Queue

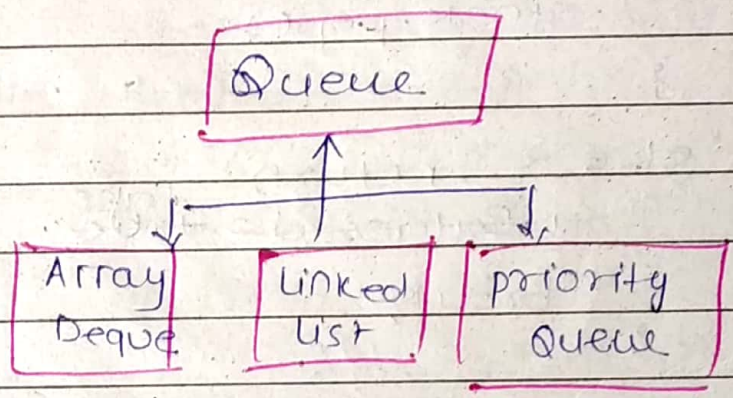
↓

works on FIFO (First in first out)

↓

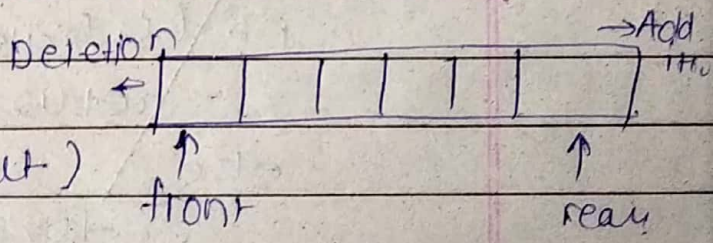
The Queue interface of the Java collections framework provides the functionality of the queue data structure.
It extends Collection framework.

In order to use the functionalities of Queue, we need to use classes that implement it.



working of Queue data Structure

FIFO
(First in First out)

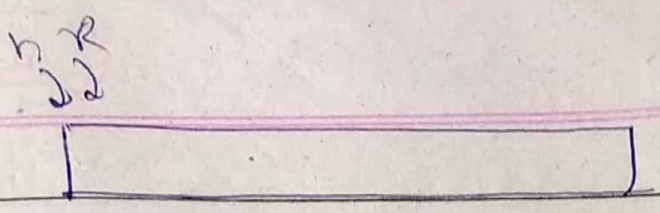


Methods of Queue

(throw Exception)	(return false/null)
1. add()	4. offer()
2. remove()	5. poll()
3. element()	6. peek()

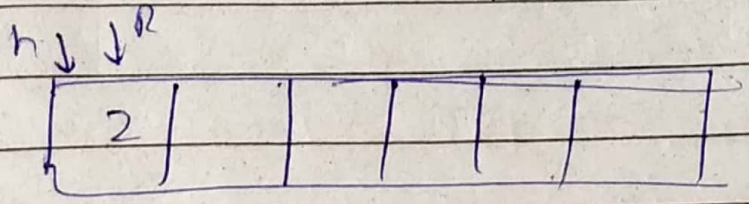
↓

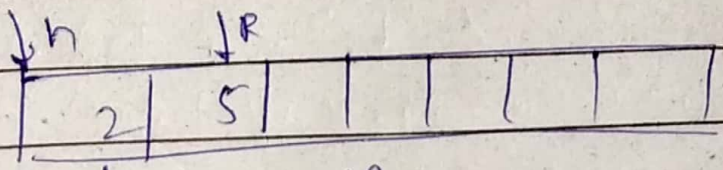
(Top element dekhne ke kaam aata hai)

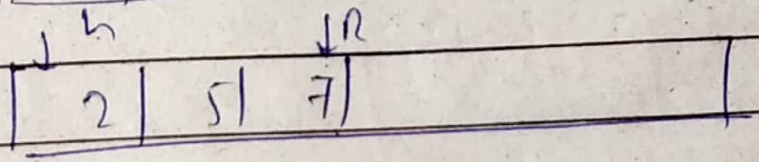


pop() → h → dequeue()
~~→ enqueue()~~

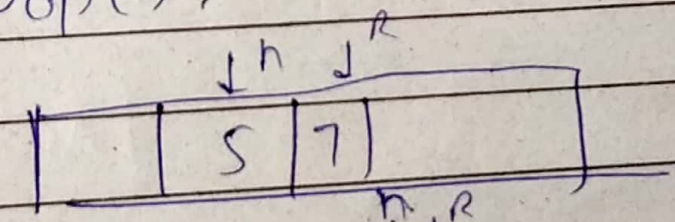
push() → R → ~~dequeue~~
↳ enqueue().

