java.lang.Iterable: is one of the root interfaces of the Java collection classes. The Collection interface extends Iterable, so all subtypes of Collection also implement the Iterable interface.

```
public interface Iterable<T> {
  public Iterator<T> iterator();
}
```

java.util.Collection: is one of the root interfaces of the Java collection classes.
It is possible to generify the various Collection and Map types and subtypes in the Java collection API.

● List ● Set ● SortedSet ● NavigableSet ● Queue ● Deque	Iterable <- Collection <- List, Set <- SortedSet <- NavigationSet, Queue -> Deque

java.util.List: an interface, you need to instantiate a concrete implementation of the interface in order to use it.

- java.util.ArrayList
- java.util.LinkedList
- java.util.Vector
- java.util.Stack

java.util.Set: an interface, where each element can only exists once in a Set.

- java.util.EnumSet
- java.util.HashSet
- java.util.LinkedHashSet
- java.util.TreeSet.

java.util.SortedSet: subtype of the **java.util.Set** interface. It behaves like a normal set with the exception that the elements are sorted internally.

java.util.NavigableSet: subtype of the **java.util.SortedSet** interface. It behaves like a SortedSet with the exception you have navigation methods available in addition to the sorting mechanisms of the SortedSet.

java.util.Queue interface. A queue is designed to have elements inserted at the end of the queue, and elements removed from the beginning of the queue.

The java.util.Deque interface is a subtype of the **java.util.Queue** interface. It represents a queue where you can insert and remove elements from both ends of the queue. "Deck"

- java.util.ArrayDeque
- java.util.LinkedList

ArrayList	Vector	Stack (LIFO)	LinkedList
extends AbstractList which extends AbstractCollection implements		extends Vector implements	extends AbstractSequentialList which extends AbstractList implements
Serializable, Cloneable, Iterable<E>, Collection<>, List<E>, RandomAccess			+Left interfaces, +Dequ +Queue, -RandomAcess
add, addAll, get, indexOf,containsAll, equals, hashCode, isEmpty, iterator, listIterator, removeAll, retainAll, subList, set, remove(index), remove(obj), sort, splitIterator, subList, toArray, clear, clone, contains			
ensureCapacity	copyInto, ensureCapacity, firstElement, lastElement, insetElementAt, removeELement, removeElementAt , SetElementAt	empty, peek, pop, push, search	addFirst, addLast, descendingIterator, getFirst, getLast, offer - tail_elem, offerFirst, offerLast, peek, peekFir peekLast, poll, pollFirst, pollLast, pop, push, removeFirst, removeLas

EnumSet	HashSet	LinkedHashSet	TreeSet
	is backed by a HashMap. It makes no guarantees about the sequence of the elements when you iterate them.	differs from HashSet by guaranteeing that the order of the elements during iteration is the same as the order they were inserted into the LinkedHashSet.	also guarantees the order of the elements when iterated, but the order is the sorting (natural/comparator)order of the elements.
extends AbstractSet extends AbstractCollection		extends HashSet	Same as HashSet
implements Serializable, Cloneable, Iterable, Collection, Set			Plus NavigableSet, Sortedset
allOf, clone, complement Of, copyOf, noneOf, of, range	add, clear, clone, contains, isEmpty, iterator, remove, size		add, addAll, ceiling, clear, clone, comparator, contains, descIterato descSet, first, floor, headSet, higher, isEmpty, iterator, lower, pollFirst, pollLast, remove, size, splitIterator, subSet, tailSet

Priority Queue extends AbstractQueue implements Serializable, Iterable, Collection, Queue **Methods**: add, clear, comparator, contains, iterator, offer, peek, poll, remove, size, toArray	**equals()** is used in most collections to determine if a collection contains a given element. The **hashCode()** method of objects is used when you insert them into a HashTable, HashMap or HashSet. 1 . If equal, then same hash codes too. 2 . Same hash codes no guarantee of being equal.

Note: Information gathered in this document has been collected from various sources on the Internet.

github.com/uvkrishnasai