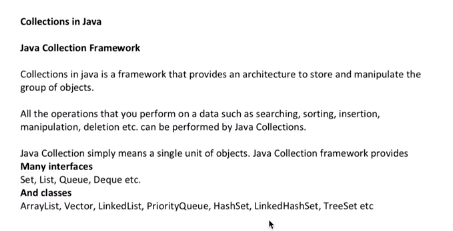
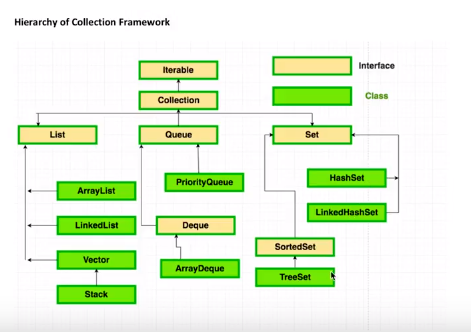
Java Collections:





**List Interface:**

**1.Array List:**

* ArrayList in Java is used to store dynamically sized collection of elements,
* An ArrayList grows its size automatically when new elements are added to it.
* ArrayList is part of Java’s collection framework and implements Java’s List interface.

**Key points to note about ArrayList in Java**

* An ArrayList is a dynamic array. It grows its size to accommodate new elements and shrinks the size when the elements are removed.
* It allows you to retrieve the elements by their index.
* It allows duplicate and null values.
* It is an ordered collection. It maintains the insertion order of the elements.
* You cannot create an ArrayList of primitive types like int, char etc. You need to use boxed types like Integer, Character, Boolean etc.
* Java ArrayList is not synchronized.

**2. LinkedList:**

Java LinkedList is a doubly linked list implementation of Java’s List and Deque interfaces. It is part of Java’s collections framework.

**Key points to note about LinkedList in Java**

* Java LinkedList maintains the insertion order of the elements.
* LinkedList can have duplicate and null values.
* The LinkedList class implements Queue and Deque interfaces.
* Java LinkedList is not thread-safe.

**Set Interface:**

**1.HashSet:**

* HashSets are used to store a collection of unique elements.
* It implements the Set interface.

**Key points to note about HashSet in Java**

* HashSet cannot contain duplicate values.
* HashSet allows null value.
* HashSet is an unordered collection. It does not maintain the order in which the elements are inserted.
* HashSet is not thread-safe.

**2.LinkedHashSet.**

A LinkedHashSet is an ordered version of HashSet that maintains a doubly-linked List across all elements.

**Key points to note about LinkedHashSet in Java**

* Contains unique elements only like HashSet. It extends HashSet class and implements Set interface.
* Maintains insertion order.

**3.TreeSet.**

* Java TreeSet class implements the Set interface that uses a tree for storage.
* The objects of the TreeSet class are stored in ascending order.

**Key points to note about TreeSet in Java**

* Java TreeSet class access and retrieval times are quiet fast.
* Java TreeSet class maintains ascending order.

**Map Interface:**

**1.HashMap:**

A Map is a collection of key-value pairs.

It maps keys to values.

**Key points to note about HashMaps in Java**

* A HashMap cannot contain duplicate keys.
* Java HashMap allows null values and the null key.
* HashMap is an unordered collection.
* Java HashMap is not thread-safe.

**2.LinkedHashMap:**

Java LinkedHashMap is a doubly linked List based implementation of Java’s Map interface.

**Key points to note about LinkedHashMaps in Java**

* A LinkedHashMap cannot contain duplicate keys.
* LinkedHashMap can have null values and the null key.
* The iteration order of the elements in a LinkedHashMap is predictable.
* LinkedHashMap is not thread-safe.

**3.TreeMap**

* Java TreeMap is a Red-Black tree based implementation of Java’s Map interface.
* The entries in a TreeMap are always sorted based on the natural ordering of the keys.

**Key points to note about TreeMap in Java**

* A TreeMap is always sorted based on keys. The sorting order follows the natural ordering of keys.
* A TreeMap cannot contain duplicate keys.
* TreeMap cannot contain the null key and can have null value.
* TreeMap is not synchronized.

**HashTable:**

* HashTable is used to store data based on Key/Value pairs.
* It also implements the Map interface
* A HashTable cannot contain null Key and null Value.
* HashTable is Synchornised.

