DoublyLinkedList

class Node{

Node pre , next;

int empNo;

String name;

float salary;

Node(int empNo,float salary,String name)

{

this.empNo=empNo;

this.salary=salary;

this.name=name;

pre=next=null;

}

}

class DoublyLinkedList

{

Node head=null;

Node tail=null;

public void insert(int empNo,float salary,String name)

{

Node nn = new Node(empNo,salary,name);

if(head==null)

{

head=nn;

tail=nn;

System.***out***.println("value inserted");

return;

}

Node temp=head;

while(temp.next!=null)

{

temp=temp.next;

}

temp.next=nn;

nn.pre=temp;

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

}

public void traverse()

{

Node temp = head;

if(head==null)

{

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

return;

}

while(temp!=null)

{

System.***out***.println("("+temp.empNo+" "+temp.salary+" "+temp.name+")-->");

temp=temp.next;

}

}

public void insertAtPos(int empNo,float salary,String name,int pos)

{

Node nn = new Node(empNo, salary, name);

if (pos == 1) {

nn.next = head;

if (head != null) {

head.pre = nn;

}

head = nn;

if (tail == null) {

tail = nn;

}

return;

}

Node temp = head;

for (int i = 1; i < pos - 1 && temp != null; i++) {

temp = temp.next;

}

if (temp == null || temp.next == null) {

System.***out***.println("Position out of bounds");

return;

}

nn.next = temp.next;

if (temp.next != null) {

temp.next.pre = nn;

}

temp.next = nn;

nn.pre = temp;

}

public void deleteAtbeg()

{

if(head==null)

{System.***out***.println("empty");}

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

head=head.next;

head.pre=null;

}

public void deleteAtEnd()

{

if(head==null)

{

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

return;

}

Node temp=head;

while(temp.next!=null)

{

temp=temp.next;

}

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

temp.pre.next=null;

}

public void deleteAtSpecfic(int empNo)

{

Node temp = head;

while(temp!=null)

{

if(temp.empNo==empNo)

{

if(temp==head)

{

head=temp.next;

}

else if(temp.pre.next == temp.next)

{

temp.pre.next=temp.next;

}

}

temp=temp.next;

}

System.***out***.println("element not found...");

}

public void deleteAtPos(int pos)

{

if(pos<=0 && head==null)

{

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

}

Node temp = head;

for(int i=1; i<pos && temp!=null ; i++ )

{

temp=temp.next;

if(temp==null)

{

System.***out***.println("out of bound");

return;

}

if(temp==head) {

deleteAtbeg();

}

else if(temp.next==null)

{

deleteAtEnd();

}

else {

temp.pre.next=temp.next;

temp.next.pre=temp.pre;

}

}

}

public boolean search(int empNo)

{

Node temp=head;

while(temp!=null)

{

if(temp.empNo==empNo)

{

return true;

}

temp=temp.next;

}

return false;

}

public void update(int old,int Nval)

{

Node temp = head;

while(temp!=null)

{

if(temp.empNo==old)

{

temp.empNo=Nval;

}

temp=temp.next;

}

System.***out***.println("element not found");

}

public void searchFromEnd(int empNo)

{

Node temp = head;

if(temp==null)

{System.***out***.println("empty");

return;

}

while(temp.next!=null)

{

temp=temp.next;

}

while(temp!=null)

{

int pos=1;

if(temp.empNo==empNo)

{

System.***out***.println("founds at position :"+pos+" "+temp.salary+" "+temp.name);

}

temp=temp.pre;

pos++;

}

System.***out***.println("not found");

}

public void sort()

{

if(head==null && head.next==null)

{

System.***out***.println("no sorting");

}

Node i,j;

for(i=head;i.next!=null;i=i.next)

{

for(j=i.next;j!=null;j=j.next)

{

if(j.salary<i.salary)

{

int tempEmp=i.empNo;

float tempSal = i.salary;

String tempName = i.name;

i.empNo=j.empNo;

i.salary=j.salary;

i.name=j.name;

j.empNo=tempEmp;

j.salary=tempSal;

j.name=tempName;

}

}

}

}

public void minMax()

{

if(head==null)

{System.***out***.println("empty");}

float min=head.salary;

float max = head.salary;

Node temp = head;

while(temp!=null)

{

if(temp.salary<min)

{

min=temp.salary;

}

if(temp.salary>max)

{

max=temp.salary;

}

}

}

public float salarySum()

{

float sum=0;

Node temp = head;

while(temp!=null)

{

sum+=temp.salary;

temp=temp.next;

}

return sum;

}

public void reverse()

{

Node temp = null ;

Node current = head;

while(current!=null)

{

temp = current.pre;

current.pre=current.next;

current.next=current;

current=current.pre;

}

if(temp!=null)

{

head=temp.pre;

}

}

}

public class task1 {

public static void main(String args[])

