

자료구조 실습 보고서

[제05주] 스택

2018/04/12

201402414 장수훈

```

1 package H05;
2
3 public class IntNodeStack implements IntStack{
4     private Node head;
5     private int size;
6
7     public IntNodeStack() {
8         this.head = null;
9         this.size = 0;
10    }
11    public boolean isEmpty() {
12        return (size == 0);
13    }
14    public int pop() {
15        if(size == 0)
16            return -1;
17        int n = this.head.getData();
18        this.head = this.head.getNext();
19        this.size--;
20        return n;
21    }

```

<IntNodeStack 클래스-1>

[14~21] POP 함수는 처음 데이터를 지우는 것 이므로 헤드 다음 노드를 헤드로 한다. 그러면 헤드가 지워지고 다음 노드부터 시작된다. 이러면 앞의것이 없어진 것이 되니까 사이즈를 줄인다.

지워지는 값을 확인하기 위해서 인트형 n 변수를 선언하였고 값으로는 헤드의 값이 들어간다.

지워질것이 없으면(size == 0) 이면 -1 을 리턴하고 아니면 지우는 함수를 실행한다.

```

22 public int popBottom() {
23     Node temp = null;
24     Node temp1 = this.head;
25     if(this.size > 1) {
26         for(int i = 1 ; i < this.size; i ++ ) {
27             temp = temp1;
28             temp1 = temp1.getNext();
29         }
30         int n = temp1.getData();
31         temp.setNext(null);
32         this.size--;
33         return n;
34     }
35     else {
36         if(size == 0) {
37             return -1;
38         }
39         int n = this.head.getData();
40         this.head = this.head.getNext();
41         this.size--;
42         return n;
43     }
44 }
45 }
46 public void push(int n) {
47     Node newNode = new Node(n);
48     newNode.setNext(this.head);
49     this.head = newNode;
50     this.size++;
51 }
52 public int size() {
53     return size;
54 }

```

<IntNodeStack 클래스-2>

[22~45] Popbottom 은 나중에 들어온 스택을 지우는 함수이다. 사이즈가 0 이면 -1 을 리턴하고(데이터가 없는데 지울때) 1 개만 있을경우 pop 과 동일하게 삭제를 하였고, 1 이상인 경우 노드를 2 개를 만들어 마지막값을 n에 집어넣고(무엇이 지워졌는지 인트값을 리턴하기 위해) 마지막전 노드를 null 을 가르키게 했다. 마지막 스택이 없어졌으므로 사이즈를 하나 줄인다.

<결과화면 출력>

내용 출력 :

2 1

번호를 입력하세요

- 1 >>> 스택에 내용 추가
 - 2 >>> 스택에 마지막으로 들어온 내용 삭제
 - 3 >>> 스택에 처음 들어온 내용 삭제
 - 4 >>> 사이즈출력
 - 5 >>> 내용출력
 - 6 >>> 종료
- : 2

마지막 스택을 삭제합니다

[2] 삭제되었습니다

번호를 입력하세요

- 1 >>> 스택에 내용 추가
 - 2 >>> 스택에 마지막으로 들어온 내용 삭제
 - 3 >>> 스택에 처음 들어온 내용 삭제
 - 4 >>> 사이즈출력
 - 5 >>> 내용출력
 - 6 >>> 종료
- : 5

내용 출력 :

1

번호를 입력하세요

- 1 >>> 스택에 내용 추가
 - 2 >>> 스택에 마지막으로 들어온 내용 삭제
 - 3 >>> 스택에 처음 들어온 내용 삭제
 - 4 >>> 사이즈출력
 - 5 >>> 내용출력
 - 6 >>> 종료
- : 2

마지막 스택을 삭제합니다

[1] 삭제되었습니다

내용 출력 :

번호를 입력하세요

- 1 >>> 스택에 내용 추가
 - 2 >>> 스택에 마지막으로 들어온 내용 삭제
 - 3 >>> 스택에 처음 들어온 내용 삭제
 - 4 >>> 사이즈출력
 - 5 >>> 내용출력
 - 6 >>> 종료
- : 2

마지막 스택을 삭제합니다

삭제할 데이터가 더이상 없습니다.

- 1과 2를 넣었고 마지막으로 들어온 스택을 하나씩 지웠다. (기능 1,2,5 사용)

내용 출력 :

2 1

번호를 입력하세요

- 1 >>> 스택에 내용 추가
- 2 >>> 스택에 마지막으로 들어온 내용 삭제
- 3 >>> 스택에 처음 들어온 내용 삭제
- 4 >>> 사이즈출력
- 5 >>> 내용출력
- 6 >>> 종료

: 3

처음 스택을 삭제합니다

[1] 삭제되었습니다

번호를 입력하세요

- 1 >>> 스택에 내용 추가
- 2 >>> 스택에 마지막으로 들어온 내용 삭제
- 3 >>> 스택에 처음 들어온 내용 삭제
- 4 >>> 사이즈출력
- 5 >>> 내용출력
- 6 >>> 종료

: 5

내용 출력 :

2

번호를 입력하세요

- 1 >>> 스택에 내용 추가
- 2 >>> 스택에 마지막으로 들어온 내용 삭제
- 3 >>> 스택에 처음 들어온 내용 삭제
- 4 >>> 사이즈출력
- 5 >>> 내용출력
- 6 >>> 종료

: 3

처음 스택을 삭제합니다

[2] 삭제되었습니다

내용 출력 :

번호를 입력하세요

- 1 >>> 스택에 내용 추가
- 2 >>> 스택에 마지막으로 들어온 내용 삭제
- 3 >>> 스택에 처음 들어온 내용 삭제
- 4 >>> 사이즈출력
- 5 >>> 내용출력
- 6 >>> 종료

: 3

처음 스택을 삭제합니다

삭제할 데이터가 더이상 없습니다.

- 1과 2를 넣었고 스택에 처음 들어온 것부터 하나씩 지웠다. (기능 1,3,5 사용)

내용 출력 :

3 2 1

번호를 입력하세요

- 1 >>> 스택에 내용 추가
- 2 >>> 스택에 마지막으로 들어온 내용 삭제
- 3 >>> 스택에 처음 들어온 내용 삭제
- 4 >>> 사이즈출력
- 5 >>> 내용출력
- 6 >>> 종료

: 4

사이즈 : 3

내용 출력 :

2 1

번호를 입력하세요

- 1 >>> 스택에 내용 추가
- 2 >>> 스택에 마지막으로 들어온 내용 삭제
- 3 >>> 스택에 처음 들어온 내용 삭제
- 4 >>> 사이즈출력
- 5 >>> 내용출력
- 6 >>> 종료

: 4

사이즈 : 2

- 사이즈 출력 (4번 기능 사용)