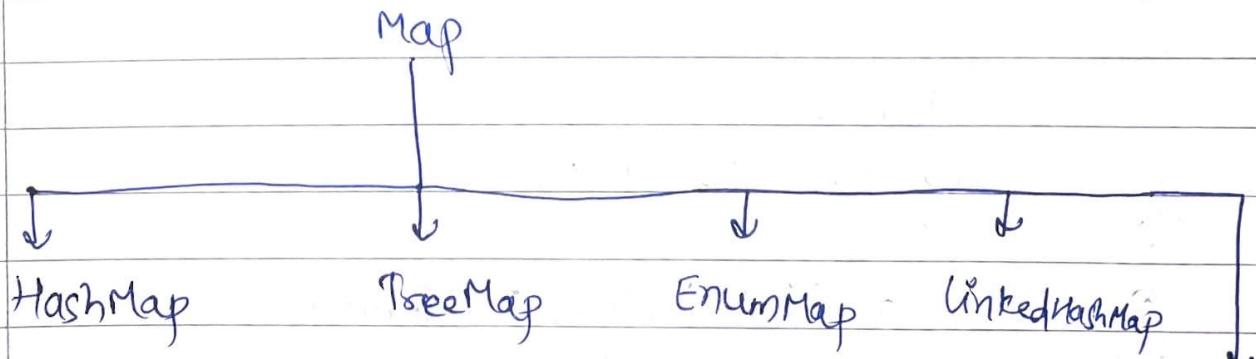
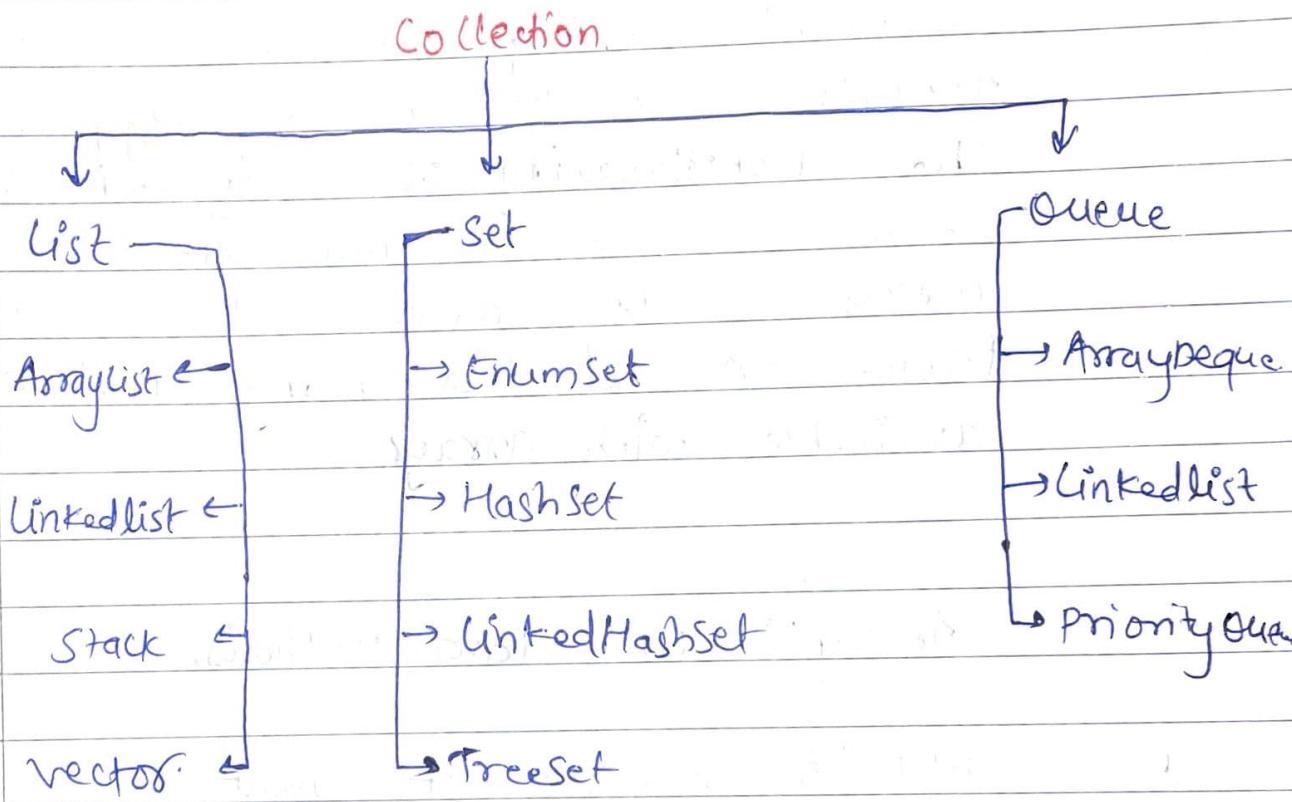


Java Collection framework



Iterator

listIterator

ArrayList :-

ArrayList is a Java class implemented using the List interface. Java ArrayList, as the name suggests, provides the functionality of a dynamic array where the size is not fixed as an array. Also as a part of the collection framework, it has many features not available with arrays.

ArrayList in Java methods.

- ① add (int index, Object element).
- ② add (Object o).
- ③ addAll (Collection c).
- ④ addAll (int index, Collection c).
- ⑤ clear () .
- ⑥ clone () .
- ⑦ contains (Object o) .
- ⑧ size () .
- ⑨ set (int index, E element) .
- ⑩ remove (Object o), remove (int index) .
- ⑪ removeAll (Collection c) .
- ⑫ removeIf (Predicate filter) .
- ⑬ removeRange (int fromIndex, int toIndex) .
- ⑭ get (int index) .
- ⑮ indexOf (Object o) .
- ⑯ isEmpty () .