{

DoublyLinkedList dd = new DoublyLinkedList();

dd.insert(01,1300,"hari");

dd.insert(02,1400,"bob");

dd.insert(03,1500,"haran");

dd.traverse();

dd.insertAtPos(4,1244,"hari",3);

dd.traverse();

dd.deleteAtbeg();

dd.traverse();

dd.deleteAtEnd();

dd.traverse();

System.***out***.println(dd.search(3));

dd.update(4, 40);

dd.sort();

dd.minMax();

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

}

}

Circular linked list :::

package day16;

class Node1{

Node1 next;

int data;

Node1(int data)

{

this.data=data;

next=null;

}

}

class CircularList{

Node1 head=null;

Node1 tail = null;

public void insertAtBeg(int data)

{

Node1 nn = new Node1(data);

if(head==null)

{

head=nn;

tail=nn;

tail.next=head;

}

nn.next=head;

head=nn;

tail.next=head;

}

public void insertAtEnd(int data)

{

Node1 nn = new Node1(data);

if(head==null)

{

head=nn;

tail=nn;

tail.next=head;

}

else {

tail.next=nn;

nn.next=head;

}

}

public void traverse() {

if(head==null)

{

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

}

Node1 temp = head;

do {

System.***out***.println(temp.data+"->");

temp=temp.next;

}while(temp!=head);

}

public void delete(int data)

{

Node1 current = head;

Node1 pre = null;

if(head==null)

{System.***out***.println("empty");

return;

}

if(head.data==data)

{

if(head==tail)

{

head=tail=null;

}

else {

head=head.next;

tail.next=tail;

}

return ;

}

do {

pre = current;

current=current.next;

if(current.data==data)

{

pre.next=current.next;

}

if(current==tail)

{

tail=pre;

pre.next=head;

}

}while(current!=tail);

}

}

public class task2 {

public static void main(String args[])

{

CircularList l = new CircularList();

l.insertAtBeg(18);

l.insertAtBeg(20);

l.traverse();

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

l.insertAtEnd(50);

l.traverse();

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

l.delete(50);

l.traverse();

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

}

}

STACK USING LINKED LIST :

package day16;

class Node1 {

int data;

Node1 next;

public Node1(int data) {

this.data = data;

this.next = null;

}

}

class StackLinkedList {

private Node1 top;

public StackLinkedList() {

this.top = null;

}

public void push(int data) {

Node1 newNode = new Node1(data);

newNode.next = top;

top = newNode;

System.***out***.println(data + " pushed to stack");

}

public int pop() {

if (isEmpty()) {

System.***out***.println("Stack Underflow! No elements to pop.");

return -1;

}

int poppedData = top.data;

top = top.next;

return poppedData;

}

public int peek() {

if (isEmpty()) {

System.***out***.println("Stack is empty!");

return -1;

}

return top.data;

}

public boolean isEmpty() {

return top == null;

}

public void display() {

if (isEmpty()) {

System.***out***.println("Stack is empty!");

return;

}

Node1 temp = top;

System.***out***.print("Stack elements: ");

while (temp != null) {

System.***out***.print(temp.data + " -> ");

temp = temp.next;

}

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

}

}

public class task3 {

public static void main(String[] args) {

StackLinkedList stack = new StackLinkedList();

stack.push(10);

stack.push(20);

stack.push(30);

stack.display();

System.***out***.println("Top element is: " + stack.peek());

System.***out***.println("Popped element: " + stack.pop());

stack.display();

}

}

QUEUE USING LINKED LIST :

package day16;

class Node3 {

int data;

Node3 next;

public Node3(int data)

{

this.data=data;

next=null;

}

}

class Queue

{

Node3 rear,front = null;

public void enqueue(int data)

{

Node3 nn = new Node3(data);

if(rear==null && front==null)

{

rear=nn;

front=nn;

System.***out***.println("inserted :"+data);

}

else {

rear.next=nn;

rear=nn;

System.***out***.println("inserted:"+data);

}

}

public void dequeue()

{

int dd = front.data;

if(front==null)

{

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

}

front=front.next;

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

}

public boolean isEmpty()

{

return front==null;

}

public int peek()

{

if(rear!=null)

{

return rear.data;

}

return -1;

}

public void display()

{

if(front==null)

{

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

}

Node3 temp = front;

while(temp!=null)

{

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

temp=temp.next;

}

}

}

public class task4{

public static void main(String args[])

{

Queue q = new Queue();

q.enqueue(20);

q.enqueue(40);

q.enqueue(60);

q.enqueue(760);

q.enqueue(860);

q.enqueue(80);

q.dequeue();

q.dequeue();

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

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

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

q.display();

}

}

inserted :20

inserted:40

inserted:60

inserted:760

inserted:860

inserted:80

20

40

false

80

display

60

760

860

80