

[HW#4] 자료구조론 실습

Circular Double Linkded List 구현

2014037901 컴퓨터공학
나 윤 환

todo#1

```
Node(int input){
    this.data = input;
    this.next = null;
    this.prev = null;
}
```

todo#2

```
public void insert(int position, int input){
    Node newNode = new Node(input);
    Node node = head;

    Node firstNode = null;
    Node lastNode = null;

    if(count > 0){
        firstNode = head.next;
        lastNode = firstNode.prev;
    }

    // position - 1 Node로 이동한다.
    for(int i = 0; i < position; i++){
        node = node.next;
    }

    // node = position - 1 Node
    // newNode의 이전노드를 node(position-1)로 바꾸고
    // newNode의 다음노드를 node(position-1)의 next노드로 설정한다.

    newNode.prev = node;
    newNode.next = node.next;

    // 그 다음 node(position-1)의 next를 newNode로 설정한다.
    if(count != 0) node.next.prev = newNode;
    node.next = newNode;

    // 만약 맨 마지막에 노드를 넣을 경우에
    if(position == count){
        // head.next = firstNode
        // firstNode의 prevNode를 newNode로 바꾸고
        // newNode(LastNode)의 nextNode를 firstNode로 바꾼다.
        head.next.prev = newNode;
        newNode.next = head.next;
    }

    if(position == 0 && count > 0){
        // 0번째 자리에 노드를 넣고, count가 0 이상이면
        // newNode(새로운 firstNode)의 prev를 lastNode로
        // lastNode의 next를 newNode(firstNode)

        newNode.prev = lastNode;
        lastNode.next = newNode;
    }

    // 리스트 길이를 1만큼 증가시킨다.
    count++;
}
```

```

public void delete(int position){

    // node(position-1)노드로 가져온다.
    Node node = head;
    for(int i = 0; i < position; i++){
        node = node.next;
    }

    // curNode(node.next)의 다음 노드의 prev값을 node(position-1)로 바꾼다.
    // node(position-1)의 다음 노드를 curNode의 다음 노드로 바꾼다.
    node.next.next.prev = node;
    node.next = node.next.next;

    // 길이를 1만큼 감소시킨다.
    count--;
}

```

```

public void print_list_forward(int position){

    // node(positonNode)를 가져온다.
    Node node = head;
    for(int i = 0; i <= position; i++){
        node = node.next;
    }

    // node(positionNode)를 startNode로 설정한다.
    Node startNode = node;

    // 무한루프 시작
    while(true){
        System.out.print(node.data+", ");
        // startNode부터 시작해서 다음노드로 진행한다.
        node = node.next;

        // 만약에 node가 startNode와 같다면 무한루프 종료
        if(node == startNode) break;
    }

    System.out.println();
}

```

```

public void print_list_backward(int position){
    // node(positionNode)를 가져온다.
    Node node = head;
    for(int i = 0; i <= position; i++){
        node = node.next;
    }

    // node(positionNode)를 startNode로 설정한다.
    Node startNode = node;

    // 무한루프 시작
    while(true) {
        System.out.print(node.data+" ");
        // startNode부터 시작해서 이전노드로 진행한다.
        node = node.prev;

        // 만약에 node가 startNode와 같다면 무한루프 종료
        if(node == startNode) break;
    }

    System.out.println();
}

```

```

public void get_data(int position){
    // node(positionNode)를 가져온다.
    Node node = head;
    for(int i = 0; i <= position; i++){
        node = node.next;
    }

    // node(positionNode)의 데이터 값 출력
    System.out.println(node.data);
}

```

todo#3

```
//TODO #3
//Math.Random을 사용하여 50개의 값을 리스트에 삽입 list1.print_lis
for(int i = 0; i < 20; i++){
    int rndInt = (int)(Math.random()*10)+1;
    list1.insert(i, rndInt);
}
```

실행화면

```
Apple-i-Air-2:day5 nayunhwan$ java Run
6, 2, 6, 6, 4, 8, 2, 3, 3, 7, 8, 2, 9, 2, 1, 1, 5, 5, 9, 10,
6, 10, 9, 5, 5, 1, 1, 2, 9, 2, 8, 7, 3, 3, 2, 8, 4, 6, 6, 2,
5, 5, 9, 10, 6, 6, 4, 8, 2, 3, 3, 7, 8, 2, 9, 2, 1, 1,
5, 1, 1, 2, 9, 2, 8, 7, 3, 3, 2, 8, 4, 6, 6, 10, 9, 5,
6
Apple-i-Air-2:day5 nayunhwan$
```