|  |
| --- |
| package datastructure; |
|  |  |
|  |  |
|  |  |
|  | /\*\* |
|  | \* |
|  | \* @author alhajjaji |
|  | \*/ |
|  | public class Lap5<T>{ |
|  | private Node<T> head=null; |
|  | private Node<T> tail=null; |
|  | private int size=0; |
|  |  |
|  | public Lap5() { |
|  | } |
|  | public boolean isEmpty() |
|  | { |
|  | return size==0; |
|  | } |
|  |  |
|  | public int size() |
|  | { |
|  | return size; |
|  | } |
|  |  |
|  | public T frist() |
|  | { |
|  | if(isEmpty()) return null; |
|  | return head.getElement(); |
|  | } |
|  | public T last() |
|  | { |
|  | if(isEmpty()) return null; |
|  | return tail.getElement(); |
|  | } |
|  |  |
|  | public void addFrist(T e) |
|  | { |
|  | head =new Node<T>(e,head); |
|  | if(isEmpty()) |
|  | tail=head; |
|  | size++; |
|  | } |
|  | public void addLast(T e) |
|  | { |
|  | Node<T> newst=new Node<>(e,null); |
|  | if(isEmpty()) |
|  | head=newst; |
|  | else |
|  | tail.setNext(newst); |
|  | tail=newst; |
|  | size++; |
|  | } |
|  | public T removeFrist() |
|  | { |
|  | if(isEmpty()) return null; |
|  | T d=head.getElement(); |
|  | head=head.getNext(); |
|  | if(isEmpty()) |
|  | tail=null; |
|  | size--; |
|  | return d; |
|  | } |
|  | public void rotate() |
|  | { |
|  | if(size>2) |
|  | addLast(removeFrist()); |
|  | } |
|  |  |
|  | public void reverse() |
|  | { |
|  | if(size>2){ |
|  | for(int i=0;i<size;i++){ |
|  | addLast(removeFrist());}} |
|  | } |
|  | public void print() |
|  | { |
|  | Node<T> i=head; |
|  | while(i!=null) |
|  | { |
|  | System.out.println(i.getElement().toString()); |
|  | i=i.getNext(); |
|  | } |
|  | } |
|  | private class Node<T>{ |
|  | private T element; |
|  | private Node<T> next; |
|  |  |
|  | public Node(T element, Node<T> next) { |
|  | this.element = element; |
|  | this.next = next; |
|  | } |
|  |  |
|  | public T getElement() { |
|  | return element; |
|  | } |
|  |  |
|  | public void setElement(T element) { |
|  | this.element = element; |
|  | } |
|  |  |
|  | public Node<T> getNext() { |
|  | return next; |
|  | } |
|  |  |
|  | public void setNext(Node<T> next) { |
|  | this.next = next; |
|  | } |
|  |  |
|  |  |
|  | } |
|  |  |
|  |  |
|  | public static void main(String[] args) |
|  | { |
|  | //merge two lists |
|  | Lap5<Integer> list=new Lap5<>(); |
|  | list.addLast(2); |
|  | list.addLast(3); |
|  | list.print(); |
|  | Lap5<Integer> list2=new Lap5<>(); |
|  | list2.addLast(4); |
|  | list2.addLast(5); |
|  | list2.print(); |
|  |  |
|  |  |
|  | } |
|  |  |
|  | } |