11주차 실습과제

20135151 이갑성

원형 리스트

**public** **class** CNode {

**int** data;

CNode link;

**public** CNode()

{

**this**.data = 0;

**this**.link = **null**;

}

**public** CNode(**int** val)

{

**this**.data = val;

**this**.link = **null**;

}

}

**public** **class** CList {

CNode head;

CNode tail;

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

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

**if**(tail == **null**)

{

tail = newNode;

head = newNode;

}

**else**

{

newNode.link = head;

head = newNode;

tail.link = head;

}

}

**public** **int** lengthC()

{

**if**(tail == **null**)

**return** 0;

**else**

{

**int** cnt = 1;

CNode p = tail.link;

**while**(p != tail)

{

p = p.link;

cnt++;

}

**return** cnt;

}

}

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

{

CNode p = head;

System.***out***.println("리스트 데이터값: ");

**while**(p != tail)

{

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

p = p.link;

}

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

}

**public** **void** Insert(CNode p, **int** val)

{

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

**if**(p == **null**)

{

p = newNode;

head = p;

tail = p;

p.link = head;

}

**else** **if**(p == tail)

{

p.link = newNode;

newNode.link = head;

tail = newNode; //중요!

}

**else** **if**(p != tail)

{

newNode.link = p.link;

p.link = newNode;

}

}

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

CList cl = **new** CList();

cl.addFirst(10);

cl.Insert(cl.head, 20);

cl.addFirst(30);

System.***out***.println("리스트의 길이: " + cl.lengthC());

cl.Print();

}

}

결과



이중연결리스트

**public** **class** DNode {

**int** data;

DNode Pre\_link;

DNode Next\_link;

**public** DNode()

{

**this**.data = 0;

Pre\_link = **null**;

Next\_link = **null**;

}

**public** DNode(**int** val)

{

**this**.data = val;

Pre\_link = **null**;

Next\_link = **null**;

}

}

**public** **class** DList {

DNode head;

**public** **void** Insert(DNode p, **int** val) {

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

**if** (head == **null**) { // 링크에 값이 아무것도 없을때

head = newNode; // head에 새로운값을 넣는다

head.Pre\_link = head; // head의 이전주소값을 자기 자신을 가르치게 한다.

head.Next\_link = head; // head의 다음주소값을 자기 자신을 가르치게 한다. 왜냐면 원형의 특성을

// 갖고 있으므로

}

**else** **if** (p == **null**) { // p노드의 다음값에 노드를 추가하려고했는데 정작 p노드가 없을때

p = newNode; // p에 새로운 값을 입력한다.

p.Pre\_link = head.Next\_link; // p(새로운값)에 이전주소를 head의 다음주소를 가르키게 한다.

p.Next\_link = head; // p(새로운값)에 다음주소를 head를 가르키게한다.

head.Next\_link = p.Pre\_link; // head의 다음주소에 p(새로운값)의 이전주소를 가르키게한다.

head.Pre\_link = p; // head의 이전주소는 p(새로운값)를 가르키게한다.

}

**else** {

newNode.Next\_link = p.Next\_link; // 새로운 노드의 다음주소를 p의 다음값을 가르키게한다.

newNode.Pre\_link = p; // 새로운 노드의 이전주소에 p를 가르키게한다.

p.Next\_link.Pre\_link = newNode; // newNode의 다음주소의 노드에서 이전 주소를 새로운 노드를

// 가르키게한다.

p.Next\_link = newNode; // p의 다음주소를 새로운 노드를 가르키게 한다.

}

}

**public** **void** Delete(DNode p) {

**if** (p == **null**) { // 삭제할 노드 p가 없는 경우

System.***out***.println("삭제할 값이 존재하지않습니다.");

} **else** **if** (p == head) // 삭제할 노드 p가 head인 경우

{

head = p.Next\_link; // head를 p의 다음주소의 노드로 변경한다.

p.Next\_link.Pre\_link = p.Pre\_link; // p의 다음주소의 노드의 이전주소를 헤드의

// 이전주소를 넣어준다.

p.Pre\_link.Next\_link = p.Next\_link; // p의 이전주소의 노드의 다음주소를 p의 다음주소를

// 넣어준다.

}

**else** **if** (p.Next\_link == head) // p의 넥스트링크가 헤드일때, => p가 마지막 노드일때

{

p.Pre\_link.Next\_link = head; // p의 이전주소의 노드의 다음주소를 head를 가르킨다.

head.Pre\_link = p.Pre\_link; // head의 이전주소는 p의 이전주소를 넣어준다.

}

**else** {

p.Pre\_link.Next\_link = p.Next\_link; // p의 이전주소의 노드에서 다음주소를 p의 다음주소를 넣어준다.

p.Next\_link.Pre\_link = p.Pre\_link; // p의 다음주소의 노드에서 이전주소에 p의 이전주소를 넣어준다.

}

}

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

System.***out***.println("값 출력: ");

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

System.***out***.println("출력할 값이 없습니다.");

} **else** {

DNode p = head;

**while** (p.Next\_link != head) {

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

p = p.Next\_link;

}

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

}

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

}

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

DList dl = **new** DList();

dl.Insert(dl.head, 10);

dl.Insert(dl.head, 20);

dl.Insert(dl.head.Next\_link, 30);

dl.Insert(dl.head.Next\_link.Next\_link, 40);

dl.Print();

System.***out***.println("Delete: " + dl.head.Next\_link.data);

dl.Delete(dl.head.Next\_link);

dl.Print();

}

}

결과

