班内序号：

**计算机工程学院/大数据学院**

**《项目版本管理实践》**

实践报告

2024-2025学年第一学期

|  |  |
| --- | --- |
| 题目 | 个人账单管理系统设计及其版本管理 |
| 专 业 |  |
| 班级 |  |
| 学生姓名 |  |
| 学生学号 |  |
| 指 导 教 师 |  |
| 日期 | 2024.12.10 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 一级指标 | 二级指标 |  | | | 分值 | | 得分 |
| 平  时  考  核  （50%） | 视频学习进度 | 及时完成学习通视频学习任务 | | | 20 | |  |
| 考核时间节点1 | 1. 完成项目版本控制操作环境的配置；创建本地仓库，编写程序。 | | | 10 | |  |
| 2. 通过Git 完成基本操作，包括提交代码、撤销更改、查看历史和状态。 | | | 10 | |
| 考核时间节点2 | 1.创建、切换、查看、修改、合并、删除分支。 | | | 10 | |  |
| 2. 处理工作区中的错误，包括暂存更改、修复 bug、合并更改到主分支，并查看 reflog 记录。 | | | 10 | |
| 考核时间节点3 | 1. 注册gitee账号，关联本地仓库和远程仓库并进行相关配置；完成项目内容并进行推送 | | | 10 | |  |
| 2. 协作开发练习，包括 fork 仓库、克隆仓库、创建分支、推送更改、发起Pull Request并合并更改到主仓库 | | | 10 | |
| 考核时间节点4（答辩环节） | 能够对过程和代码实现进行完整阐述，能流畅地回答答辩所提问题 | | | 20 | |  |
| 合计（百分制） | | | | | |  |
| 期  末  考  核  （50%） | 作品完成度 | 工作量达标，系统功能点实现完整，设计合理，逻辑条理性好，独立完成。 | | | 30 | |  |
| 版本控制实现效果 | 能够正确运用各种命令完成仓库创建、本地远程仓库同步、与他人的协助顺利实现。 | | | 30 | |  |
| 报告完成度 | 报告文字描述规范，语言通顺，结构完整，思路描述清晰，结果分析准确。 | | | 40 | |  |
| 合计（百分制） | | | | | |  |
| 评  语 |  | | | | | | |
| 总  评 |  | 分 | 签名 |  | | 202 年 月  日 | |
| □ 优秀 □ 良好 □ 中等 □ 及格 □ 不及格 | |

**一、作业目的**

学习并掌握当前主流的基于分布式的版本管理工具Git一族的使用，能够使用基于Git的免费代码托管平台分享项目、开展项目协作及向他人的项目贡献提交。

**二、作业要求与内容**

实验内容：采用Git工具箱，以Gitee为远程仓库，管理自己的开发项目，提交自己的修改记录，开源自己的项目以供他人拉取，以及给他人的项目贡献自己的提交。分为6个周次的内容，每个周次的内容下面具体描述，每个步骤截图说明作为实验报告内容。

