

项目说明文档

开发环境:代码在Mac系统和Linux系统通用,如果是windows系统,需要将system("clear")注释掉或者改为system("cls").

main.cpp说明

```
void showMainMenu()  
{  
    system("clear");  
    cout << "请选择功能:" << endl;  
    cout << "1.添加费用项目" << endl;  
    cout << "2.按月汇总" << endl;  
    cout << "3.查询" << endl;  
    cout << "q.退出" << endl;  
}  
  
int main(int argc, const char * argv[]) {  
    while (1) {  
        // 获取选择的项目  
        showMainMenu();  
        cout << "请输入选项编码:";  
        char choice = '0';  
        cin >> choice;  
        // 根据项目显示不同的功能界面  
        switch (choice) {  
            case '1':  
                {  
                    InputStatement i;  
                    i.showInputMenu();  
                    i.inputAccount();  
                }  
                break;  
            case '2':  
                {  
                    SummarizeInfo s;  
                    s.getMonth();  
                    s.getMonthInfo(1);  
                }  
                break;  
            case '3':  
                {  
                    SearchInfo si;  
                    si.showSearchMenu();  
                }  
            }  
        }  
    }  
}
```

```
break;
case 'q':
    exit(0);
break;
default:
    break;
}
}
return 0;
}
```

- showMainMenu(): 显示菜单栏
- Main:通过用户选择的输入切换到相应的功能界面,在功能界面完成具体功能.
 - 通过switch...case..进行选择

InputStatement.cpp说明

该文件负责需求文档第一个功能,提示用户输入明细,并存入到本地。

```

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
*   funcName: InputStatement::show_input_menu()
*   parameters: void
*   return: void
*   comment: 打印类别显示菜单
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
*/

void InputStatement::showInputMenu()
{
    system("clear");
    cout << "类别编码和类别名称的对应关系如下：" << endl;
    cout << "收入类：a1-生活费，a2-奖学金，a3-稿费" << endl;
    cout << "支出类：b1-学习用品，b2-生活用品，b3-游戏" << endl;
}

/* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
*   funcName: InputStatement::inputAccount()
*   parameters: void
*   return: void
*   comment: 输入账目
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * */
*/

void InputStatement::inputAccount()
{
    cout << "请输入明细，格式如下：\na1 2020-1-26 2000.0 1月生活费" << endl;
    cout << "请输入添加明细的数量:";
    int accountCount;
    cin >> accountCount;

```

```

for(int i = 0; i < accountCount; i++){
    cout << "请输入收支明细:";
    string classID;
    string date;
    float money;
    string remark;
    // 从控制台获取输入的内容
    cin >> classID >> date >> money >> remark;
    cout << classID << date << money << remark;
    // 将内容写到本地
    ofstream ofs;
    ofs.open("./account.txt", ios::app);
    if (ofs.is_open()) {
        // 文件打开正常就将内容写到文件中
        ofs << classID << " " << date << " " << money << " " << remark <<
endl;

        ofs.close();
    }
    else{
        cout << "文件打开失败" << endl;
    }
}
}

```

- inputAccount()说明:
 - 首先提供输入样式.
 - 获取一次性需要输入的明细的数量,便于一次输入多条账目明细.
 - 通过循环依次获取输入的内容,并以追加写的方式写进文件中完成保存.

SummarizeInfo.cpp说明

```

void SummarizeInfo::getMonthInfo(int flag)
{
    // 定义接受文件内容的变量
    string classID;
    string date;
    float money;
    string remark;

    // 读取文件中存储的内容
    ifstream ifs;
    ifs.open("./account.txt", ios::in);
    if(!ifs.is_open()){
        cout << "文件打开失败" << endl;
    }
    else{

```

```

while (ifs >> classID && ifs >> date && ifs >> money && ifs >> remark)
{
    AccountInfoPer infoPer;
    infoPer.classID = classID;
    infoPer.date = date;
    infoPer.money = money;
    infoPer.remark = remark;
    infoVec.push_back(infoPer);
    int ret = date.find(this->dateStr);
    // date中包含datestr则表明是这个月份数据
    if (!ret) {
        //      cout << classID << date << money << remark << endl;
        this->computeTotalMoney(classID, money);
    }
}
ifs.close();
this->showInfo(1);
this->showDetailinfo();
}
}

```

- 主要功能:
 - 通过只读的方式打开文件,并通过while循环依次读取每一行的内容,并赋值给相应的临时变量.
 - 定义了一个结构体AccountInfoPer存放每一行的内容,便于后面的排序.
 - 通过容器依次把AccountInfoPe类型数据存放,使用容器主要为了方便后续排序.

```
bool myCompare(AccountInfoPer &v1, AccountInfoPer &v2)
{
    return v1.money < v2.money;
}
```

- 功能说明: 自定义比较函数,用于自定义数据类型容器的排序,因为是按照金额排序,所以比较money.

SearchInfo.cpp说明

该类主要完成查询,定义了两种查询方式:

- 通过金额查询大于该金额的条目的明细
- 通过类型查询所有该类型的条目
- 来个方式都可以统计出总金额

```
/* * * * * * * * * * * * * * * * * * * * * */  
* funcName: SearchInfo::searchMoneyMoreThan()  
* parameters: void  
* return: bool  
* comment: 查询大于某个金额的明细  
* * * * * * * * * * * * * * * * * * * * * */  
  
void SearchInfo::searchMoneyMoreThan()
```

```

{
    cout << "请输入金额:";
    float moneyFlag;
    cin >> moneyFlag;

    // 定义存放文件中内容的变量
    string classID;
    string date;
    float money;
    string remark;
    // 遍历文件,并根据条件进行筛选
    ifstream ifs;
    ifs.open("./account.txt", ios::in);
    if (!ifs.is_open()) {
        cout << "文件打开失败~" << endl;
    }
    else{
        while (ifs >> classID && ifs >> date && ifs >> money && ifs >> remark)
        {
            if (money > moneyFlag) {
                cout << classID << " " << date << " " << money << " " <<
remark << endl;
                this->totalMoney += money;
            }
        }
        cout << "金额大于" << moneyFlag << "元的总消费为:" << this->totalMoney <<
"元" << endl;
    }
    ifs.close();

    // 等待键盘输入
    cout << "请输入任意键继续...";
    char waitFlag;
    cin >> waitFlag;
    if (waitFlag != ' ')
    {
        return;
    }
}

```

- 功能说明:循环便利文件中的每一条记录,通过每条记录的金额和用户输入的金额进行比较,满足条件的就输出,并将满足条件的金额进行累加完成统计功能.

同理通过类型名查询就是遍历的时候吧与输入类型相同的数据进行输出和累加进行统计.