

网上购物仿真系统

本课程共安排 6 次上机，将实现一个简单的网上购物仿真系统（要求采用面向对象程序设计范型、C++语言实现）。

本文给出 6 次任务对应的系统需求、测试提示、对程序的最小要求。

上机时间、验收安排及其他要求请阅读《2025 年上机须知 and 第 1 次任务.docx》。

系统假设：

系统用户分为两大类：管理员和顾客。管理员的主要功能是维护系统中的后台数据；顾客可以通过系统完成商品的购买。

假定管理员只有 1 名，顾客有多名。

【界面实现提示】

(1) 无需实现为 GUI 形式，编写最一般的控制台程序(字符交互式)即可。

(2) 鉴于程序将有很多功能，你应将程序的主控部分，设计为“菜单驱动”方式。可按需设立多层多组菜单。（见老师讨论函数指针时提供的示例“学在西电/课程资料/课堂中的示例程序/菜单选择....cpp”）

【数据存储提示】

整体实验的初期，数据可仅存储在内存，但这是临时性的。

(至少到)实验后期，你应将应用相关数据（如顾客、商品、订单等等）保存到各自对应的文件中。文件内容的格式由你自行设计，应能方便程序读、写（包括追加、删除、更新数据），若也便于人阅读/修改更好（便于调试和测试）。

第一次功能需求：

1. 实现登录功能

(1) 管理员必须登录系统之后才能操作后台数据；顾客未登录系统时，可以查询商品信息。若要购买商品，必须登录系统。

(2) 管理员和顾客通过账户名和密码登录系统。

(3) 管理员的账户和密码通过提前预设的方式指定。

(4) 顾客在首次登录系统时，需要先进行注册，注册成功后设置账户和密码（注册时需要提供的信息由你自行设计）。

2. 管理顾客的信息

- (1) 保存顾客注册的信息和账户信息；
- (2) 顾客可以修改自己的密码。

功能验证测试用例（参考）：

1. 验证管理员账户是否成功建立。(1) 给出正确的管理员账户名和密码，能够成功登录系统；(2) 给出错误的管理员账户名或密码，登录失败，并给出恰当的提示。
2. 验证顾客账户创建功能。(1) 注册功能，现场演示新顾客注册功能，注册完成后用该账户信息登录系统，应能够成功登录。(2) 给出错误的顾客账户或密码，登录失败，并给出恰当的提示。(3) 用一个已经存在的顾客账户，现场演示修改密码。使用修改之后的密码登录系统，应能够成功登录。针对该账户，使用不正确的密码，登录失败，并给出恰当的提示。

源程序最小要求：

1. 至少创建两个类：管理员和顾客（是否使用继承机制可选）
2. 类中的数据成员的访问控制为 `private`；成员函数的访问控制 `public`

第二次功能需求：

1. 实现管理员的商品管理功能

- (1) 能够根据管理员的输入，添加新的商品信息，包括商品的类别（可自行设定）、名称、单价、描述和库存等信息（可以自行添加你认为必要的信息）；
- (2) 能够修改、删除指定商品的信息。

2. 用户查询功能

- (1) 用户可以浏览商品列表，查看商品相关信息。
- (2) 精确匹配查询：根据用户给出的关键字（如商品名称、商品类别、价格等等）查询商品信息，并显示查询结果。
- (3) **（选做）**模糊匹配查询：当用户给出的查询关键字与商品相关信息不一致时，给用户列出最接近的商品信息。

功能验证测试用例（参考）：

1. 管理员的商品管理功能。前置条件：管理员登录系统。（1）根据学生设计的商品信息，输入 1-2 条新的商品信息；查看商品列表，确认新输入的商品是否在其中，信息是否正确。（2）从商品列表中选择特定商品，进行修改/删除操作；查看商品列表，确认修改/删除的商品信息。

2. 用户查询功能。前置条件：顾客不登录系统。（1）查看商品列表，应能够正确显示所有商品信息。（2）给出任意商品信息中包含的关键字（验证 1-2 种不同的关键字），应能够显示与该关键字一致的商品信息。若所给出的关键字不在商品所包含的信息项内，应给出查询失败的提示。（3）给出任意关键字，应能够给出与其相近的商品信息。（选做内容，没做的不验收）。

