**站立会议记录**

会议时间： 2025年09月09日\_

会议地点：\_信息学院南楼B123\_

项目组名称：\_第7组\_

参会人员：\_刘浩洋，周以撒，林子涵，余沐阳\_\_\_\_

**一、已完成任务列表：**

1、周以撒:重载了date类的所有比较运算符

2、刘浩洋:完成了record\_manager中searchbydate(),modifybydate(),deletebydate()

3、林子涵:实现了itypeToString()与menu基本框架

4、余沐阳:实现了otypeToString()与main函数

**二、存在的问题：**

问题1 (会上已解决)：当前使用 float 类型存储金额，是否存在精度丢失风险？是否应改用 double 或整数类型（如以“分”为单位）？

解决：改为用double存储money

问题2（会上已解决）：当前系统使用 vector<record\*> 存储记录，在程序退出时若未正确释放内存，存在内存泄漏风险。

解决:在析构函数中释放内存

问题3（后续解决）：在 modifybydate()和deletebydate()中，当同一天有多条记录时，用户选择序号后缺乏确认提示，易导致误修改。

**三、今日分派任务列表：**

1、周以撒: 完成date和record剩余部分

2、刘浩洋:完成record\_manager剩余部分

3、林子涵:完成income\_record和menu剩余部分

4、余沐阳:完成spend\_record与main剩余部分

**白盒测试（分支覆盖）**

为验证 record\_manager 类中关键业务逻辑的正确性与完整性，我们对以下三个核心函数进行了白盒测试，采用分支覆盖策略，确保所有 if-else、循环等控制结构的真假分支均被测试用例执行。

测试目标函数:

record\_manager::searchbydate(int y, int m, int d) record\_manager::modifybydate(int y, int m, int d) record\_manager::deletebydate(int y, int m, int d)

**白盒测试用例设计与覆盖分析**

函数一：searchbydate(int y, int m, int d)

函数逻辑简述：

1. 遍历所有记录，使用三重条件判断年、月、日是否匹配。
2. 若匹配，则输出记录并递增计数器 count。
3. 若遍历结束后 count 仍为 1，说明无匹配记录，输出提示信息。

控制流分支：

if (records[i]->getdate().getyear() == y && records[i]->getdate().getmonth() == m && records[i]->getdate().getday() == d) → 真（匹配）、假（不匹配） if (count == 1) → 真（无记录）、假（有记录）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 输入 | 记录集状态 | 预期输出 | 覆盖分支 |
| 1 | 2025/09/01 | 包含该日期的记录（1条） | 输出表头和该记录count>1，不执行 | if(count==1) |
| 2 | 2025/01/01 | 无匹配记录 | 输出表头，然后提示“No\_records\_found” | if(count==1)为真，执行提示 |
| 3 | 2025/09/09 | 包含该日期的记录（多条） | 输出表头和所有匹配记录 | 多次进入if匹配分支，验证循环处理 |

分支覆盖结果：100%。所有 if 分支和循环迭代路径均被覆盖。

函数二：modifybydate(int y, int m, int d)

函数逻辑简述：

1. 创建 recordsToModify 向量存储匹配记录。
2. 遍历 records，将日期匹配的记录加入 recordsToModify。
3. 若 recordsToModify 为空，输出提示并返回。
4. 否则显示记录列表，用户输入选择（0取消，1~n选择）。
5. 若选择0，取消修改。
6. 否则对选中记录逐项询问是否修改金额、日期、对方、类型。

控制流分支：

if (records[i]->getdate().getyear() == y && ... ) → 真（加入）、假（跳过） if (recordsToModify.empty()) → 真（无记录）、假（有记录） if (choice == 0) → 真（取消）、假（继续） 每个修改项（金额、日期、对方、类型）均有 if (modifyX == 1) 分支

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例 | 输入 | 匹配记录数 | 用户选择 | 修改项 | 预期行为 | 覆盖分支 |
| 1 | 2025/01/01 | 0 | - | - | 提示“No\_records\_found”，返回 | if(recordsToModify.empty())为真 |
| 2 | 2025/09/01 | 1 | 1 | 全部不修改 | 成功退出，无修改 | if(choice==0)为假，所有modifyX==0分支 |
| 3 | 2025/09/01 | 1 | 1 | 修改金额和日期 | 成功修改两项 | if(choice==0)为假，modifyMoney==1,modifyDate==1 |
| 4 | 2025/09/01 | 2 | 0 | - | 提示“Modification\_cancelled” | if(choice==0)为真 |
| 5 | 2025/09/01 | 2 | 2 | 修改对方和类型 | 成功修改对应记录 | 验证多记录选择与修改 |

分支覆盖结果：100%。所有主干分支和用户交互分支均被覆盖。

函数三：deletebydate(int y, int m, int d)

函数逻辑简述：

1. 排序并创建 recordsToDelete 和 indices 向量。
2. 遍历 records，将匹配记录及其原始索引存入两向量。
3. 若为空，提示并返回。
4. 否则显示记录，用户选择：1~n（单条）、-1（全部）、0（取消）。
5. 若选择0，取消。
6. 若选择-1，确认后从后往前删除所有匹配记录。
7. 若选择1~n，确认后删除对应单条记录。
8. 最后重新排序并保存。

控制流分支：

if (records[i]->getdate().getyear() == y && ... ) → 真（加入）、假（跳过） if (recordsToDelete.empty()) → 真（无记录）、假（有记录） if (choice == 0) → 真（取消）、假（继续） if (choice == -1) → 真（全删）、假（单删） 删除单条或全部时均有确认分支：if (confirm != 1) → 真（取消）、假（执行删除）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 测试用例 | 输入 | 匹配记录数 | 用户选择 | 确认 | 预期行为 | 覆盖分支 |
| 1 | 2025/01/01 | 0 | - | - | 提示“No\_records\_found”，返回 | if(recordsToDelete.empty())为真 |
| 2 | 2025/09/01 | 1 | 0 | - | 提示“Deletion\_cancelled” | if(choice==0)为真 |
| 3 | 2025/09/01 | 2 | -1 | 0 | 提示“Deletion\_cancelled” | if(confirm!=1)为真（全删取消） |
| 4 | 2025/09/01 | 2 | -1 | 1 | 成功删除所有2条记录 | if(choice==-1)为真，confirm==1 |
| 5 | 2025/09/01 | 2 | 1 | 0 | 提示“Deletion\_cancelled” | if(confirm!=1)为真（单删取消） |
| 6 | 2025/09/01 | 2 | 2 | 1 | 成功删除第2条记录 | if(choice!=-1)为假，confirm==1 |

分支覆盖结果：100%。所有删除路径和确认分支均被覆盖。

**总体测试结论**

通过设计针对性的白盒测试用例，我们对 record\_manager 类的三个核心函数实现了分支覆盖。测试验证了： 日期匹配逻辑正确（使用年、月、日三重判断）。 空/非空记录集的处理路径完整。 多记录场景下的用户选择与确认流程可执行。