Stack :-

The stack is a linear data structure that is used to store the collection of objects. It is based on LIFO.

Methods in stack class.

- ① empty()
- ② peek()
- ③ pop()
- ④ push(object element).
- ⑤ search (object element).

LinkedList :-

LinkedList is a part of the collection framework present in java.util package. This class is an implementation of the LinkedList data structure which is a linear data structure where the elements are not stored in contiguous locations and every element is a separate object with a data part and address part. The elements are linked using pointers and addresses. Each element is known as a node.

Methods for Java LinkedList :-

- ① add(int index; E element).
- ② add(E e).
- ③ addAll(int index, Collection<E> c).
- ④ addAll(Collection<E> c).
- ⑤ addFirst(E e).
- ⑥ addLast(E e).
- ⑦ clear().
- ⑧ clone().
- ⑨ contains(Object o).
- ⑩ descendingIterator().
- ⑪ element().
- ⑫ get(int index).
- ⑬ getFirst().
- ⑭ getLast().
- ⑮ indexOf(Object o).
- ⑯ lastIndexOf(Object o).
- ⑰ listIterator(int index);
- ⑱ offer(E e).
- ⑲ offerFirst(E e).
- ⑳ offerLast(E e).
- ㉑ peek().
- ㉒ peekFirst().
- ㉓ peekLast().
- ㉔ poll()
- ㉕ pollFirst()
- ㉖ pollLast()
- ㉗ pop()

`Push(E e)`

`remove()`

`remove(int index)`

`remove(Object o)`

`removeFirst()`

`removeLast()`

`Set<int, Element> removeFirst()`

`size()` returns size of collection

`toArray()`

`toString()`

A queue is a linear data structure.

Queue :-

It is a linear data structure which processes the elements in FIFO (first in first out fashion). It is an ordered list of objects, where insertion of elements occurs at the end of the list, and removal of elements occur at the beginning of the list.

Methods of Java Queue Interface.

①

`add(Object).`

→ Equivalent

②

`offer(Object).`

③

`Object remove()`

④

`poll()`

⑤

`element()`

⑥

`peek()`

Priority-Queue :-

A priority queue is used when the objects are supposed to be processed based on the priority. It is known that a queue follows the FIFO algorithm, but sometimes the elements of the queue are needed to be processed according to the priority, that is when the Priority Queue comes into play.

Methods for Java priority-queue :-

Same as Queue

ArrayDeque :-

The ArrayDeque in Java provides a way to apply resizable-array. In addition to the implementation of the Deque interface. It is also known as Array Double Ended Queue or Array Deck. This is a special kind of array that grows and allows users to add or remove from both sides of the queue.

Methods for Java ArrayDeque :-

Same as LinkedList

- ① offerFirst (element), offerLast (element).

③

Peak first ()

④

peak last ()

⑤

Pell first () like olfactory standards

⑥

Pell last () like olfactory standards

when presented to all . retronasal
and oral smell pathways play

trigeminal pathway with olfactory fibers

bottidulatory and olfactory pathways

are folded

cups to be placed to mouth

FE , olfactory pathway with most

salivary epithelium state at high levels

and short time of prior stimulation to all

regions involved with mouth, after which

so as not allow further transmission

and superior sensitivity FE , different oral environment

involves in the sense of olfaction

involves

olfactory

pathway

and trigeminal

pathway

and gustatory

pathway

Set :-

A set is a collection that cannot contain duplicate elements. It models the mathematical set abstraction. The set interface contains only methods inherited from Collection and adds the restriction that duplicate elements are prohibited.

HashSet :-

HashSet in Java is a class from the Collection framework. It allows you to store multiple values in a collection using a hash table. The hash table stores the values in an unordered method with the help of hashing mechanism. It contains unique elements.

Methods of HashSet in Java.

- ① add(E e).
- ② clear()
- ③ clone().
- ④ contains(Object o).
- ⑤ isEmpty()
- ⑥ iterator()
- ⑦ remove(Object o)

① size()

LinkedHashSet :-

- LinkedHashSet maintains a linked list of the entries in the set, in the order in which they were inserted.
- * It is an ordered version of HashSet.

Methods :-

Same as HashSet.

TreeSet :-

TreeSet provides an implementation of the set interface that uses a tree for storage. Objects are stored in a sorted and ascending order. Access and retrieval times are quite fast, which makes TreeSet an excellent choice when storing large amounts of sorted information that must be found quickly.

Methods

Same as HashSet.

Map :-

A map contains values on the basis of key, i.e. key and value pair. Each key and value pair is known as an entry. A Map contains unique keys.

HashMap :-

HashMap is the implementation of Map, but it doesn't maintain any order.

Methods of Map interface :-

- ① put (Object key, Object value).
- ② putAll (Map map).
- ③ putIfAbsent (K key, V value).
- ④ remove (Object key), remove (Object key, Object val).
- ⑤ keySet ()
- ⑥ clear ()
- ⑦ containsValue (Object value).
- ⑧ containsKey (Object key)
- ⑨ equals (Object o).
- ⑩ get (Object key).
- ⑪ isEmpty ()
- ⑫ replace (K key, V value).
- ⑬ size ()
- ⑭ getKey (), getValue (),

TreeMap :-

TreeMap is the implementation of Map and SortedMap. It maintains ascending order.

LinkedHashMap :-

LinkedHashMap is the implementation of Map. It inherits HashMap class. It maintains insertion order.