**（一）知识点学习**

请登录超星平台，将课程视频完整学习。

**（二）项目题（编程实现）**

设计一个简易的个人账单管理系统，帮助用户轻松管理自己的财务收支。该系统需要具备记录、查询、统计等功能，并能够有效地帮助用户管理他们的预算，具体功能如下：

1.记录收入：用户可以录入新的收入情况，每条记录包括：日期、金额（收入金额必须为正数）、类别（如工资、奖金等）和备注。

2.记录支出：用户可以录入新的支出情况，每条记录包括：日期、金额（支出金额必须为正数）、类别（如餐饮、交通、购物等）和备注。

3. **（第三周任务添加）**账单展示：系统能够展示所有录入的账单记录，收入和支出分开显示。

4. **（第三周任务添加）**账单查询：用户可以按照指定日期、日期范围或类别查询账单，收入和支出分开显示。

5.**（第五周任务添加）**预算管理：用户可以设置每个月支出的预算限额，系统显示当前剩余可用预算。

6.**（第五周任务添加）**月度统计：系统可以统计每月的总收入、总支出和各类别的金额。

7.菜单界面：系统提供一个简单的文本菜单界面，使用户能够通过选择菜单选项执行不同的操作。用户可以通过输入菜单选项的序号来执行相应的操作。

8.错误处理：系统能够检测并处理用户输入的错误，如非法的日期格式、负数金额等，并给出相应的错误提示。

9.退出功能：用户可以通过菜单选项安全退出程序。  
【示例】

=================================

欢迎使用个人账单管理系统

=================================

请选择操作：

1. 记录收入

2. 记录支出

3. 查看所有账单**（第三周任务添加）**

4. 查询账单**（第三周任务添加）**

5. 设置月度预算**（第五周任务添加）**

6. 查看月度统计报告**（第五周任务添加）**

7. 退出系统

请输入选项序号：***1***

=================================

欢迎使用个人账单管理系统

=================================

请输入收入信息：

日期（YYYY-MM-DD）：***2024-08-18***

金额：***3000***

类别（如工资、奖金等）：***工资***

备注：***优秀工作者***

收入已成功记录！

按任意键返回主菜单..

**（三）版本控制与协作6周次任务**

**第一周任务（知识点1 2 3）：安装和初始化**  
1.安装Git，并验证安装是否成功（git --version）。  
2.配置全局用户名和邮箱地址。  
3.创建一个文件夹，初始化一个本地仓库。

4.完成项目题的部分功能**（记录收入、记录支出）**，将代码文件保存在第三步创建的文件夹中。

**第二周任务（知识点4 5）：基本操作**  
1.将代码文件添加（add）到本地仓库，并进行一次有意义的提交(commit)，说明信息为“添加功能：记录收入、记录支出”。

2.在代码文件中添加注释后添加到本地仓库；撤销本次添加（reset），并回滚（checkout）工作区的修改，使代码恢复到修改前。

3.再次在代码文件中添加功能的注释，添加并提交，说明信息为“注释1”，最后将本次提交进行回退（revert）。

4.查看提交历史（log），并查看当前仓库的状态（status）。

**第三周任务（知识点4 8 9）：分支管理**

1.在本地仓库中，创建一个新的分支，命名为feature，查看所有分支。  
2.切换到feature分支，在代码文件中添加新的功能**（查看所有账单、查询账单）。**  
3.提交修改到feature分支，提交信息为“feature添加功能：查看所有账单、查询账单”。

