1. 概述

1.1 功能实现情况

本个人收支管理系统功能有二个：输入收支数据并保存、根据用户输入显示当前月份的综合收支情况，同时给用户提供查看当前月收支明细的功能。

目前，已经实现了上述两个功能，但是在实现的过程中也是无可避免的出现一些问题，比如在输入数字进行功能选择时，使用if语句进行字符比对的过程中忽略了字符串与整型数据的区别，导致程序出现异常，等等，更多的程序异常将在后续具体说明。

1. 程序概要设计

2.1 函数说明

2.1.1 主函数

该函数为程序运行的主函数，以while循环开始，按照用户的输入调用对应的函数进行操作，最后回到主函数，直到用户输出退出指令，程序结束。

2.1.2 菜单初始化函数

主函数运行时首先运行此函数，根据初始定义的字典生成菜单，并使用print()函数输出，供用户选择。

2.1.3 数据插入函数

该函数接收一个用户输入的字符串，如a1,2020-3,2000.0,生活费，然后使用split()函数根据,分为四部分，分别赋值给\_type(类别),\_time(时间),\_amount(金额),\_remark(备注)，随后使用open函数打开一个名为data.csv，将类别，时间，金额，备注作为一行以追加的形式写入文件。

2.1.4 数据汇总函数

该函数根据用户输入的时间，使用open()函数打开data.csv文件并使用readlines()读取所有行，然后遍历所有行，使用in方法来判断用户输入的时间是否在当前行中，若在，对应类别的金额增加。在显示月综合汇总之后，会提示是否显示详细数据，若用户输入y，则重新遍历所有行，并将符合条件的数据以升序排列并显示。

2.2.1 函数调用关系

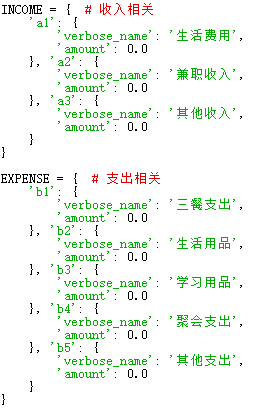
1. 程序详细设计

3.1 代码分析

主要数据变量

|  |  |  |
| --- | --- | --- |
| 变量名 | 变量类型 | 说明 |
| INCOME | dict | 收入类别字典 |
| EXPENSE | dict | 支出类别字典 |
| \_type | string | 收支类别 |
| \_time | string | 时间 |
| \_amount | string | 金额 |
| \_remark | string | 备注 |
| data\_list | list | 存放文件所有行数据 |
| total\_income | int | 总收入 |
| total\_expense | int | 总支出 |
| new\_data\_list | list | 存放所有切割后的行数据 |
| Input\_data\_split | list | 用户输入数据切割列表 |

程序开始定义了两个字典，一个INCOME用于存放收入类别相关的数据，一个EXPENSE用于存放支出类别的数据。结构如下：

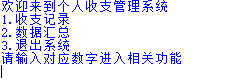


程序开始执行时，首先进入底部的main函数，输出欢迎信息，随后输出菜单，方便用户使用各功能

代码如下：

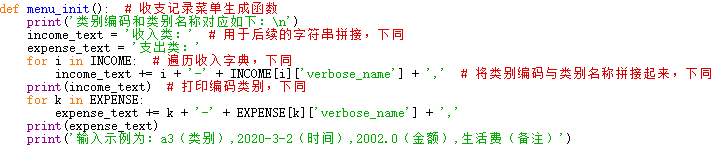


运行截图

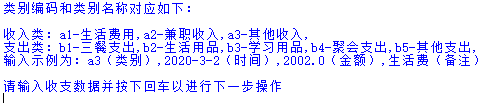


当用户输入1并回车时调用收支记录菜单生成函数，进入收支记录，在其中可以记录用户的收支情况。

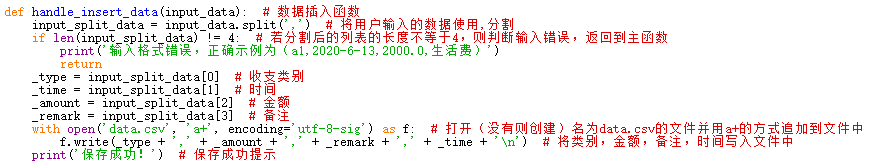
代码如下：



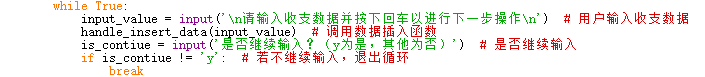
运行截图



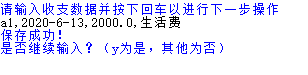
给用户输出友好的提示信息以及输入示例，在用户输入正确的数据后提示用户保存成功。代码如下：



