

功能模块实现

登陆菜单

敖瑞成

陈腾

☐ 菜单窗口（敖瑞成）

- 函数原型：`ShowWindowMenu()`
- 打印登录时的选择界面：管理员或用户登录，或购物卡申请
- 返回选择的操作：管理员0，用户1，购物卡2
- 给退出

☐ 主操作页面（敖瑞成）

- 函数原型：`MainOperator(int n)`
- 对菜单窗口返回的值进行判断，并打印出相应的界面
- 若为0或1，管理员与用户的账号密码输入，并调用接口
- 若为2，调用购物卡申请接口

☐ 检查用户卡号密码（敖瑞成）

- 函数原型：`CheckCon(int number, char password[6])`
- 检查用户的卡号是否存在，若存在，检查密码是否正确
- 密码正确返回true，其他情况返回false

☐ 检查管理员账户密码（敖瑞成）

- 函数原型：`CheckAdm(int account, char password[6])`
- 检查管理员的账号是否存在，若存在，检查密码是否正确
- 密码正确返回true，其他情况返回false

☐ 传入登录用户（敖瑞成）

- 函数原型：`BuildCon(int number)`
- 从消费者链表中寻找卡号为number的用户，建立一个consumer
- 返回建立好的Consumer

☐ 传入登录管理员（敖瑞成）

- 函数原型：`BuildAdm(int account, Apply_Node* apply)`
- 从管理员链表中寻找账号为account的用户，建立一个administer
- 返回建立好的Administer

☐ 申请购物卡（陈腾）

- 函数原型：`AskCard(Apply_Node* apply)`
- 实现询问界面，输入并保存相应信息
- 用信息建立一个Apply节点并插入

☐ 管理员登录（敖瑞成）

- 函数原型 `EnterAdm(Administer& administer, Apply_Node* apply)`
- 实现管理员的所有操作
- 正常退出返回true，若选择注销返回false

☐ 用户登录（敖瑞成）

- 函数原型: `EnterCon(Consumer& consumer)`
- 实现用户的所有操作
- 正常退出返回true, 若选择注销返回false

☐ 更新管理员信息（敖瑞成）

- 函数原型: `UpdateAdm(bool judge, Administer& administer)`
- 若judge为true, 在管理员链表中找到该管理员并实现信息覆盖
- 若judge为false, 在管理员链表中找到该管理员并删除

☐ 更新用户信息（敖瑞成）

- 函数原型: `UpdateCon(bool judge, Consumer& consumer)`
- 若judge为true, 在用户链表中找到该用户并实现信息覆盖
- 若judge为false, 在用户链表中找到该用户并删除

管理员操作面(在EnterAdm中调用)

陈腾

☐ 菜单窗口（陈腾）

- 函数原型: `ShowwindowAdm()`
- 显示管理员的操作界面, 并提供选择, 不同选择调用不同接口
- 退出返回true, 注销返回false

☐ 制卡（陈腾）

- 函数原型: `MakeCard(Apply_Node* card);`
- 根据提供信息新建一个Consumer_Node并插入
- 初始余额为0, 激活状态为false

☐ 审批（陈腾）

- 函数原型: `CheckApply(Apply_Node* apply)`
- 查看Apply_Node列表, 若存在打印全部信息, 若不存在则打印不存在
- 提供选择, 让管理员输入同意批准的卡号, 并调用制卡接口

☐ 发卡（陈腾）

- 函数原型: `SendCard(Consumer_Node* card)`
- 将该卡的激活状态改为true

☐ 查看信息（陈腾）

- 函数原型: `ShowInfo()`
- 打印所有卡的信息

☐ 修改密码（陈腾）

- 函数原型: `ChangePsw()`
- 制作交互界面并修改该管理员的密码

用户操作面(在EnterCon中调用)

刘天一

李浩宇

☐ 菜单窗口 (刘天一)

- 函数原型: `ShowWindowCon()`
- 检测该用户是否被激活, 若没有被激活进行提示
- 显示用户的操作界面, 并提供选择, 不同选择调用不同接口
- 退出返回true, 注销返回false

☐ 报表 (刘天一)

- 函数原型: `ShowInfo(Consumer consumer)`
- 显示该用户的所有信息
- 余额和积分保留两位小数

☐ 修改密码 (刘天一)

- 函数原型: `ChangePsw(Consumer& consumer)`
- 修改该用户的密码

☐ 存款 (刘天一)

- 函数原型: `Save(Consumer& consumer)`
- 检查是否激活, 未激活则不能存款
- 在用户原来余额的基础上加上存入余额

☐ 余额交易 (李浩宇)

- 函数原型: `BuyBySave(Consumer& consumer)`
- 打印所有商品和价格 (两位小数), 并显示出交互界面 (买什么, 买多少)
- 计算价格, 并检查用户购物卡合法性 (是否激活, 余额是否充足)
- 进行交易, 调用扣除余额接口, 按照消费的30%进行增加积分

☐ 扣除余额 (李浩宇)

- 函数原型: `MinusSave(Consumer& consumer, float balance)`
- 根据传入所需扣除的金额进行扣除

☐ 积分交易 (李浩宇)

- 函数原型: `BuyByScore(Consumer& consumer)`
- 打印所有商品和所需积分 (两位小数), 并显示出交互界面 (买什么, 买多少)
- 计算积分, 并检查用户购物卡合法性 (是否激活, 积分是否充足)
- 进行交易, 调用扣除积分接口

☐ 扣除积分 (李浩宇)

- 函数原型: `MinusScore(Consumer& consumer, int score)`
- 根据传入所需扣除的积分进行扣除

CSV文件读写

☐ 读取管理员文件

☐ 写入管理员文件

☐ 读取购物卡申请文件

☐ 写入购物卡申请文件

☐ 读取用户文件

☐ 写入用户文件