4.切换回master分支，将feature分支合并（merge）到master分支，合并成功后删除feature分支，查看当前仓库的状态。

**第四周任务（知识点4 5 6）：暂存工作区**   
1. 在代码中引入一个错误，确保当前的工作区无法正常运行。使用stash命令暂存未提交的修改。查看暂存列表，确保暂存成功。

2. 创建并切换到 bugfix 分支，使用 git stash apply将暂存的更改应用到工作区。

3. 在 bugfix 分支上修复bug，提交更改，并切换回原分支，使用 git merge 将修复合并。

4.查看操作历史（reflog）。

**第五周任务（知识点3 4 7 10 11）：远程仓库基本操作**  
1.在Gitee上注册一个账号，并创建一个空的远程仓库，将本地仓库与远程仓库关联（remote）。

2.配置 SSH 秘钥以便安全地与远程仓库通信。

3.继续完成项目的功能**（预算管理、月度统计）**，添加到本地仓库并提交，提交信息为“添加功能：预算管理、月度统计”。

4. 推送本地提交到远程仓库。

**第六周任务（知识点3 4 7 8 9 10 11）：远程仓库协同开发**

1.选择一位合作同学，fork该同学的仓库（全部分支）到自己账号下。

2.克隆（clone）自己账号下刚fork的仓库到本地开发环境。

3.在本地创建新的分支，进行代码更改后，将更改推送至自己的远程fork仓库对应分支。

4. 从自己的fork仓库中的新分支发起Pull Request，请求将更改合并到原仓库的master分支并通过审核。

**三、完成情况**

在本次作业中，我成功完成了个人账单管理系统的设计与实现，主要包括以下几个方面：

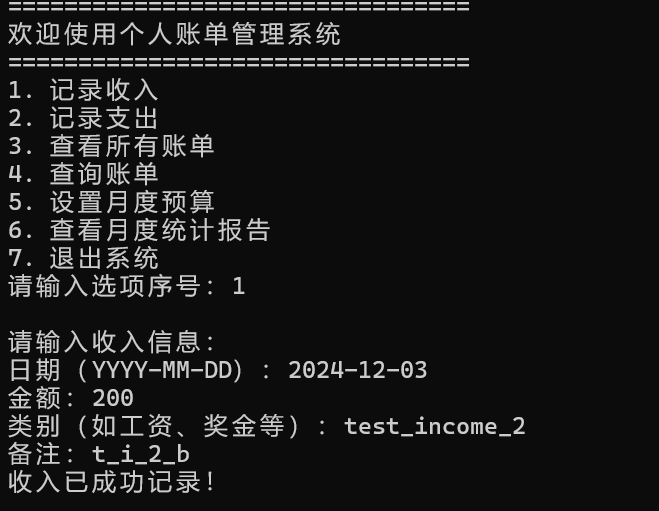
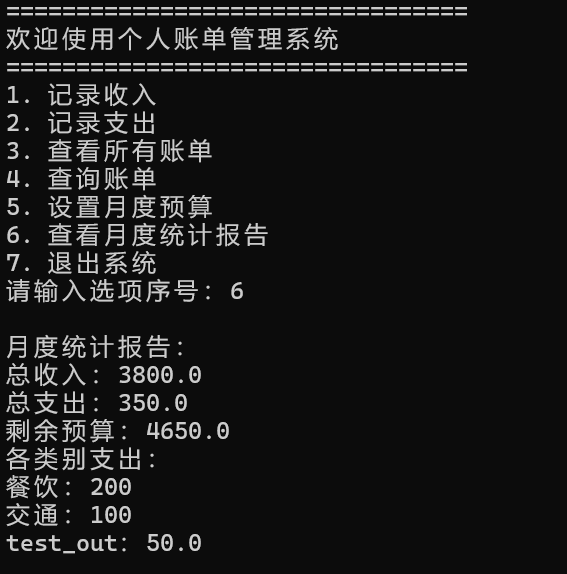
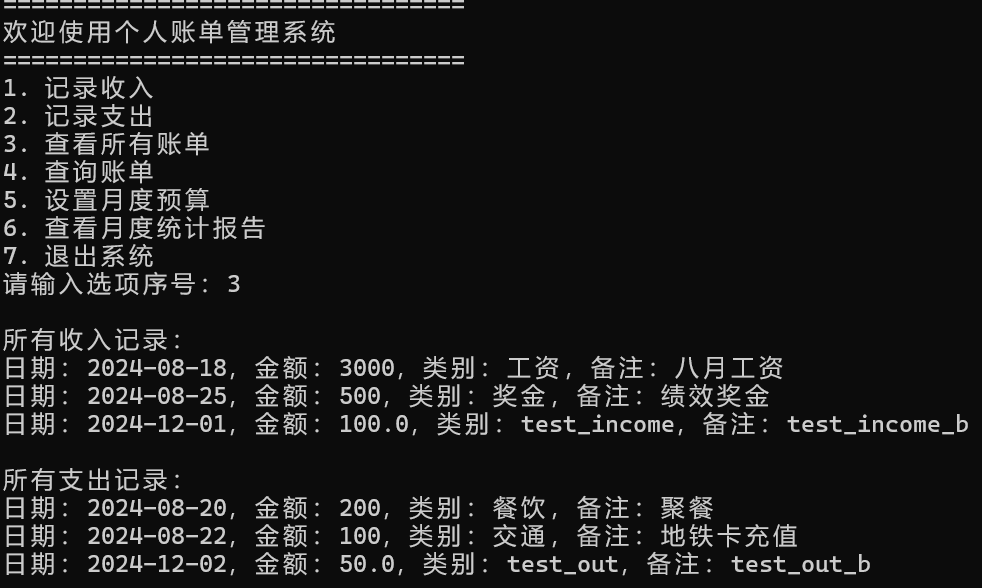
收入与支出记录功能： 实现了用户录入收入与支出信息的功能，每条记录包括日期、金额、类别以及备注。通过用户输入的内容，系统能够将这些数据保存到一个 JSON 文件中，以便后续查询和统计使用。

