Group Members: Emily Yan, Obaidullah Darwishi, Julian Juarez, and Keith Cacayorin

Project Summary

- A cafe menu management system that organizes menu items, allows for searching and sorting, and provides information about items when requested
- We'll be using a LinkedList as our primary data structure to store menu items

OOP Requirements

Abstraction

- The MenuItem class is abstract, providing a common interface for all menu items while deferring specific implementation details to subclasses.
- Abstract methods like getCategory() ensure all MenuItem subclasses implement necessary functionality.

Encapsulation

- Private attributes with public accessor methods control access to data
- E.g. price data in MenuItem and customer information in Customer class

Polymorphism

- The system uses polymorphism through the MenuItem class hierarchy.
- The Menu class can work with any MenuItem subtype (Food or Beverage) without knowing the specific subclass.
- This enables operations like searching and sorting to work across different menu item types.

Data Structures Implementation

- LinkedList
- The system uses a LinkedList to store and manage menu items.
- This fulfills the project requirement for utilizing a LinkedList data structure.

The LinkedList Implementation will support:

- Dynamic addition and removal of menu items
- Traversal for search operations
- Iteration for display and sorting

Group Members: Emily Yan, Obaidullah Darwishi, Julian Juarez, and Keith Cacayorin

Algorithms Implementation

- Search Algorithm
    - Linear search will be implemented to find menu items by name, category, or other criteria.

Sorting Algorithm

- The Menu class includes sortByPrice() and other sorting methods.
- These will implement the sorting algorithms

File I/O Implementation

- The Menu class includes loadFromFile() and saveToFile() methods.
- These methods handle reading menu items from external files and saving updates.

Stages of Development

1. **Core Class Implementation**
    - Develop the Person and MenuItem hierarchies
    - Implement basic composition and methods for each class

2. **LinkedList Integration**
    - Create the LinkedList data structure for storing menu items
    - Implement basic operations (add, remove, traverse)

3. **Algorithm Development**
    - Implement search functionality
    - Develop sorting capabilities for menu items

4. **File I/O Integration**
    - Add file reading capabilities for initial menu loading
    - Implement file writing for menu updates

5. **User Interface Development**
    - Create interfaces for both employees and customers
    - Implement different views based on user roles

Group Members: Emily Yan, Obaidullah Darwishi, Julian Juarez, and Keith Cacayorin

UML Diagram