

电商交易平台设计与实现

随着移动互联网的发展，电商平台已经广泛地融入人们的生活。此次作业的任务是使用 C++ 语言，基于面向对象的程序设计方法，设计并实现一个简单的电商交易平台，提供用户管理、商品管理、交易管理等功能。

本作业包含三个题目，使得该电商交易平台功能逐步增强。前两个题目为单机版，运行时体现为一个进程。第三个题目为网络版，要求采用 CS 结构，客户端和服务端为不同的进程。注意完成题目要求，建议先有系统的整体设计方案，再去分三阶段逐步实现系统，避免后期程序改动巨大。

本作业必须提交三个程序，即每个题目各提交 1 个程序，不能由于第三个程序满足前两个的功能要求，就只提交一个。每个程序的代码只完成自身的功能，三个程序之间应该体现出功能逐步增加的关系。

除程序外，本作业要求提交实验报告，可以提交 word 或者 pdf 版本。报告内容至少包含：整个程序（最后产出的程序）的总体设计，以及实现中的重要问题和解决方案、想法、经验、教训等。报告格式参考模板，可自行调整、修改，但要求组织结构合理，描述清楚。

题目一：账户管理子系统和商品管理子系统（单机版）

题目一程序要求支持以下功能：

- 1) 用户注册&登录：支持新用户注册平台账号，已注册用户用平台账号登录平台，要求已注册用户的信息长久保留。
- 2) 修改账户密码：支持登录后对用户账号的密码修改。
- 3) 余额管理：支持用户账号中余额的查询、充值、消费等。
- 4) 添加商品：支持商家添加新商品，要求已添加的商品信息长久保留。
- 5) 展示平台商品信息：支持针对不同类型用户、无论登录与否均展示平台商品信息。
- 6) 搜索平台商品信息：支持依据某种条件（比如：名称）对平台商品进行筛选，并展示筛选结果。
- 7) 商品信息管理：支持商家对其商品的信息进行管理，包括但不限于价格管理、剩余量管理等。

主要要求如下：

- 要求 1：账户至少需要账号名、密码、账户余额、账户类型（商家/消费者）等内容。

- 要求 2: 请把所有的用户账户信息写入文件 (要求使用文件存储各类信息, 以练习对文件的操作和流的输入输出), 注册新账户的时候, 要求注册的账户名不能已经存在于文件中, 即账户名唯一。
- 要求 3: 请至少设计一层继承体系 (用户基类-用户子类)。设计一个用户基类, 然后让商家类和消费者类等用户子类继承它, 具体的用户是用户子类的实例对象 (<<张三 >> 是消费者类的实例对象)。用户基类为抽象类, 不能实例化, 请至少具有一个纯虚函数 getUserType()用于指示用户类别。
- 要求 4: 电商平台上至少有三类商品: 如食物、服装、图书等, 每类商品中至少有三个具体的商品 (如图书中可以有《C++ Primer》、《Effective C++》等), 每个具体的商品请至少包含商品描述、商品原价、商品剩余量等数据。所有的商品信息需要存储在文件或数据库中, 不能写在代码中。
- 要求 5: 请至少设计一层继承体系 (商品基类-商品子类)。设计一个商品基类, 然后让图书类、电子商品类和服装类等商品子类继承它, 具体的商品是商品子类的实例对象 (<<C++ Primer >> 是图书类的实例对象)。商品基类请至少具有一个虚函数 getPrice()用于计算具体商品的价格。
- 要求 6: 支持对同一品类下所有产品打折的活动。
- 要求 7: 支持一定的错误场景处理能力, 例如读文件错误, 输入数据不合法等等。

题目二：交易管理子系统（单机版）

在题目一的基础上支持在电商平台上购物的功能, 请实现以下功能:

- 1) 购物车管理: 支持消费者向购物车添加、删除指定数量的商品, 也支持消费者修改当前购物车商品的拟购买数量。
- 2) 订单生产: 选择购物车的商品生成订单, 计算并显示订单总金额。
- 3) 网上支付: 消费者使用余额支付订单, 支付成功后, 消费者被扣除的余额应转至商家余额中。

主要要求如下:

- 要求 1: 当订单生成后, 处于未支付状态时, 应对应数量的商品冻结, 不能参与新订单的产生, 避免商品被超额售卖。
- 要求 2: 支持一定的错误场景处理能力, 如生成订单失败, 支付失败等。

题目三：电商交易平台（网络版）

在题目一、二的基础上, 将单机版电商交易系统修改为网络版。网络版要求

实现如下功能：

- 1) 用户登录：用户通过客户端以账号密码登录平台。
- 2) 展示平台商品信息：通过客户端展示平台商品信息。
- 3) 搜索平台商品信息：通过客户端依据商品名称对平台商品进行搜索筛选，并展示筛选结果。
- 4) 购物车管理：通过客户端支持消费者向购物车添加、删除指定数量的商品。
- 5) 订单生产：系统生成订单，通过客户端展示用户的订单信息。
- 6) 订单支付：用户通过客户端向系统提交订单支付的申请，并展示系统完成支付的状态。

主要要求如下：

- 要求 1：网络版需要实现的功能的要求与单机版要求一致。
- 要求 2：要求采用传统 CS 结构而非 BS 结构，客户端与服务器系统之间使用 socket 进行通信，不能使用 rpc 框架。
- 要求 3：支持一定的错误场景处理能力。

作业的其他要求

(1) 程序设计要求

- 如有必要的友元函数，要在实验报告和程序中说明每个友元函数的不可替代性，为什么一定要用友元才能实现。
- 自己编写的代码，除主函数和必要的友元函数外，不允许出现任何一个非类成员函数。
- 任何不改变对象状态（不改写自身对象数据成员值）的成员函数均需显式标注 `const`。

(2) 代码规范性要求

- 代码需遵循课件提出的编码规范要求。
- 通过开发环境自动生成的界面类代码（如有），全部数据成员和成员函数需在类声明时加以注释，函数体内的必要步骤要加以注释。
- 其他全部类代码的数据成员和成员函数的声明和实现均需加以注释，成员函数的必要步骤要加以注释。