Deque in Java

⇛

```
d.offerFirst(10);
d.offerLast(20);
d.pollFirst(10);
d.pollLast(20);
```

Other

```
d.peekFirst();
d.peekLast();
```

* → The above function don't throw exceptions, they return special values to indicate failure.

Eg:- offer return false when operation unsuccessful.

For exceptions

→ add instead of offer.
remove instead of poll.
element ——→ peek.

Traversing Deque

1) `Iterator it = d.iterator();`
`while (it.hasNext()) { System.out.print(it.next() + " "); }`

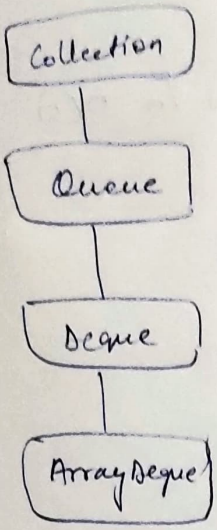
2) `for (int x : d) { System.out.print(x + " "); }`

Last to first

`Iterator it = d.descendingIterator();`

——— rest same ———

ArrayDeque in Java.



* LinkedList has both List & Queue func but ArrayDeque is Queue specific class.

⇒ ArrayDeque can be used as Deque, Queue, Stack & is faster.

⇒ ArrayDeque preferred instead of standard Stack (as) vector class is inherited by Standard Stack class
∴ causes extra overhead due to being thread safe.

⇒ Implemented using import java.util.ArrayDeque.

⇒ ArrayDeque<Integer> ad1 = new ArrayDeque<Integer>();
↳ Same line for stack push, pop & peek fns.

```
ad1.add(10);
ad1.add(20);

System.out.println(ad1);
```

⇒ (1) All fns present discussed in last page.
(2) All fns of Stack present too.
(3) All fns of Queue present too.
} All O(1).

⇒ * ArrayDeque inherits from Deque & Queue interface.
Stack is not an interface in java.

Data Structure with min/max operation (Deque Ques 1)

Q. Design a DS that supports the following operations in O(1) time complexity.

1. insertMin(x)
2. insertMax(x)
3. getMin()
4. getMax()

Sol. ArrayDeque.

⇒ Insert Min at front and Max at rear, then just we peek First/Last and poll First/Last.