

Программирование
Домашняя работа 2
Интернет-магазин



Общая информация

Дедлайн: 13 июня, 23:59

Отправка решения: через Canvas LMS в виде единого архива формата zip.

Поздние отправки будут засчитаны с понижением оценки (см. правила во вводной презентации курса). Рекомендуем Вам начать отставку итогового варианта не позднее 23:00 на случай технических проблем на стороне сервера либо на стороне Вашего интернет провайдера.

Общие требования

- Работы будет проверяться вручную. Ваша программа должна быть ориентирована на пользователя и должна информативным способом взаимодействовать с ним (запросы необходимо выполнять с подсказками, вывод с сопровождающими сообщениями)
- Следуйте общепринятой стилистике кода на Python. Строгая проверка PEP-8 проводиться не будет, но если код будет сложным для чтения и понимания из-за многочисленных несоответствий общепринятой стилистике, оценка может быть снижена
- Используйте именованные константы вместо “магических чисел”.
- Используйте принципы структурного программирования. Разбивайте логику на отдельные небольшие функции, функции группируйте в модули.
- В программе должна быть защита от некорректных действий пользователя. Общий принцип: ни одно действие пользователя (например, ввод строки вместо числа) не должно приводить к аварийному завершению программы.

Описание задания

В данном домашнем задании вам необходимо реализовать прототип интернет-магазина с двумя разными интерфейсами (для клиента и администратора / владельца). Каждый интерфейс реализуется в виде отдельного консольного приложения Python, но оба приложения должны работать с общим модулем, реализующим ключевые функции системы.

Упрощенная модель интернет-магазина состоит из следующих составных частей (в скобках указаны атрибуты):

- Пользователи (имя, логин, пароль)
- Товары (наименование, текущая цена, текущее количество на складе)
- Заказы (номер, дата создания, статус, набор позиций). Каждая позиция заказа связана с товаром из каталога интернет магазина, в ней указывается цена и количество в рамках заказа.

С точки зрения приложения клиента необходимо реализовать следующие функции:

- Просмотр каталога товаров
- Авторизация в системе по логину и паролю

- Создание заказа, добавление в него товаров. В заказ можно добавлять только те товары, которые есть на складе. Номер заказа должен назначаться автоматически но так, чтобы он не совпадал с уже имеющимися в системе заказами.
- Подтверждение и оплата заказа. В момент подтверждения заказа необходимо еще раз убедиться в том, что все товары есть в необходимом количестве на складе (так как в интервале между добавлением товара в корзину и подтверждением заказа клиентом кто-то другой может купить товар либо администратор может отредактировать данные).
Операции, связанные с оплатой заказа, находятся за пределами данного ДЗ. Считается, что оплата всегда проходит успешно.
- Просмотр списка своих заказов
- Просмотр отдельно взятого заказа
- Редактирование открытого заказа (добавление, удаление позиций, уменьшение/увеличение количества товара в рамках одной позиции)

Просмотр клиентом каталога товаров должен быть возможен без авторизации в системе, однако, все операции с заказами могут выполняться только после входа в систему.

С точки зрения владельца интернет-магазина необходимы следующие функции:

- Просмотр каталога товаров
- Изменение текущей цены отдельного товара
- Изменение количества отдельного товара на складе
- Просмотр открытых заказов (заказов, которые оплачены, но не доставлены клиенту)
- Изменение статуса заказа (перевод заказа либо в состояние “Отправлен” либо в состояние “Доставлен”)

Любой заказ может находиться в следующих состояниях:

1. Создан - клиент создал заказ, но еще может его редактировать, оплата не произведена
2. Оплачен - клиент оплатил заказ, редактирование заказа больше не возможно
3. Отправлен - заказ собран и отправлен клиенту
4. Доставлен - заказ доставлен клиенту

Создание заказа, а также переход из состояния “Создан” в состояние “Оплачен” всегда происходит на стороне клиента. Переходы между состояниями “Оплачен” → “Отправлен” → “Доставлен” происходит на стороне приложения владельца.

Важно: текущая цена товара может не совпадать с ценой товара в заказах (один и тот же товар может иметь разную цену год назад, полгода назад и в настоящий момент времени). Итоговая цена каждого товара в заказе фиксируется в момент его подтверждения и оплаты клиентом.

В реальности все системы интернет-магазинов строятся с использованием базы данных, которая решает важнейшую задачу одновременной работы с общими данными многих пользователей. Для данного домашнего задания можно считать, что в каждый момент времени с системой работает только один пользователь, поэтому для хранения данных будет использоваться обычный файловый ввод-вывод.

Рекомендуемый порядок выполнения задания

1. Создайте проект с тремя отдельными файлами исходного кода (client.py (приложение для клиента), owner.py (приложение для владельца) и common.py (модуль для общих функций двух приложений))
2. С учетом общего описания системы выше выберите структуры данных для программы. Использование объектно-ориентированного подхода в данном домашнем задании не является обязательным, можно опираться на стандартные коллекции Python. На первом этапе рекомендуется создать в общем модуле тестовый набор данных (без использования внешних файла(ов) с данными).
3. Реализуйте необходимые функции двух приложений, протестируйте основные сценарии. Особое внимание обратите на следующую группу сценариев:
 - (a) Клиент авторизуется в системе
 - (b) Клиент создает заказ, добавляет в него ряд товаров, после чего выходит из системы
 - (c) В систему заходит администратор и изменяет текущее количество товара (ранее добавленного клиентом в корзину) на складе и/или его цену, после чего выходит из системы
 - (d) Клиент повторно входит в систему. Важно, чтобы при просмотре ранее созданного заказа клиент видел, что какой-то товар уже считается по обновленной цене. Также в момент оплаты необходимо еще раз проверять количество товара на складе. Если владелец на предыдущем шаге изменил его в меньшую сторону и необходимого клиенту количества товара нет на складе, следует вывести сообщение и не допустить перехода заказа в состояние "Оплачен". У клиента после этого будет возможность удалить товар из корзины либо уменьшить его количество.
4. Добавьте в программу файловый ввод-вывод. Формат и количество файлов никак не регламентируются. В файле(-ах) необходимо хранить всю информацию интернет-магазина (пользователи, товары, заказы). При выполнении любой операции изменения данных как на стороне приложения клиента, так и на стороне приложения владельца, необходимо сразу перезаписывать файл(ы), не дожидаясь выхода из системы.

Интерфейс приложения

Интерфейс приложения полностью остается на Ваше усмотрение. Единственное требование - взаимодействие с пользователем должно быть информативным, на каждом этапе пользователь должен четко понимать, что от него требуется.

Рекомендации

- Состояния заказа можно хранить в виде класса с числовыми константами (атрибутами класса):

```
class OrderStatus:  
    CREATED = 1  
    PAID = 2  
    ...
```