3. 随机验证第一次上机需求的相关功能，确保两次上机是同一个程序，而不是两段独立的代码。

说明：（1）商品信息可以事先存储在数组中，或者每次运行时输入，也可以在文件中存储（后面有专门的持久化要求，此次不限制）。（2）用户包括管理员和顾客，两个权限都能查询商品列表。

源程序最小要求：

1. 在第一次代码的基础上，至少增加两个类：商品和商品列表，同时应表达出管理员、顾客与商品/商品列表之间的关系；商品与商品列表之间的关系。

2. 类中的数据成员的访问控制为 `private`；成员函数的访问控制 `public`。

第三次功能需求：创建购物车

每个顾客都有一个购物车，顾客只有在登录状态时，才能操作购物车。在未登录状态操作购物车，系统需提示必须先登录才能操作购物车。

1. 向购物车中添加商品。顾客可以从商品列表中将商品加入购物车中。如果添加的商品与购物车已有商品重复，提示顾客商品已存在；若顾客仍要加入，修改购物车中该商品的购买数量。

2. 删除购物车中的商品。顾客可以从购物车中删除商品，删除的方式可以是一次删除

一个商品，也可以一次删除多个商品。

3. 修改购物车中商品的购买数量。顾客可以修改购物车中商品的购买数量，若数量改为 0，默认表示要删除该商品，询问顾客是否删除商品，若选择是，删除该商品；若选择否，保持原有购买数量不变。

4. 查询购物车内的商品信息。显示顾客购物车内所有商品的信息，并给出购物车内商品的总数以及总价格。

特别说明：顾客仅能修改购物车中商品的购买数量，其余信息不能修改。商品信息只能由管理员修改，管理员修改商品信息之后，商品列表以及购物车中的商品信息同步改变。

功能验证测试用例（参考）：

1. 添加购物车。（1）先显示商品列表，从商品列表中选择一个或多个商品加入购物车，显示购物车信息，应能够看到刚加入的商品信息。（2）从商品列表中选择一个已经在购物车中的商品，加入购物车后，购物车中的商品记录数不变，重复的商品购买数量增加。

2. 删除购物车中的商品。先显示购物车信息，从中选择一个或多个商品删除，再次显示购物车信息，不存在已经删除的商品信息。

3. 修改购物车信息。（1）增加或减少某个商品的购买数量（不为 0），再次显示购物车信息，该商品的购买数量已变化。（2）将某个商品数量改为 0，应显示是否删除该商品的提示信息，选择是，删除商品；选择否，不改变商品信息。

4. 查询购物车信息。应能够显示出购物车中的所有商品信息、商品总数以及总价格。

5. 随机验证第一、二次上机需求的相关功能，确保两次上机是同一个程序，而不是两段独立的代码。

说明：购物车信息可以临时存储，也可以在文件中存储（后面有专门的持久化要求，此次不强制）。

源程序最小要求：

1. 在第二次代码的基础上，至少增加一个类：购物车，同时应表达出顾客与购物车、购物车与商品/商品列表之间的关系。

2. 类中的数据成员的访问控制为 `private`；成员函数的访问控制 `public`。

第四次功能需求：基础购物流程

顾客在登录状态才能进行购买操作，若在未登录状态，系统需提示必须先登录才能购买商品。顾客可以从购物车或商品列表进行购买，一次可以选择购买一件或多件商品。

1. 生成订单。顾客选择购买商品后，为本次购买生成订单。订单信息包括订单编号、所购商品信息、购买时间、订单总额、收货地址等。订单生成后，需要在商品列表中更新所购买商品的库存。若购买数量超过商品库存数，要提示用户库存不足。若顾客从购物车发起购买请求，订单生成后从购物车删除订单中的商品。

2. 订单状态维护。订单状态在待发货、已发货、已签收三个状态之间流转，新创建的订单默认为“待发货”状态。订单状态的转换可以由管理员手动设置，也可以通过设置时间自动控制状态转换，例如，新订单创建 10 秒之后，自动转换为已发货状态。

