The Iterator Pattern

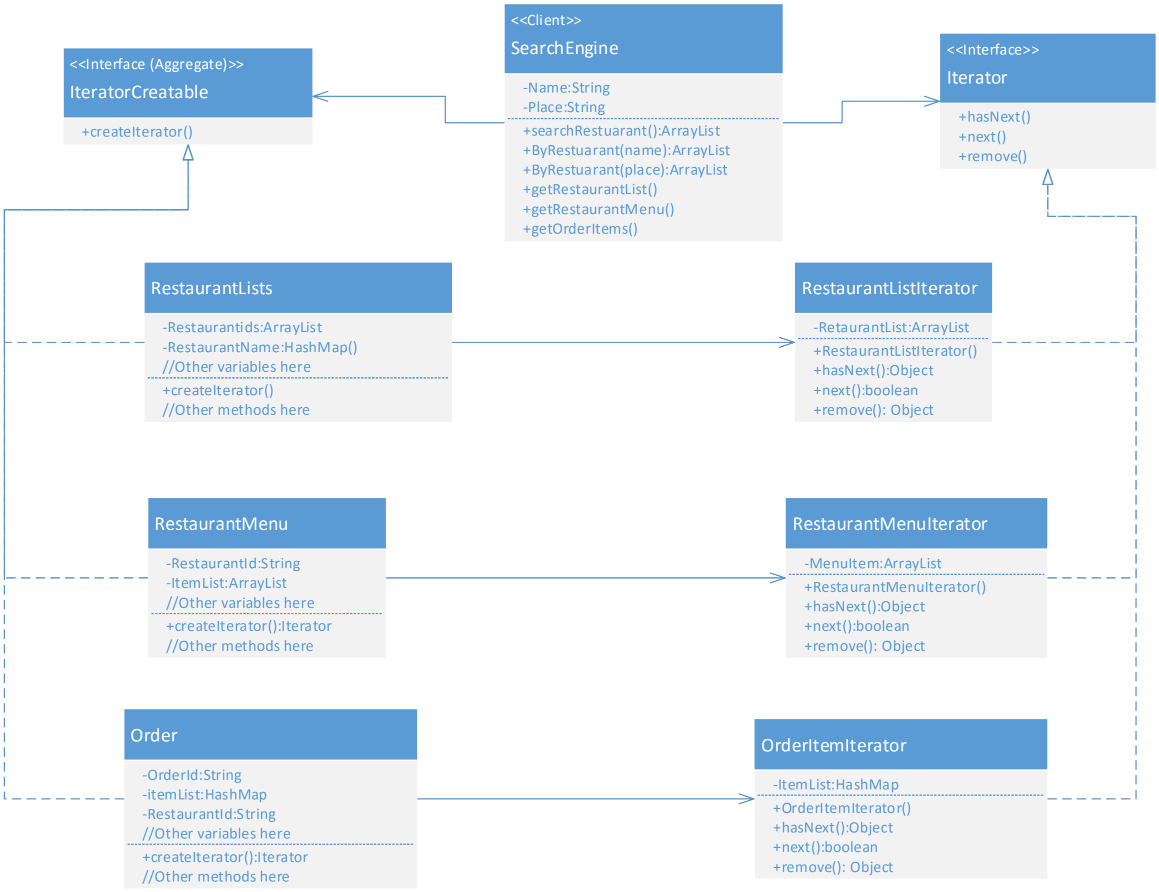
Object Oriented Goal: Traversal of elements of an aggregate without exposing the underlying implementation.

Definition: Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation.

Implemented on classes:

1. SearchEngine
2. IteratorCreatable
    a. RestaurantLists
    b. RestaurantMenu
    c. Order
3. Iterator
    a. RestaurantListsIterator
    b. RestaurantMenuIterator
    c. OrderItemIterator

Design Snapshot:



Design Explanation:

We need a way to iterate over different types of data structures present in different classes. We need a common interface which defines a way to iterate over a collection of elements. In this design, the Pending Orders and Order Items need a common interface through which the Order handler can easily traverse the elements from both the classes.

We can implement the Iterator pattern in this case. The below points explain the implementation of Iterator Pattern in our design.

- The SearchEngine is the client class which requires to traverse over the restaurant menu, restaurant list and order list collection.
- The IteratorCreatable is the interface which has an abstract method createIterator() which will be implemented by the classes which needs to create an iterator for themselves.
- The RestaurantLists, Order and RestaurantMenu are the classes which have the different types of data structures to store the lists of items respectively. These classes implement the IteratorCreatable interface and so also implement the method createIterator() which creates particular concrete iterator for the class to traverse the list of elements.
- The Iterator interface defines a set of methods which needs to be implemented by the concrete iterators so that traversal methods can be implemented.
- The RestaurantListsIterator, OrderItemIterator and RestaurantMenuIterator are the concrete Iterator classes which implements the methods hasNext(), next() and remove() for the traversal and removal of elements for the respected collections of the classes.
- Now, the Order Handler client only knows about the IteratorCreatable interface & the Iterator interface. Whenever the SearchEngine class wants to iterate or traverse over the collections from RestaurantLists, Order and RestaurantMenu class, it deals with the interfaces mentioned above and uses the methods from the concrete iterators to iterate over the elements.