

컴프2 6주차 : 스택의 활용

과제 목표 : 스택을 활용할 수 있다.

학번 : 201902722

이름 : 유형곤

1번 과제 : GoodWord

문제해결과정 및 코드 설명 :

GoodWord는 유명한 스택 문제로, 다음과 같은 규칙을 따르면 문제를 해결할 수 있습니다.

1. 스택이 비어있거나, top에 현재 문자와 다른 문자가 들어있다면 스택에 그 문자를 push한다
2. top에 현재 문자와 같은 문자가 들어있다면 pop한다.

1~2번을 문자열의 끝까지 수행한 뒤, 스택이 비어있다면 좋은 단어이다.

이 문제는 스택이 아니라 리스트를 사용하는 조건이 있는데, 사실 스택은 배열, 리스트로 구현할 수 있기 때문에 스택 인터페이스를 쓰지 않을 뿐이지, 리스트로도 스택을 구현할 수 있습니다. 다만 리스트에 add를 하는 방향과 remove를 하는 방향만 하나로 통일시켜준다면 (예를들어 addFirst로 원소를 추가하면 removeFirst로 원소를 제거해야합니다.) 스택과 같이 사용할 수 있습니다.

```

int cnt = 0;
for(int i = 0; i < n; i++) {
    String line = br.readLine();
    LinkedList<Character> list = new LinkedList<Character>();
    //문자열을 스택에 넣는다.
    for(int j = 0; j < line.length(); j++) {
        char ch = line.charAt(j);
        if(ch == 'A') {
            if(list.size() == 0) {
                list.addFirst(ch);
            } else if(list.get(0) == 'A') {
                list.removeFirst();
            } else if(list.get(0) == 'B') {
                list.addFirst(ch);
            }
        } else {
            if(list.size() == 0) {
                list.addFirst(ch);
            } else if(list.get(0) == 'B') {
                list.removeFirst();
            } else if(list.get(0) == 'A') {
                list.addFirst(ch);
            }
        }
    }
    if(list.size() == 0) {
        cnt++;
    }
}
System.out.println(cnt);

```

실행결과:

```

<terminated> GoodWord [Java Application] C:\Program Files\
3
AABB
ABBA
ABAB
2

```

2번 과제 : VPS (Valid Parenthesis String)를 검사하는 문제

문제해결과정 및 코드 설명 :

2번 과제는 유효한 괄호 문자열을 검사하는 문제로, 대표적인 스택을 활용하는 문제입니다.

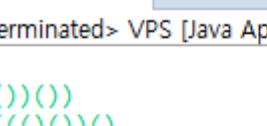
왜냐하면 나중에 열린 괄호가 가장 먼저 닫히기 때문에, 스택의 후입선출이라는 특징을 활용할 수 있기 때문입니다. 유효한 괄호 문자열의 검사는 다음과 같이 진행할 수 있습니다.

1. 괄호를 여는 문자열 (이 나오면, 스택에 넣는다.
2. 괄호를 닫는 문자열)이 나오면, 스택에서 제거한다. 단, 이때 스택이 비어있다면 유효하지 않은 문자열이다.
3. 1~2를 문자열의 끝까지 진행했을 때, 스택이 비어 있다면 유효한 문자열이고, 그렇지 않다면 유효하지 않은 문자열이다.

또한, 이 문제는 스택을 실제로 사용하지 않고 열린 괄호의 개수를 세는 것으로도 문제를 해결할 수 있습니다. 왜냐하면 스택에 들어있는 괄호의 차이점(인덱스 등)을 여기서는 구분할 필요가 없기 때문입니다.

```
for(int i = 0; i < n; i++) {  
    String line = br.readLine();  
    Stack<Character> stack = new Stack<Character>();  
    String answer = null;  
    //문자열을 스택에 넣는다.  
    for(int j = 0; j < line.length(); j++) {  
        char ch = line.charAt(j);  
        if(ch == '(') {  
            stack.push(ch);  
        } else {  
            if(stack.size() == 0) {  
                answer = "NO";  
                break;  
            } else {  
                stack.pop();  
            }  
        }  
    }  
  
    if(answer == null) {  
        if(stack.size() == 0) {  
            answer = "YES";  
        } else {  
            answer = "NO";  
        }  
    }  
    sb.append(answer).append("\n");  
}
```

실행결과 :



The screenshot shows a Java application window titled "Console" with a close button. The text inside the window is as follows:

```
<terminated> VPS [Java Application] C:\Program Files\Ja
|5
(( ))( )
((( ))( ))( )
(( ))( )(( ))
((( ))( ))( )((( ))( ))( )
(( ))( ))( ))( ))( ))( )
(( ))( ))( ))( ))( ))( )
(( ))( ))( )
NO
NO
YES
NO
YES
NO
```