

Queue

Queue is an interface which is implemented by two classes

1. LinkedList
2. Priority Queue

Queue follows FIFO (First In First Out) Data Structure.

1. LinkedList :-

LinkedList is a class which implements List as well as Queue Interface.

class LinkedList implements List, Queue

Queue q1=new LinkedList();

Here q1 will exhibits behaviour of queue.

Note :-

All features are same, only difference is LinkedList implementing Queue interface.

Objects will not be sorted rather insertion order is maintained.

Methods

Offer(Object) :- Used to add the element to the queue

Poll() :- Used to remove the element from the queue.

Peek() :- Used to fetch the object which is ready to get removed.

```
ex:- public class ObjClass {
    public static void main(String[] args) {
        Queue hs=new LinkedList<>();
        hs.offer(102);
        hs.offer(122);
        hs.add(21);
        hs.offer(142);
        System.out.println(hs); //[102, 122, 21, 142]
        System.out.println(hs.peek()); //102
        while(!hs.isEmpty()) {
            System.out.println(hs.poll());
        }
        System.out.println(hs); //[]
    }
}
```

2. PriorityQueue :-

Introduced in 1.2 version.

Default capacity is 11.

It allows only homogeneous objects.

Basically data structure of Queue is FIFO but in PriorityQueue internal sorting will happen.

Duplicate Objects are allowed.

null objects are not allowed.

It removes the elements in ascending order.

ex :-

```
public class CollFrWork {
    public static void main(String[] args) {
        Queue hs=new PriorityQueue<>();
        hs.offer(102);
        hs.offer(122);
        hs.add(21);
        hs.add(21);
        System.out.println(hs); //[21, 21, 102, 122]
        System.out.println(hs.peek()); //21
        while(!hs.isEmpty()) {
            System.out.println(hs.poll()); //[21, 21, 102, 122]
        }
        System.out.println(hs); //[]
    }
}
```

ex2 :- printing values in descending order using Comparator

```
public class CollFrWork implements Comparator {
    @Override
    public int compare(Object o1, Object o2) {
        Integer in1 = (Integer)o1;
        Integer in2 = (Integer)o2;
        if(in1<in2) {
            return 1;
        }
        else if(in1>in2) {
            return -1;
        }
        return 0;
    }
    public static void main(String[] args) {
        Queue hs=new PriorityQueue<>(new CollFrWork());
        hs.offer(102);
        hs.offer(122);
        hs.add(21);
        hs.add(21);
        System.out.println(hs); //[122, 102, 21, 21]
        System.out.println(hs.peek()); //122
        while(!hs.isEmpty()) {
            System.out.println(hs.poll()); //[122, 102, 21, 21]
        }
        System.out.println(hs); //[]
    }
}
```