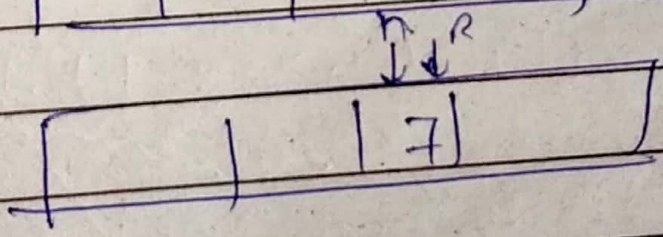






pop();

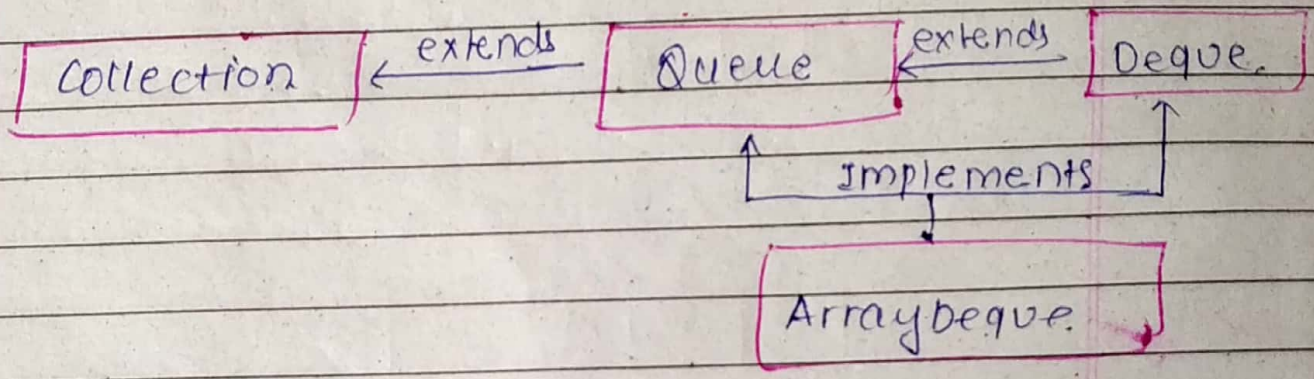




Array Deque.

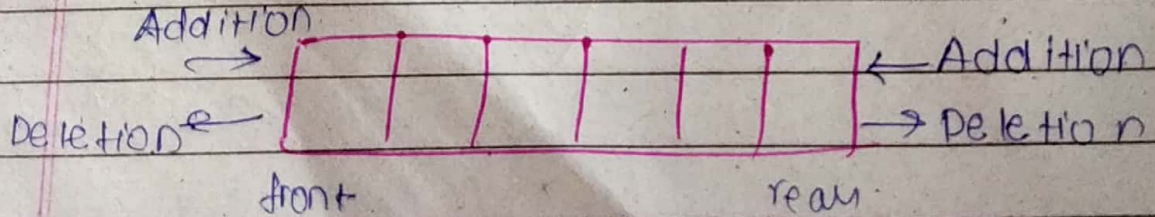
An Array Deque (also known as an "Array Double Ended Queue", pronounced as "Array Deck") is a special kind of a growable array that allows us to add or remove an element from both sides.

An Array Deque implementation can be used as a stack (last-in-first-out) or a Queue (first-in-first-out).



Deque

In a regular queue, elements are added from the rear and removed from the front. However, in a deque, we can insert & remove elements from both front & rear.



** ArrayDeque are much faster than LinkedList.

return null / false

Operation	Method	Method throwing Exception
1. Insertion from head.	offerFirst(e)	addFirst(e)
2. Removal from head	pollFirst()	removeFirst()
3. Retrieval from head	peekFirst()	getFirst()
4. Insertion from tail	offerLast(e)	addLast(e)
5. Removal from Tail	pollLast()	removeLast()
6. Retrieval from Tail	peekLast()	getLast()

Implementation of ArrayDeque

```
package deque;
import java.util.ArrayDeque;
public class mainclass {
    public static void main (String [] args) {

        ArrayDeque<Integer> ad = new ArrayDeque<>();

        ad.addFirst(12);
        ad.addFirst(23);

        ad.pop();
        ad.peek();
        System.out.println(ad.peek());
    }
}
```

⇒ Output : 12