

227. Basic Calculator II

Description

Hints

Submissions

Discuss

Solution

Pick One

Implement a basic calculator to evaluate a simple expression string.

The expression string contains only **non-negative** integers, `+`, `-`, `*`, `/` operators and empty spaces . The integer division should truncate toward zero.

Example 1:

Input: "3+2*2"
Output: 7

Example 2:

Input: " 3/2 "
Output: 1

Example 3:

Input: " 3+5 / 2 "
Output: 5

Note:

- You may assume that the given expression is always valid.
- Do not use the `eval` built-in library function.

```

5 public class L227 {
6     public int calculate(String s) {
7
8         if(s == null || s.length() == 0) {
9             return 0;
10        }
11        Stack<Integer> stk = new Stack<>();
12        int num = 0, res = 0;
13        char lastSign = '+';
14        char [] cArray = s.toCharArray();
15
16        for(int i = 0; i < cArray.length; i++) {
17            char c = cArray[i];
18            if(c >= '0' && c <= '9') {
19                //这个是怕num不止一位数
20                num = num * 10 + c - '0';
21            }
22            if(c == '+' || c == '-' || c == '*' || c == '/' || i == cArray.length - 1) {
23                if(lastSign == '+' || lastSign == '-') {
24                    int temp = lastSign == '+' ? num : -num;
25                    stk.push(temp);
26                    res += temp;
27                }
28                //如果上一个乘法或者是除法，那么把上次结果弹出，计算后再加入
29                if(lastSign == '/' || lastSign == '*') {
30                    res -= stk.peek();
31                    int temp = lastSign == '*' ? stk.pop() * num : stk.pop() / num;
32                    stk.push(temp);
33                    res += temp;
34                }
35                lastSign = c;
36                num = 0;
37            }
38        }
39        return res;
40    }
41 }

```