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());

}

}