|  |  |
| --- | --- |
| ***Roll No*** | ***22SW040*** |
| ***Section*** | ***01*** |
| ***Name*** | ***Farooque Sajjad*** |
| ***LAB No*** | ***08*** |

***Code***

***package DSA\_LAB\_08;***

***public interface Queue{***

***int size();***

***Object remove();***

***Object first();***

***void add(Object e);***

***}***

***/////////////////////////////////////////////////////////***

***package DSA\_LAB\_08;***

***public class LinkListWithQueue implements Queue{***

***private static class Node {***

***Object data;***

***Node next = this;***

***Node prev = this;***

***public Node(Object data, Node next, Node prev) {***

***this.data = data;***

***this.next = next;***

***this.prev = prev;***

***}***

***public Node(Object data) {***

***this.data = data;***

***}***

***}***

***private Node head = new Node(null);***

***private int size;***

***@Override***

***public void add(Object e) {***

***head.prev.next = new Node(e,head.prev,head);***

***head.prev = head.prev.next;***

***++size;***

***}***

***@Override***

***public Object first() {***

***if(size==0) throw new IllegalStateException("Queue is Empty");***

***else return head.next.data;***

***}***

***@Override***

***public Object remove() {***

***if(size==0) throw new IllegalStateException("Queue is Empty");***

***Object temp = head.next.data;***

***head.next = head.next.next;***

***head.next.prev = head;***

***--size;***

***return temp;***

***}***

***@Override***

***public int size() {***

***return size;***

***}***

***public void displayQueue(LinkListWithQueue head1) {***

***LinkListWithQueue copy = new LinkListWithQueue();***

***if (head1.size() == 0) {***

***throw new IllegalStateException("Queue is Empty");***

***}***

***System.out.print("Displaying Queue: ");***

***int s = head1.size();***

***for (int i = 1; i <= s; i++) {***

***Object element = head1.first();***

***copy.add(element);***

***System.out.print(element + " ");***

***head1.remove();***

***}***

***for (int i = 0; i < s; i++) {***

***head1.add(copy.first());***

***copy.remove();***

***}***

***System.out.println();***

***}***

***public static void main(String[] args) {***

***LinkListWithQueue t1 = new LinkListWithQueue();***

***t1.add("Task : 01 ");***

***t1.add("Task : 02 ");***

***t1.add("Task : 03 ");***

***t1.add("Task : 04 ");***

***t1.add("Task : 05 ");***

***t1.add("Task : 06 ");***

***t1.add("Task : 07 ");***

***t1.add("Task : 08 ");***

***t1.add("Task : 09 ");***

***t1.add("Task : 10 ");***

***t1.add("Task : 11 ");***

***System.out.println(t1.first());***

***t1.remove();***

***System.out.println(t1.first());***

***t1.displayQueue(t1);***

***System.out.println("Number Of Task : " + t1.size());***

***System.out.println("First Task : " + t1.first());***

***t1.remove();***

***System.out.println("First Task after Removing : " + t1.first());***

***t1.displayQueue(t1);***

***}***

***}***

***/////////////////////////////////////////////////////////***

***package DSA\_LAB\_08;***

***public class ArrayImplementationOfQueue implements Queue {***

***int head, tail, size, capacity;***

***Object[] queue;***

***public void resizeQueue() {***

***Object[] newQueue = queue;***

***queue = new Object[2 \* queue.length];***

***System.arraycopy(newQueue, 0, queue, 0, size);***

***head = 0;***

***tail = size - 1;***

***capacity \*= 2;***

***}***

***public ArrayImplementationOfQueue(int capacity) {***

***queue = new Object[capacity];***

***this.capacity = capacity;***

***this.head = this.size = 0;***

***this.tail = -1;***

***}***

***public boolean isEmpty() {***

***return size == 0;***

***}***

***public boolean isFull() {***

***return size == capacity;***

***}***

***@Override***

***public int size() {***

***return size;***

***}***

***@Override***

***public void add(Object e) {***

***if (isFull()) {***

***resizeQueue();***

***}***

***tail = (tail + 1) % capacity;***

***queue[tail] = e;***

***size++;***

***}***

***@Override***

***public Object first() {***

***if (isEmpty()) {***

***throw new IllegalStateException("Queue is Empty cannot push");***

***}***

***return queue[head];***

***}***

***@Override***

***public Object remove() {***

***if (isEmpty()) {***

***throw new IllegalStateException("Queue is Empty cannot push");***

***}***

***Object temp = queue[head];***

***head = (head + 1) % capacity;***

***size--;***

***return temp;***

***}***

***public void displayQueue(ArrayImplementationOfQueue queue1) {***

***for (int i = 0; i < queue1.size(); i++) {***

***System.out.print(queue1.queue[(queue1.head + i) % queue1.capacity] + " ");***

***}***

***System.out.println();***

***}***

***public static void main(String[] args) {***

***ArrayImplementationOfQueue t1= new ArrayImplementationOfQueue(5);***

***t1.add("Task : 01 ");***

***t1.add("Task : 02 ");***

***t1.add("Task : 03 ");***

***t1.add("Task : 04 ");***

***t1.add("Task : 05 ");***

***t1.add("Task : 06 ");***

***t1.add("Task : 07 ");***

***t1.add("Task : 08 ");***

***t1.add("Task : 09 ");***

***t1.add("Task : 10 ");***

***t1.add("Task : 11 ");***

***System.out.println(t1.first());***

***t1.remove();***

***System.out.println(t1.first());***

***t1.displayQueue(t1);***

***System.out.println("Number Of Task : " + t1.size());***

***System.out.println("First Task : " + t1.first());***

***t1.remove();***

***System.out.println("First Task after Removing : " + t1.first());***

***t1.displayQueue(t1);***

***}***

***}***

***Screen Shots***



