

자료 구조 Lab 006 :

Lab18006.zip : LabTest.java, lab006.java, lab.in, lab.out, lab006.pdf

제출

lab006.java 를 학번.java 로 변경하여 이 파일 한 개만 제출할 것.

다음은 Linked List의 사용에 관한 문제이다.

이번 연습은 **Doubly Linked List with Header Node** 에서 원소의 **삽입은 처음 위치에서** 일어나고, 원소의 **삭제는 임의의** 위치에서 일어나는 기능을 제공하는 DbList라는 클래스를 구현하는 것이다.

수행 예는 다음과 같다.

```
kmucs@localhost: ~/dbox/classes181/ds/lab18/lab18006
kmucs@localhost:~/dbox/classes181/ds/lab18/lab18006$ java LabTest
Dbl >
ins 5
CMD : ins 5
Dbl List From the Last (1) : 5
List has 1 elements

Dbl >
ins 8
CMD : ins 8
Dbl List From the Last (2) : 5 8
List has 2 elements

Dbl >
ins 7
CMD : ins 7
Dbl List From the Last (3) : 5 8 7
List has 3 elements

Dbl >
del 8
CMD : del 8
Dbl List From the Last (2) : 5 7
List has 2 elements

Dbl >
```

사용자가 사용하는 명령어의 syntax는 다음과 같다. LabTest.java의 main() 함수에 정의되어 있다.

- `ins : ins item`

리스트의 맨 앞에 `item`을 추가한다.

- `del : del item`

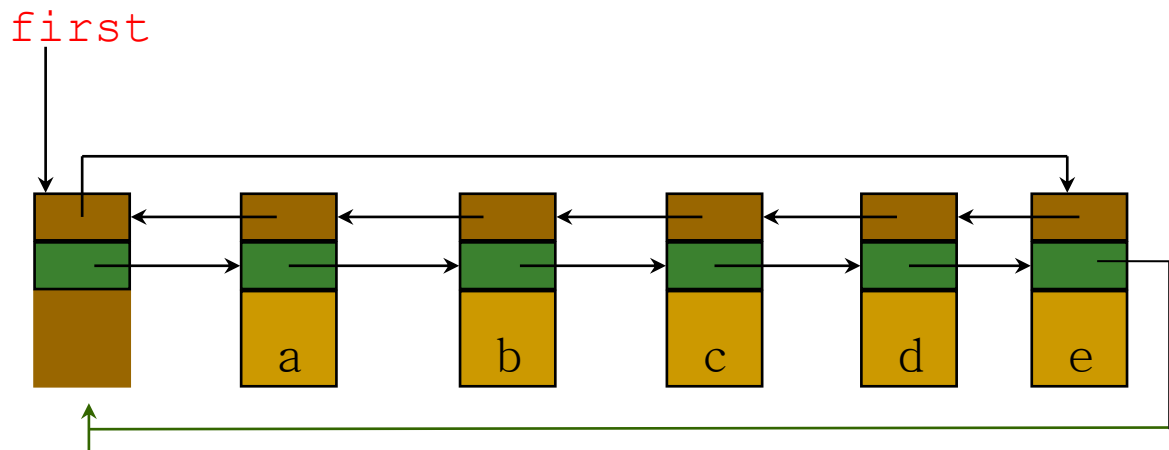
`item`을 원소로 갖는 노드를 찾아서 그 노드를 제거한다.

- `quit : quit`

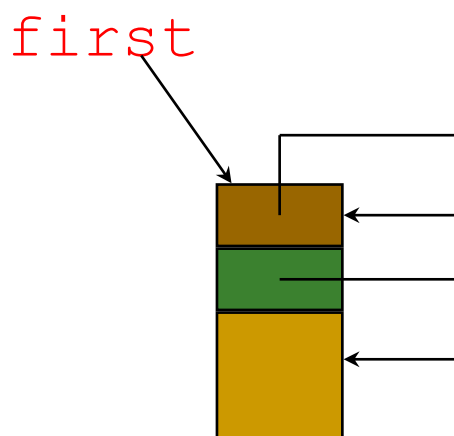
프로그램을 종료한다.

명령이 실행되고 나면 리스트의 내용이 출력되는데, 괄호 안의 숫자는 리스트의 원소의 수이다. 그리고 리스트는 맨 마지막 원소부터 앞으로 가면서 출력한다. 이 내용은 이미 구현되어 있으니 구현하지 않아도 된다.

구현할 `DblList`클래스는 원소들을 아래와 같은 구조로 유지한다.



리스트가 empty 일경우는 다음과 같다.



구현이 필요한 부분은 다음 세 함수이다. 구체적인 수행내용도 함께 설명한다.

- `final int Size();`
이 함수는 현재 리스트에 들어 있는 원소의 수를 `return`하는 함수이다. 위 그림에서는 5를 리턴한다.
- `void InsertFront(final T e);`
이 함수는 리스트의 맨 앞에 새로운 노드를 하나 추가 하고, 그의 데이터 필드에 `e`를 입력하는 함수이다.
- `boolean Delete(final T e);`
`e`를 원소로 갖는 노드를 찾아서 이를 제거하는 함수이다. 해당 노드가 존재하면 `true`를 리턴하고, 그렇지 않을 경우 `false`를 리턴한다.

프로그램 테스트

컴파일

```
$ javac lab006.java LabTest.java
```

실행

```
$ java LabTest
```

주어진 **input**으로 실행

```
$ java LabTest < lab.in
```

주어진 **output**과 비교

```
$ java LabTest < lab.in > abc
```

```
$ diff abc lab.out
```