NIELIT Gorakhpur

Course Name: A Level (2nd Sem) **Subject: JAVA**
Topic: Queue interface in Java **Date: 02-06-20**

Queue interface:

A **Queue** is designed in such a way so that the elements added to it are placed at the end of Queue and removed from the beginning of Queue. Java Queue interface orders the element in FIFO(First In First Out) manner. The most common classes that implement Queue interface in Java are:

- PriorityQueue
- LinkedList

PriorityQueue class:

The PriorityQueue class provides the facility of using queue. But it does not orders the elements in FIFO manner. It inherits AbstractQueue class. In a priority queue, added objects are according to their priority. By default, the priority is determined by objects' natural ordering. It does not allow NULL objects. PriorityQueue is **not thread safe**.

```
import java.util.*;
class PQ{
public static void main(String args[]){
PriorityQueue<String> queue=new PriorityQueue<String>();
queue.add("Nolan");
queue.add("Burton");
queue.add("Camroon");
queue.add("Bay");
queue.add("Tarantino");
System.out.println("head:"+queue.element());
System.out.println("head:"+queue.peek());
System.out.println("iterating the queue elements:");
Iterator itr=queue.iterator();
while(itr.hasNext()){
System.out.println(itr.next());
}
queue.remove();
queue.poll();
System.out.println("after removing two elements:");
Iterator<String> itr2=queue.iterator();
while(itr2.hasNext()){
System.out.println(itr2.next());
}
} }
```

Java Deque Interface:

Deque is a Queue in which you can add and remove elements from both sides i.e it supports element insertion and removal at both ends. The name deque is short for "double ended queue" and is usually pronounced "deck". Array deques stops the use of Null elements. Any concurrent access by multiple threads is not supported.

ArrayDeque class:

The ArrayDeque class provides the facility of using deque and resizable-array. It inherits AbstractCollection class and implements the Deque interface. Null elements are not allowed. we can add or remove elements from both sides.

```
import java.util.*;
public class AD {
public static void main(String[] args) {
                    Deque<String> dq = new ArrayDeque<String>();
            dq.add("Stark");
            dq.add("Dent");
            dq.addLast("Wayne");
            dq.addFirst("Parker");
            dq.add("Kent");

            System.out.println("Elements in Deque:"+dq);


System.out.println("Removed element: "+dq.removeLast());
```

Removing last element

```
System.out.println("Head: "+dq.element());
```

returns the head of the Deque. Head is the first element of Deque

```
System.out.println("poll(): "+dq.pollLast());
```

removes and returns the tail of the Deque(last element).

```
System.out.println("peek(): "+dq.peek());
```

**returns null if the Deque is empty.
Same as element method**

```
System.out.println("Elements in Deque:"+dq);
        }
      }
```