* 1. 1. Class Design
  2. Why you have single class or multiple classes in your design.
  3. In my design, I opted for multiple classes to ensure a clear separation of concerns and to adhere to the principles of object-oriented programming (OOP). By breaking down the functionality of the application into smaller, more specialized classes, I was able to achieve better encapsulation and maintainability. For example, I separated the model, view, and controller logic into distinct classes to facilitate modularity and ease of testing. This design approach promotes code reuse and makes it easier to extend or modify the application in the future.

2. Data structure

- Why choosing your data structure (used in the solution).

I chose HashMaps for storing order details and customer information in my solution due to their efficiency in data retrieval and their ability to provide fast access to stored elements. With HashMaps, I can use unique keys to store and retrieve order details and customer information quickly, which is essential for a pizza ordering application where quick access to data is necessary for order processing.

-Compare your chosen data structure with at least one more data structure. Your comparison should be talking about performance.

Comparing HashMaps with alternative data structures:

HashMaps:

Advantages:

Efficient for data retrieval and storage.

Provides constant-time performance for basic operations such as get and put.

Allows for fast access to stored elements using unique keys.

Drawbacks:

May consume more memory compared to other data structures like ArrayLists or LinkedLists.

Iterating over elements may not guarantee any specific order, which could be a drawback depending on the application's requirements.

Alternative Data Structure (e.g., ArrayList):

Advantages:

Offers predictable iteration order, which may be beneficial in certain scenarios.

Can be more memory-efficient for small datasets.

Drawbacks:

Slower performance for data retrieval compared to HashMaps, especially for large datasets.

May not provide fast access to elements based on unique keys, which could impact efficiency in a pizza ordering application where quick data access is crucial.

Overall, the choice of HashMaps was motivated by their efficiency in data retrieval and their ability to provide fast access to stored elements, which are essential requirements for managing order details and customer information in my solution.