Restaurant Ordering App

Proposal

Vaughn Cox

Purpose

This project is designed to demonstrate understanding of core Object-Oriented Programming concepts taught in CSCI 121, among other things we have learned so far, as well as retaining a lot o the information learned from and used in the Bank assignment.

Overview

The Restaurant Ordering App simulates an online fast-food ordering system. It's structured similarly to the banking project, but repurposed for a restaurant context. The program models customer accounts, menu navigation, and the purchasing process in a text-based environment.

Key Classes

The **Account** class manages the customer's balance. It allows deposits and handles automatic deductions when purchases are made. This class is based on the simpler form of the checking account class from the bank project, with unnecessary features removed.

The **PurchaseManager** class presents available food items and prices using a menu() method. The start() method handles input, selection, and payment logic, drawing from the account as needed.

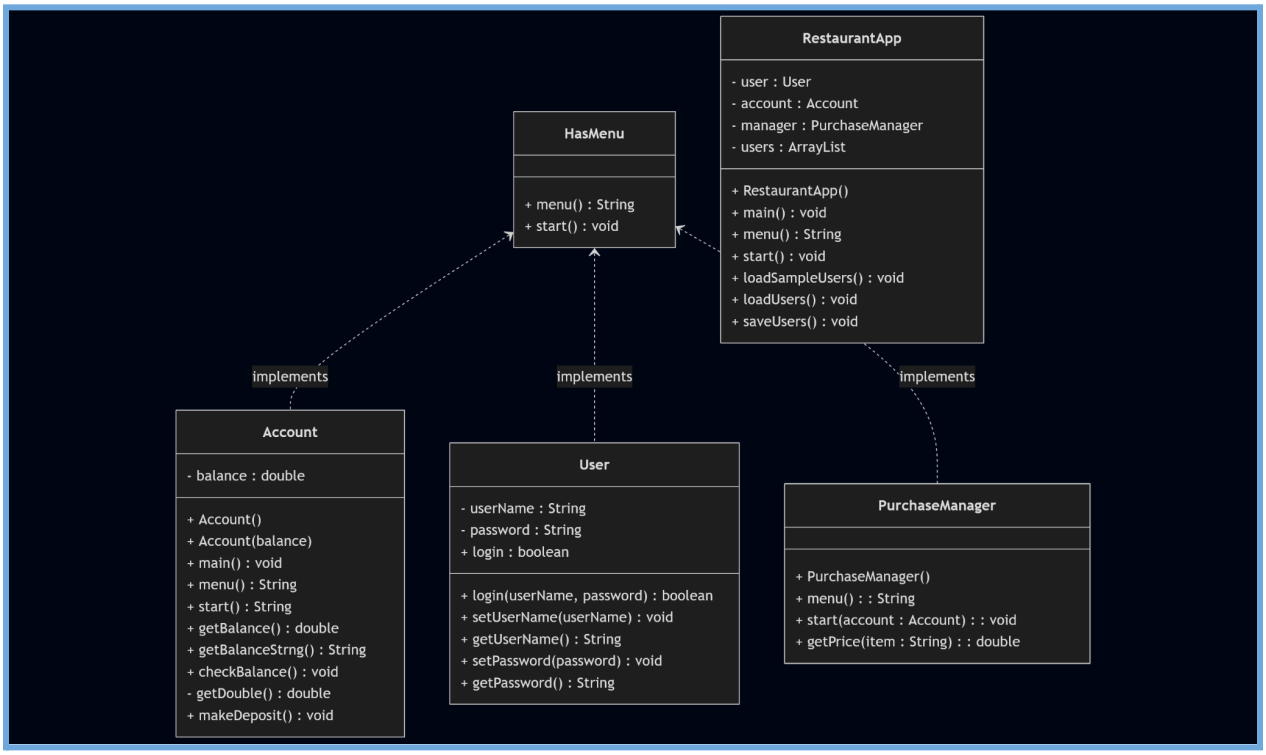
The **HasMenu** interface is reused exactly as in the banking app. It allows for the implementation of the menu() and start() methods across both the Account and PurchaseManager classes.

A **main RestaurantApp** class will tie the components together, initializing the account and handling the overall program loop.

Anticipated Use Cases

The app simulates a user logging in, checking their balance, viewing a fast-food menu, and purchasing food. The system automatically checks if enough funds exist before allowing a transaction. All interaction is done through a simple command-line interface.

UML



Technology / Implementation

The app will be written in Java using only built-in libraries and no GUI components. It will run in a command-line environment and use standard input and output for interaction. It will follow standard Java conventions and run on any system with a JRE installed.

Milestones

- UML approval
- Basic Account functionality
- Menu display (PurchaseManager)
- Start input flow and purchase logic

- Full system integration (RestaurantApp)
- Balance check and error handling
- Add saving/loading with serialization
    - Optional: After all other milestones, perhaps a rewards system.