Java Deque Interface

- The interface called Deque is present in java.util package.
- It is the subtype of the interface queue.
- The Deque supports the addition as well as the removal of elements from both ends of the data structure. Therefore, a deque can be used as a stack or a queue.
- We know that the stack supports the Last In First Out (LIFO) operation, and the operation First In First Out (FIFO)is supported by a queue.
- As a deque supports both, either of the mentioned operations can be performed on it. Deque is an acronym for "double ended queue".

ArrayDeque class

- We know that it is not possible to create an object of an interface in Java. Therefore, for instantiation, we need a class that implements the Deque interface, and that class is ArrayDeque.
- It grows and shrinks as per usage. It also inherits the AbstractCollection class.

The important points about ArrayDeque class are:

- Unlike Queue, we can add or remove elements from both sides.
- Null elements are not allowed in the ArrayDeque.
- ArrayDeque is not thread safe, in the absence of external synchronization.
- ArrayDeque has no capacity restrictions.

Methods of ArrayDeque

```
import java.util.ArrayDeque;

class Deque1
{
   public static void main(String[] args)
   {
      ArrayDeque<String> ad1 = new ArrayDeque<>();

      //(1)//
      /*public boolean add(E e)
      {
         addLast(e);
         return true;
      }*/

      //(2)//
      /* public boolean offer(E e)
      {
         return offerLast(e);
      } */

      //(3)//
```

```
/* public boolean offerLast(E e)
{
   addLast(e);
   return true;
} */

//(4)//
/*public void addLast(E e) */


//All above methods will Inserts the specified element at the end of this deque.
//Offer() will return false if it fails to insert the element on a size
//restricted Queue,Where as add() will throw an IllegalStateException

ad1.add("One");
ad1.add("Two");
ad1.add("Three");
System.out.println(ad1);//[One, Two, Three]

ad1.add("Four");
System.out.println(ad1);//[One, Two, Three, Four]

ad1.offer("Five");
System.out.println(ad1);//[One, Two, Three, Four, Five]

ad1.offerLast("Six");
System.out.println(ad1);//[One, Two, Three, Four, Five, Six]

ad1.addLast("Seven");
System.out.println(ad1);//[One, Two, Three, Four, Five, Six, Seven]

/*_____________________________________________________________________*/
//5//
/* public boolean offerFirst(E e)
{
   addFirst(e);
   return true;
} */

//(6)//
/*public void addFirst(E e) */

//both methods will Inserts the specified element at the front of this deque.

ad1.offerFirst("eight");//[eight, One, Two, Three, Four, Five, Six, Seven]
System.out.println(ad1);

ad1.addFirst("nine");
System.out.println(ad1);//[nine, eight, One, Two, Three, Four, Five, Six, Seven]
```

```
/*_____________________________________________________________________________*/
    //(6)//
    /* public E peek()
    {
       return peekFirst();
    } */

    //(7)//
    /*public E peekFirst()*/

    //Both methods returs first element of Deque, but do not remove that element

    System.out.println(ad1.peek());//nine
    System.out.println(ad1);//[nine, eight, One, Two, Three, Four, Five, Six, Seven]

    System.out.println(ad1.peekFirst());//nine
    System.out.println(ad1);//[nine, eight, One, Two, Three, Four, Five, Six, Seven]

    //(8)//
    /* public E peekLast() */
    //returs last element of Deque, but do not remove that element

    System.out.println(ad1.peekLast());//Seven
    System.out.println(ad1);//[nine, eight, One, Two, Three, Four, Five, Six, Seven]
/*_____________________________________________________________________________*/

    //(9)//
    /*  public E remove()
    {
       return removeFirst();
    } */

    //(10)//
    /*  public E removeFirst()
    {
       E e = pollFirst();
       if (e == null)
          throw new NoSuchElementException();
       return e;
    } */
    //both methods removes first element of Deque and returns that element

    System.out.println(ad1.remove());//nine
    System.out.println(ad1);//[eight, One, Two, Three, Four, Five, Six, Seven]

    System.out.println(ad1.removeFirst());//eight
    System.out.println(ad1);//[One, Two, Three, Four, Five, Six, Seven]
```

```
//(11)//
/*  public boolean remove(Object o)
{
   return removeFirstOccurrence(o);
} */

//removes first occurance entered element and returns boolean

System.out.println(ad1.remove("Three"));//true
System.out.println(ad1);//[One, Two, Four, Five, Six, Seven]

//(12)//
/*public E removeLast()
{
   E e = pollLast();
   if (e == null)
      throw new NoSuchElementException();
   return e;
} */

// removes last element of Deque and returns that element

System.out.println(ad1.removeLast());//Seven
System.out.println(ad1);//[One, Two, Four, Five, Six]
/*_____________________________________________________________________________*/

//(13)//
/*public E poll()
{
   return pollFirst();
} */

//(14)//
/* public E pollFirst() */

//both methods removes first element of Deque and returns that element

System.out.println(ad1.poll());//One
System.out.println(ad1);//[Two, Four, Five, Six]

System.out.println(ad1.pollFirst());//Two
System.out.println(ad1);//[Four, Five, Six]


//(15)//
/* public E pollLast() */
```

```
        //removes last element of Deque and returns that element

        System.out.println(ad1.pollLast());//Six
        System.out.println(ad1);//[Four, Five]
/*_____________________________________________________________________________ */

        //(16)//
        /*public int size()*/

        System.out.println(ad1.size());//2
/*_____________________________________________________________________________*/

        //(17)//
        /* public E getFirst() */

        //(18)//
        /* public E getLast() */

        //above methods returns first and last elements of queue
          //System.out.println(ad1.getFirst()); //If no element available in Deque then R.E: NoSuchElementException
          //System.out.println(ad1.getLast()); //If no element available in Deque then R.E: NoSuchElementException

        System.out.println(ad1.getFirst());//Four
        System.out.println(ad1.getLast());//Five
        System.out.println(ad1);//[Four, Five]
/*_____________________________________________________________________________ */

        //(19)//
        /* public boolean contains(Object o) */

        //returns true if this deque contains the specified element else false

        System.out.println(ad1.contains("Four"));//true
        System.out.println(ad1.contains("six"));//false
/*_____________________________________________________________________________*/

        //(20)//
        /* public void clear() */
        // Removes all of the elements from this deque.

        ad1.clear();
        System.out.println(ad1);//[]

    }
}
```