**站立会议记录**

会议时间： 2025年08月25日\_

会议地点：\_信息学院南楼B227\_

项目组名称：\_第 7组\_

参会人员：\_\_\_余沐阳，周以撒，林子涵，刘浩洋\_\_

1. **已完成任务列表：**

系统设计由小组成员共同讨论完成

1.系统设计思路与需求分析完成：

明确项目目标为开发一个“个人财务管理系统”，支持收入与支出记录的增删改查、分类统计、按时间段查询、数据持久化存储等功能。

采用面向对象编程（OOP） 思想进行模块化设计，核心设计原则包括：封装、继承、多态和抽象。

2.类结构设计与职责划分完成：

确定系统主要由以下几个类构成：

date 类：记录日期

record 基类：抽象财务记录的公共属性

income\_record 类：继承自 record，表示收入记录，新增 income\_category 枚举类型

spend\_record 类：继承自 record，表示支出记录，新增 spend\_category 枚举类型

record\_manager 类：作为系统核心管理器，负责所有记录的统一理，包括添加、显示、搜索、修改、删除、排序、统计、文件读写等业务逻辑。

**二、存在的问题：**

问题1（会上已解决）：date 类中如何确保日期的合法性（如 2 月 30 日、非闰年 2 月 29 日等）？是否应在构造函数或 set 方法中进行校验？

解决：计划在 date 类中增加 bool isvalid() const 成员函数，用于判断年、月、日组合是否合法。

问题2（会上已解决）：

record 基类中的 dt（日期）成员为 date 类型，而派生类 income\_record 和 spend\_record 需要频繁访问日期进行排序或查询，是否应将比较操作（如按日期排序）封装在 date 类中？

解决：

计划在 date 类中重载比较运算符（<, >, == 等），支持日期的直接比较。后续在 record\_manager 中实现排序功能时，可直接通过 record->dt < other->dt 进行逻辑判断，提升代码可读性与复用性。同时，date 类的比较逻辑已通过友元函数实现，确保外部类也能便捷使用。

问题3（后续解决）：

当用户修改或删除某一天的记录时，系统如何处理同一天存在多条记录的情况？是否需要提供更细筛选条件？

**三、今日分派任务列表：**

1、周以撒：完成date类与record基类的数据与操作方法的定义

2、刘浩洋：完成record.manager的头文件。

3、林子涵：完成收入记录类income\_record的头文件，继承自record, 添加收入类型枚举

4、余沐阳：完成支出记录类spend\_record的头文件，继承自record, 添加支出类型枚举