Popular implementations of List, Set and Map

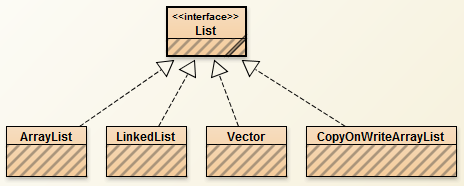
In previous post we have seen [List interface](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-5list-interface.html), [Map Interface](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-3map-interface.html), we also saw [difference between List and Set](http://data-structure-learning.blogspot.com/2015/05/difference-between-list-and-set.html) and [when to you either of them](http://data-structure-learning.blogspot.com/2015/05/when-to-use-list-and-set.html).

I have written post on [iterating list(6 ways)](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-6iterating-over.html) and iterating [map(5 ways)](http://data-structure-learning.blogspot.com/2015/05/different-ways-to-traverse-map.html).

[Java Collections](http://data-structure-learning.blogspot.com/p/java-collections.html) are used almost daily in java programs. You can read about [Java Collections Interview Question here](http://data-structure-learning.blogspot.com/p/java-collections_26.html).

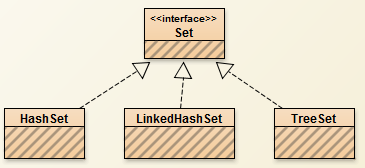
Now let us see what the popular implementations of these popular interfaces are.

List Interface popular Implementations:



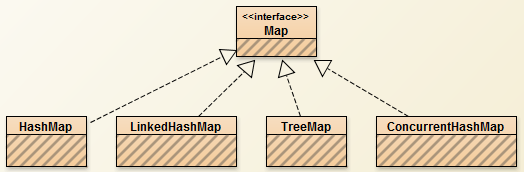
[ArrayList](http://data-structure-learning.blogspot.com/2015/05/explaination-arraylist-class.html), Vector and CopyOnWriteArrayList support Random access. LinkedList class implements the linked list with previous and next pointers. ArrayList and LinkedList are not thread safe and can be synchronized externally by Collections.synchronizedList(new ArrayList<>()). Read about [synchronized method here](http://data-structure-learning.blogspot.com/2015/06/synchronized-collections-by-collections.html). CopyOnWriteArrayList is thread safe version of ArrayList.

Set Interface popular Implementations:



HashSet, LinkedHashSet and TreeSet are popular implementations of Set interface. HashSet is backed by HashMap and it does not maintain the insertion order. LinkedHashSet maintains the insertion order. TreeSet maintains the natural order via [Comparable](http://data-structure-learning.blogspot.com/2015/06/comparable-interface-for-sorting.html) and [Comparator](http://data-structure-learning.blogspot.com/2015/06/comparator-interface.html). All of these implementations are not synchronized. They can by synchronized externally by Collections.synchronizedSet(new HashSet<>()). Read about [synchronized method here](http://data-structure-learning.blogspot.com/2015/06/synchronized-collections-by-collections.html).

Map Interface Popular Interfaces:



HashMap, LinkedHashMap, TreeMap and ConcurrentHashMap are popular implementations of Map interface. HashMap and ConcurrentHashMap do not maintain order of insertion while LinkedHashMap maintains order of insertion. TreeMap maintains natural order via [Comparable](http://data-structure-learning.blogspot.com/2015/06/comparable-interface-for-sorting.html) and [Comparator](http://data-structure-learning.blogspot.com/2015/06/comparator-interface.html). TreeMap has a tree structure of Red-Black tree.

HashMap, LinkedHashMap and TreeMap are not thread safe while ConcurrentHashMap is thread safe. Hashtable is also one of the implementation of Map interface and is thread safe but slow as compared to ConcurrentHashMap. [HashMap and Hashtable have several differences](http://data-structure-learning.blogspot.com/2015/05/difference-between-hashtableand-hashmap.html). You can synchronize the implementations of Map externally by Collections.synchronizedMap(new HashMap<>()). Read about [synchronized method here](http://data-structure-learning.blogspot.com/2015/06/synchronized-collections-by-collections.html).

Read about difference between List, Set and Map. Also read about when to use List, Set and Map.