10주차 실습과제

20135151 이갑성

소스

**public** **class** Node {

**int** data;

Node link;

**public** Node()

{

**this**.data = 0;

link = **null**;

}

**public** Node(**int** val)

{

**this**.data = val;

link = **null**;

}

}

**public** **class** List {

Node head;

**public** **void** addFirst(**int** val) {

Node newNode = **new** Node(val);

**if** (head == **null**)

head = newNode;

**else** {

Node temp = head;

head = newNode;

newNode.link = temp;

}

}

**public** **void** addLast(**int** val) {

Node newNode = **new** Node(val);

**if** (head == **null**)

head = newNode;

**else** {

Node temp = head;

**while** (temp.link != **null**) {

temp = temp.link;

}

temp.link = newNode;

}

}

**public** **void** Print() {

Node temp = head;

**while** (temp.link != **null**) {

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

temp = temp.link;

}

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

}

**public** **void** Insert(Node target, **int** val) {

Node newNode = **new** Node(val);

**if** (head == **null**)

head = newNode;

**else** {

newNode.link = target.link;

target.link = newNode;

}

}

**public** **void** Delete(Node target) {

**if** (target.link == **null**) {

target = **null**;

} **else** **if** (target.link != **null**) {

target.link = target.link.link;

}

}

**public** **void** Deletelast() // 마지막 노트값 삭제

{

Node target1 = head;

Node target2 = **null**;

**while** (target1.link != **null**) {

target2 = target1;

target1 = target1.link;

}

target2.link = **null**;

}

**public** Node FindNode(**int** val)

{

Node target = head;

**while**(target.link != **null**)

{

**if**(target.data == val)

{

System.***out***.println("value값 "+val +"에 해당되는 data를 가진 노드를 찾았습니다. => 결과값:" + target.data);

**return** target;

}

target = target.link;

}

**return** **null**;

}

**public** **void** Reverse() {

Node p = head;

Node q = **null**;

Node r = **null**;

**while** (p != **null**) {

r = q;

q = p;

p = p.link;

q.link = r;

}

head = q;

}

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

List lt = **new** List();

lt.addFirst(10);

lt.addLast(20);

lt.addFirst(30);

lt.addLast(40);

lt.Print();

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

lt.FindNode(10);

lt.Deletelast();

lt.Reverse();

lt.Print();

}

}

결과값

