Collections

Collection interface is the root of the Collection framework in java

* The List,Queue, set implements Collection interface

List:

Contains Ordered elements

May include duplicates

Supports index based search, can be easily inserted irrespective of the position

->List is extended by ArrayList, LinkedList, Vector..Stack extends Vector

---------------------------------------------------------------------------------------------------

**ArrayList :** 1.We can grow the size if we want to add more elements(dynamic resizing 50% of original size)

2. Non synchronized

**Linked List :**

1.Implements List and Deque interfaces 4.Not support accessing elements randomly

2. Maintains insertion order 5.Use ListIterator to iterate LinkedList elements

3. Non synchronized 6.Have duplicate and null values

**Vector:**

1.It is Synchrozied 4. Vector increase size by doubling arraysize

2.It maintains insertion order 5.Its legacy class

3.It is thread safe

**Stack :**

1.LIFO

2.Elements are added as well as removed from rear end

**Set:(I) (Hashset,LinkedHashset,TreeSet are classes.Sortedset is Interface)**

Doesn’t define an order for the elements

**No Duplicates are allowed**

**Doesn’t support index based search**

Hashset:

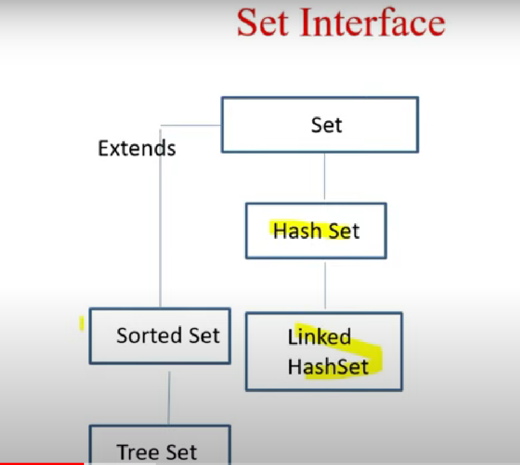
1.It implicitly implements hashtable 3.Only one null element can be added

2.Contains only unique elements 4.Its unordered as set

LinkedHashSet:

1.**Ordered version of hashset** which maintains a doubly Linked List across all elements

2.It preserves insertion order



**Sorted Set:**

1.All elements of a Sortedset must implement **Comparable** interface

2.It’s a set sorted in **ascending order**

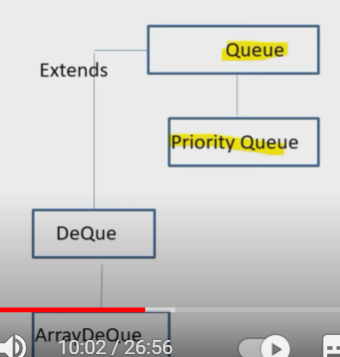
**TreeSet:**

1. Uses tree for storage(self balancing binary search tree like red black tree(used for sorting purpose))
2. Objects are stored in sorted and ascending order

**Queue: (**Here PriorityQueue,ArrayDeque are classes , Queue,DeQue are interfaces**)**

Follows FIFO approach – First in First Out

Elements added at rear end and removes from front end



Priority Queue:

1.It is Queue with priority associated with each element

2.High priority element is served before a low priority element irrespective of their insertion order

Deque:

1.Deque refers to double ended queue

2.Elements can be added and removed from either end

ArrayDeque:

1.way to apply resizable- array in addition to implementation of deque interface

2. no capacity restrictions

Map :

1.Map doesn’t implements the Collection

2.Can have duplicates

3.Represents key , value pair

4.It can only contain a unique key