24. Collections in Java - Part 3 | Deque and List in Depth.

Deque is an interface, using which we can push elements both at the start and at the end of the queue.

Deque (I) $\xrightarrow{\text{implements}}$ ArrayDeque (C).

If the ArrayDeque is Full, then a new ArrayDeque is created, and all the contents are copied and saved and we use the new reference → O(N) operation

Not Thread-Safe version.	Thread-Safe version.
PriorityQueue	PriorityBlockingQueue
ArrayDeque	ConcurrentLinkedDeque
Deque [Not thread Safe]	

List

The List Iterator lets us have forward traversal,

1. hasNext() → forward traversal
2. hasPrevious() → backward traversal,

- replaceAll( ) ⇒ this method accepts functional interface and replaces all the value, with the value that we have provided,

list1.replaceAll((Integer val) → val+100);

Similarly Foreach

list.forEach((Integer val) → val-100);

list.forEach((Integer val) → Sout(val));

~~LinkedList~~

LinkedList	LinkedList
non-thread Safer Version.	Thread Safe Version.
LinkedList	CopyOnWriteArrayList

Vector → thread safe —extends→ Stack → thread safe

↳ every method inside it is synchronised, so it uses monitor Locks

Because its child of Vector.

Hence we don't use implementation of Deque.