账单查询与展示功能： 在系统的基础框架上，实现了账单展示和查询功能，能够根据用户的需求展示所有账单，并允许按日期、类别或日期范围查询相应的账单记录。

预算管理与月度统计： 在系统的第五周任务中，添加了预算管理功能，用户可以设定每月的支出预算，系统会根据收入和支出的情况自动计算并显示剩余可用预算。此外，我还实现了月度统计功能，可以生成每月的收入总额、支出总额以及各类别的支出金额。

错误处理与用户界面： 在系统中加入了错误处理机制，确保用户在输入数据时，如果有不合法的日期或负数金额，程序会给出友好的提示，避免崩溃。在用户界面上，设计了一个简单的文本菜单，用户可以通过选择菜单选项执行不同的操作，系统通过文本界面向用户展示信息和提示。

文件存储与数据管理： 数据存储采用了 JSON 格式，以保证数据的持久化。在程序运行过程中，确保了读取和写入操作的正确性，同时处理了文件路径问题，确保打包后的可执行文件也能正常读取和保存数据。

打包与可执行文件生成： 在完成代码开发后，使用 PyInstaller 将 Python 程序打包为可执行文件，并解决了文件路径的兼容问题。通过 --add-data 参数，我成功将 JSON 文件包含在打包后的 .exe 文件中，并通过 resource\_path 函数确保在打包后程序能够正确加载资源文件。

**四、总结（感想或心得等）**

在完成个人账单管理系统的过程中，我深入理解了 Python 编程、文件处理、数据存储以及打包的各个方面。最初，面对如此多功能的实现，我感到有些迷茫，但随着任务的推进，我逐步掌握了如何将多个功能模块整合到一个系统中。尤其是在实现数据存储和读取部分时，我通过 JSON 文件来持久化数据，使得用户每次使用时都能够查看和操作之前的账单信息。这一过程不仅让我掌握了文件操作的基本技巧，还增强了我对数据管理的理解。

在调试和打包过程中，我遇到了文件路径问题。尤其是在使用 PyInstaller 打包成可执行文件后，我发现资源文件没有被正确加载。经过调试，我学习了如何使用 sys.\_MEIPASS 获取正确的文件路径，并且通过修改代码确保了无论是在开发环境还是打包后的环境中，都能够正确加载资源文件。这让我更加熟悉了 PyInstaller 的打包机制，并提高了我处理问题的能力。

在整个开发过程中，我深刻体会到编程不仅仅是写出代码，更重要的是如何设计一个合理的架构，如何处理用户输入的各种异常情况，如何让程序具备更好的鲁棒性和用户体验。尤其是错误处理和界面的设计，使得程序更加友好，用户体验也得到了显著提升。

总之，完成这个个人账单管理系统的任务，不仅提升了我的编程能力，也让我更加理解了软件开发的流程。从需求分析、功能实现到最终的打包发布，每一步都需要细心和耐心，这让我受益匪浅。在未来的编程学习中，我会继续积累经验，不断优化自己的代码结构和设计思想。

**五、参考文献**