不仅支持基本的收入和支出记录，还提供统计分析功能展现选定时间内的各月金额总数，帮助用户更好地理解自己的财务状况。

数据持久化：通过文件存储机制，确保用户的交易记录能够持久保存，便于随时查看和管理。同时每个用户的数据库独立，不相互干扰，

2、创新之处

本设计的创新之处在于：

自定义时间范围统计：用户可以自由选择统计时间范围，提供灵活的数据分析功能，满足不同用户的需求。

丰富的交互提示界面：各种非法输入（密码不合适、添加数据不合适）都有相关的提示。

3、不足之处

1）软件在一开始在用户未登录时，并没有做权限管理，在打开软件时，程序会去读取上次的参数，但是此时用户没有登录，因此是找不到这个文件的，就会在终端中报错，但是并不影响软件的运行。



**图 22解析错误**

2）并没有提供用户上传用户图片的渠道，只能通过后台添加图片，需要在未来继续开发。

4、困难及解决方式

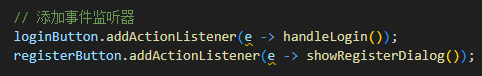
1）对于输入IO六以及一些函数不会使用，通过查看CSDN，或询问AI得出相关函数的使用方法，或者简单的例子，来更好的理解函数。

2)在开发过程中遇到了进调试发现每个输出都是正常的，但是图里就是不更新，后经过加注释，发现创建了两个对象，展现的第一个对象，但是一直在修改第二个对象，导致无法达到理想的效果。

5、心得、展望、体会

1）GUI开发与传统的终端输出并不一样，需要考虑元素的布局，同时如何将用户从GUI传入的参数给到对应的函数，将函数的输出传回，以及模块的刷新，都需要考虑的更多。

2)针对按钮等模块需要添加监听事件才能实现读取鼠标的电机，或者键盘的输入等操作。是单独的一个模块。



**图 22添加监听**

3）Java是面向对象的一门程序，要减少代码的复用，实现提取公用代码，总结为一个具体的函数才能更好的减少代码量，本次实验中1000多行的主函数代码读起来确实十分吃力，如果在写的时候不添加注释真的读不懂。

4）UML图的实现，找了很多方案，相较于Eclipse自动生成，Vscdoe就没有这种功能，通过查询Github探寻开源项目，最后选择**Java to Draw.io UML+draw.io**来实现框图的绘制。对面向对象开发有了新的理解。