3. 查询订单。顾客可以查看自己所有状态的订单信息。

4. 修改订单。顾客可以修改“待发货”状态的订单，包括取消订单、修改收货地址，其余订单信息不能修改。顾客若取消订单，需同时更新商品列表中对应商品的库存。

5. 删除订单。顾客可以删除“已收货”状态的订单。

特别说明：在实现本次需求的过程中，需注意所有设计到商品库存的操作，需同时更新商品列表中的库存信息，避免数据不一致。

功能验证测试用例（参考）：

1. 生成订单（购买流程测试）。在未登录状态下尝试购买，系统应提示：“请先登录”。
从商品列表购买：（1）顾客登录后，从商品列表选择商品，确认数量；（2）生成订单，检查订单信息：订单编号、商品信息、购买时间、订单总额、收货地址正确；（3）商品库存同步减少。
从购物车购买：（1）顾客登录后，从购物车选择商品，确认购买；生成订单后检查：购物车中对应商品被删除；（2）商品库存同步减少。
库存不足情况：购买数量超过库存，系统应提示：“库存不足”，并禁止生成订单。
2. 订单状态维护。（1）订单初始状态：创建订单后，状态应为“待发货”。（2）状态自动转换：订单创建 10 秒后，自动转为“已发货”状态；可人工调整订单状态。（3）签收订单。管理员自动将“已发货”状态的订单设为“已签收”。
3. 订单查询。登录顾客账号，进入订单查询界面；能显示所有订单信息，包括订单状

态、商品详情、订单总价、创建时间和收货地址。

4. 订单修改：（1）修改待发货订单。订单为“待发货”状态时，修改收货地址；再次查询订单，地址已更新；（2）取消待发货订单。取消订单后：订单状态变为“已取消”或删除；商品库存同步增加。（3）修改已发货或已签收订单系统提示：“订单已发货或签收，不可修改”。
5. 订单删除。删除“已签收”状态的订单；再次查询订单，不再显示已删除订单。
6. 随机集成测试（之前功能回归）。随机验证顾客登录、购物车管理、商品列表更新功能；确认程序整体逻辑一致，无功能冲突。

源程序最小要求：

1. 类设计与关系体现 （1）在第三次代码基础上，需新增至少两个类：订单、订单列表。（2）体现订单列表与订单之间的关系：订单列表包含 0 到多个订单；用户（管理员、顾客）与订单/订单列表之间的关系；订单/订单列表与商品/商品列表之间的关系。
2. 访问权限与封装要求：所有类的数据成员设置为 `private`；所有类的成员函数设置为 `public`；数据成员只能通过成员函数访问或修改。

第五次功能需求：在系统中增加促销活动

系统支持商品的各种促销优惠活动。若顾客购买的商品参与促销活动，在计算订单总额时需按促销价计算。

促销活动都有有效期的限制；所有的促销活动的相关信息均由管理员进行设置。

1. 限时折扣。给商品设置折扣活动（如 8 折、9 折等），不同的商品可以设置不同的折扣。
2. 满减活动。订单总额满足限定额度给予一定的满减，如满 300 减 50 等。
3. 叠加促销活动。折扣与满减可以叠加，如果商品参与叠加促销，优先计算折扣再计算满减。
4. 若商品参与促销活动，在商品列表中需显示相应促销标签（折扣或是满减）。

功能验证测试用例（参考）：

1. 功能验证测试用例：

前置条件： 管理员成功登录系统。

场景	操作步骤	测试数据示例	预期结果
限时折扣设置	设置商品折扣活动	可乐（9折，期限：2025-04-01至2025-04-10）	商品列表显示“9折”标签，有效期内购买时按9折计算
满减活动设置	设置满减优惠活动	满300减50，期限：2025-04-01至2025-04-15	商品列表显示“满300减50”标签，订单满足条件时自动减免
叠加促销活动	同时设置折扣与满减	饮料类商品8折，满200减20	商品列表显示“8折”“满200减20”标签，顾客购买时叠加优惠

2. 顾客购买促销商品的订单金额计算

前置条件： 顾客已登录。

场景	操作步骤	测试数据示例	预期结果
单一折扣计算	购买参与折扣活动商品	原价10元商品（9折），数量2	实际支付金额： $10 \times 2 \times 0.9 = 18$ 元
单一满减计算	订单金额满足满减活动	商品订单总额310元，促销满300减50	实际支付金额： $310 - 50 = 260$ 元
叠加促销计算	同时享受折扣与满减	饮料商品原价100元（8折），购买3件，共300元；满200减20	折扣后金额： $100 \times 3 \times 0.8 = 240$ 元，再满减后金额： $240 - 20 = 220$ 元

