package queue.to.linkedlist;

public class QueueToLinkedList {

Node front;

Node rear;

public QueueToLinkedList ()

{

front = null;

rear = null;

}

private class Node{

int i;

Node next;

Node(int i){

this.i = i;

}

public void displayData(){

System.out.println("i= " + i);

}

}

public void insertLast(int i){

Node newNode = new Node(i);

if(isEmpty()){

front = newNode;

}else{

rear.next = newNode;

}

rear = newNode;

}

public int removeFirst(){

int temp = front.i;

if(front.next == null){

rear = null;

}

front = front.next;

return temp;

}

public void displayList(){

Node current = front;

while(current != null){

current.displayData();

current = current.next;

}

}

public int nodeData(){

return front.i;

}

public boolean isEmpty(){

return front == null;

}

public void insert(int item){

insertLast(item);

}

public int remove(){

if(isEmpty()){

throw new RuntimeException("Queue is empty..");

}

return removeFirst();

}

public int peek(){

if(isEmpty()){

throw new RuntimeException("Queue is empty..");

}

return nodeData();

}

public static void main(String[] args) {

QueueToLinkedList queue = new QueueToLinkedList ();

queue.insert(3);

queue.insert(6);

System.out.println("-- Displaying Queue data--");

queue.displayList();

System.out.println("Item peeked- " + queue.peek());

System.out.println("-- Removing Queue elements--");

System.out.println("Item removed- " + queue.remove());

System.out.println("Item removed- " + queue.remove());

}

}