保存成功后会询问是否继续输入，代码如下：



运行截图：



存入数据文件截图：



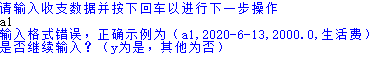
若用户输入为非y，则退出收支记录，重新运行主函数。

在用户输入这一步中，添加了数据验证。若输入的数据根据,分割之后的长度不为4，则判断用户的输入错误，同时打印错误信息，并让用户重新输入。

判断用户输入是否正确的代码如下：



运行截图

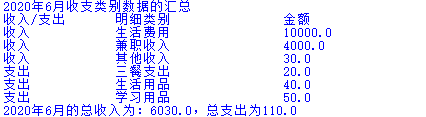


当用户输入2时，调用summary函数，进入数据汇总功能。

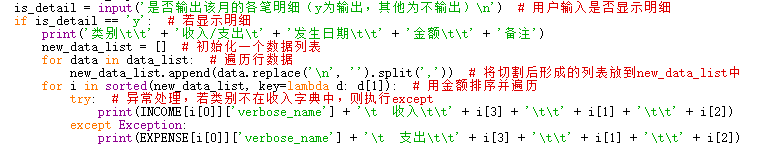
系统会提示用户输入需要查看的年份以及月份，并在用户回车后读取本地存储的data.csv文件，进行数据处理，然后打印展示当月各类别的金额汇总。代码如下：



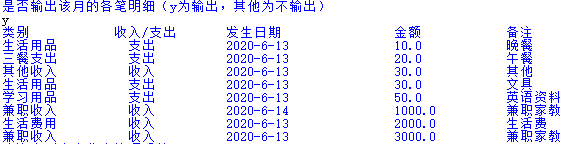
运行截图：



输出本月收支汇总后，系统会提示是否输出本月各笔明细，若用户输入y，则系统会显示本月的各笔明细，代码如下：



运行截图：



在显示完本月明细后，会返回主函数，重新打印主菜单供用户选择

此时输入3回车退出系统。

1. 使用情况
2. 总结

5.1 程序的优点

1. 为了在显示类别简化操作，将类别编码与类别名称写入字典中形成映射关系，在后续数据的展示中可以很方便的将类别编码转为类别名称。

2. 在程序设计的过程中，根据实际情况对用户的输入进行了校验，比如在用户输入收支数据时，使用split()函数对数据进行切割，若切割后的列表的长度不为4则判定用户输入错误，提示用户重新输入。

3.在主函数以及数据录入时采取while循环，这样在用户输入数据时可以一直循环输入，简化操作。当用户操作完成时会自动返回到主菜单，这样方便用户操作。

5.2遇到的问题

5.2.1数据类型不一致

在一开始设计主菜单时，输入数字进入各自代表的功能，但是在实际开发中忽略了字符串型与整形数据在比较时的不一致，一开始写成了



但是运行后出现了错误，随后意识到了这个错误，更改了过来。

5.2.2 数据编码问题

在一开始存储数据时，使用的utf-8编码，但是发现在读取中数据显示正常，但是打开csv文件后发现里面的文字都是乱码，意识到可能在写入csv文件时编码有问题，后将编码改为utf-8-sig，csv文件打开后显示正常。

5.2.3 没有及时中止循环

在显示收支数据时，为了让类别编码转为，类别名称显示出来，采用了遍历程序开始时定义的收支字典与文件中数据比对的方法来实现，但是在遍历的过程中，忘了在比对成功后中止循环，造成了不必要的时间/空间浪费。在发现问题后，每次比对成功后使用break退出循环。

5.2.4 字符串拼接的问题

为了显示本月的总收入与总支出，需要将字符串与收入/支出数据拼接起来，在拼接的过程中，忘记字符串不能与整型/浮点型数据直接拼接，需要使用str()函数将其转为字符串类型才能进行拼接。、

5.2.5 写入文件遇到的问题

在写入文件时，习惯性写成了’w’，造成每次输入都覆盖之前数据的严重后果。在发现问题后，及时将’w’改为’a+’，将每次输入的数据都追加到上一次写入的数据后边。

5.2.6 不同数据类型之间的运算问题

在进行数据汇总时，需要将本地存储的数据读取出来，进行算术运算，但是在实际开发当中，进行运算时忘记将字符串型的金额转为可进行算术运算的整型/浮点型，导致程序运行出错。发现问题后将字符串类型统一转为float进行运算。

5.2.7 列表索引问题

在Python中，列表的索引应该从0开始，在实际开发中，误认为索引是1开始，导致数据索引错误，引起程序异常，现已更正。

5.3 收获

在课程学习中，我对Python的相关数据类型，比如dict（字典），tuple（元祖），list（列表）等等相关的概念都比较模糊，不知道该怎么去区分以及使用。所以在课程设计一开始，各种程序报错也是让我手足无措，但是在不断的敲代码的过程中，我遇到问题就去书本里翻看，在网络查找，不知不觉中，我对Python的认识逐渐清晰了起来，我从一开始的手足无措渐渐开始得心应手起来，对Python的函数调用关系、各种数据类型之间的操作以及常见的错误类型都有了进一步的认识。经过这次课程设计，我意识到自己Python这门语言依旧所知甚少，在网络上我也了解到，Python正在成为人人都会的一门语言，其在各行各业的用途也是越来越广，甚至幼儿园的小朋友门也在学。对于我所处的领域来说，它的用途也是极广的，Python可以用来做金融数据分析，这可以大大简化我今后的工作，这更坚定了我进一步学习Python的决心。在今后，我会继续学习Python，并将其用在实际生活当中，为自己增添一技之长。