OBJECT ORIENTED PROGRAMMING LAB

Name: Justin v kalappura

Roll No: 10

Batch: S2 MCA B

Date: 31-05-2022

Experiment No.: 27

Aim:

Program to demonstrate the creation of queue object using the Priority Queue class.

PROCEDURE:

```
import java.util.*;
public class Collection_Framework_Queue {
    public static void main(String args[]) {
        Queue<Integer> q = new PriorityQueue<Integer>(new Comp());
        int ch;
        Scanner sc = new Scanner(System.in);
        do {
            System.out.println("\n1.ADD\n2.PEEK\n3.POLL or
REMOVE\n4.DISPLAY\n5.EXIT");
            System.out.println("Enter your choice : ");
            ch = sc.nextInt();
            switch (ch) {
                case 1:
                    System.out.println("\n\tEnter Integer : ");
                    int n1 = sc.nextInt();
                    q.add(n1);
                    System.out.println("\n\tADDED SUCCESSFULLY ! ! ! ");
                    break;
                case 2:
                    if (q.isEmpty()) {
                        System.out.print("\n\tQueue Empty ! ! !");
```

```
            } else {
                System.out.print("\n\tPeeked element is " + q.peek());
            }
            break;
        case 3:
            if (!q.isEmpty()) {
                System.out.print("\n\tRemoved element is " + q.poll());
            } else {
                System.out.print("\n\tQueue Empty ! ! !");
            }
            break;
        case 4:
            if (!q.isEmpty()) {
                System.out.print("\nSize of queue : " + q.size());
                System.out.print("\nQueue elements  : " + q);
                System.out.println("\nQueue elements are");
                for (int i : q) {
                    System.out.println(i);
                }
            } else {
                System.out.print("\n\tQueue Empty ! ! !");
            }
            break;
        case 5:
            break;
        default:
            System.out.println("\n\tPlease enter valid choice ! ! ! ");
      }
    } while (ch != 5);
```

```
    }
}
class Comp implements Comparator<Integer> {
    public int compare(Integer a, Integer b) {
        return a % 10 > b % 10 ? 1 : -1;
    }
}
```

Output Screenshot:

```
D:\jomol>javac  Collection_Framework_Queue.java

D:\jomol>java Collection_Framework_Queue

1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
1

        Enter Integer :
2

        ADDED SUCCESSFULLY ! ! !

1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
1

        Enter Integer :
4

        ADDED SUCCESSFULLY ! ! !
```

```
1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
1

        Enter Integer :
4

        ADDED SUCCESSFULLY ! ! !

1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
4

Size of queue : 2
Queue elements  : [2, 4]
Queue elements are
2
4

1.ADD
2.PEEK
3.POLL or REMOVE
```

```
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
4

Size of queue : 2
Queue elements  : [2, 4]
Queue elements are
2
4

1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
2

        Peeked element is 2
1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
3

        Removed element is 2
```

```
1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
3

        Removed element is 2
1.ADD
2.PEEK
3.POLL or REMOVE
4.DISPLAY
5.EXIT
Enter your choice :
5

D:\jomol>
```