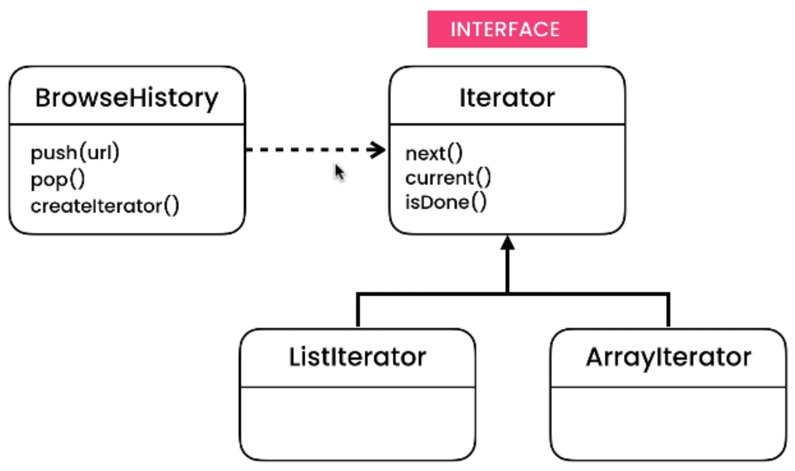
Iterator Design Pattern

This pattern is used when there is a need to traverse elements of a collection without depending on its underlying representation.

Class Diagrams



Explanation

Consider a web browser, it has the concept of history. We must have the ability to traverse URLs. the BrowseHistory class was created to save the URLs with the help of posh and pop methods. methods in the Iterator interface are needed to remove dependency between the main code and the BrowseHistory class, so they are used instead of using the posh and pop directly. These three methods can't place in the BrowseHistory class because of the single responsibility principle. The implementations of the Iterator methods are in the ListIterator and Arrayliterator classes. If the data structure for saving URLs in the BrowseHistory class is a list, then we create ListIterator class and if it is an array, we create Arrayliterator. In this implementation, if the data structure of saved URLs in BrowseHistory class changes from list to array, just adding Arrayliterator class is needed, and that works instead of ListIterator class. in this way, the changes will not cascade in the main code.

- ListIterator or Arrayliterator are implemented in BrowseHistory class to access its private fields.