3. 商品列表促销标签显示

前置条件： 顾客（登录或未登录均可）浏览商品列表。

操作步骤	测试数据示例	预期结果
浏览促销商品列表	可乐（9折）；薯片（满100减10）	商品列表明确显示“9折”、“满100减10”等促销标签

4. 促销活动有效期测试

操作步骤	测试数据示例	预期结果
超过促销有效期购买	可乐促销期限至2025-04-10，2025-04-11购买	商品列表无折扣标签，购买时按原价结算

源程序最小要求：

1. 新增类设计与关系体现。(1)在第四代码基础上新增表达促销活动的类层次结构(体现出继承关系)。(2) 类关系清晰体现。

2. 封装与访问权限要求。类中数据成员全部设置为`private`；对外接口函数设置为`public`，提供相应的`get`、`set`方法，禁止外部直接修改数据成员。

第六次功能需求：数据持久化

1. 用户信息持久化。将管理员、顾客信息存储在文件中。账户信息的变化需同步到文件中。
2. 商品信息持久化。所有商品信息存储在文件中，在系统启动时，读取文件加载商品信息。管理员能够通过对文件的编辑完成商品的添加、修改和删除。商品信息文件发生修改后，需重新加载商品信息。管理员通过界面操作对商品信息的修改，需同步到文件中。
3. 购物车信息持久化。将购物车的信息存储在文件中。顾客对购物车的修改操作需同步到文件中。
4. 订单信息持久化。将订单信息存储在文件中，订单状态的变化需同步到文件中。
5. （选做）统计分析（在前 5 次的需求中已经完成了持久化，或有余力的同学）
- 对顾客的购买数据进行统计，生成统计分析数据，内容包括商品类别、商品名称、按类别分类计算的购买金额、购买频度等。

功能验证测试用例（参考）：

1. 用户信息持久化验证

步骤	操作	测试数据示例	预期结果
1	新增顾客账号	用户名：user01， 密码：pass123	用户信息文件新增记录
2	修改用户密码	修改 user01 密码为 newpass	用户信息文件中密码更新
3	删除顾客账号	删除 user01	用户信息文件删除对应记录

2. 商品信息持久化验证

步骤	操作	测试数据示例	预期结果
1	系统启动加载商品	启动系统	商品列表正确加载显示
2	文件手动修改商品	直接修改文件中“可乐” 库存为 50	重启系统，商品库存正确更新 为 50
3	界面修改商品信息	在界面修改“薯片”单价 为 6 元	文件商品信息同步更新

3. 购物车信息持久化验证

步骤	操作	测试数据示例	预期结果
1	添加购物车商品	用户 user01 添加可乐 2 瓶	购物车文件新增记录
2	修改购物车商品数量	user01 购物车可乐数量改为 3 瓶	文件同步修改数量为 3
3	删除购物车商品	user01 删除购物车内可乐	文件删除对应商品记录

4. 订单信息持久化验证

步骤	操作	测试数据示例	预期结果
1	创建订单	user01 创建订单：薯片 5 包	订单文件新增记录
2	修改订单状态	订单状态由“待发货”变为“已发货”	文件订单状态同步更新为“已发货”
3	删除订单	user01 删除已签收订单	文件删除该订单记录

5. 文件与界面数据一致性验证。（1）随机选择多次进行文件手动修改，验证系统界面数据自动同步；（2）随机通过界面操作，验证文件中数据自动同步更新；（3）随机集成验证第二次（商品管理）和第三次（购物车管理）功能，确保系统整体逻辑稳定。

6. （选做）顾客购买数据统计分析验证

步骤	操作	测试数据示例	预期结果
1	顾客购买数据生成	user01 购买可乐 10 瓶、薯片 20 包	系统统计文件记录准确
2	统计数据 分析	查看商品购买频度及金额	显示正确统计结果，含购买类别、名称、频度和金额

源程序最小要求：

1. 访问权限与封装要求：所有类的数据成员设置为 `private`；所有类的成员函数设置为 `public`；数据成员只能通过成员函数访问或修改。
2. 在第 5 次代码基础上，新增数据持久化管理类，负责用户、商品、购物车、订单信息读写文件操作。
3. 文件不存在、文件读取失败、数据格式异常均明确处理；提供具体错误提示和日志信息，便于排查问题。