Question: https://leetcode.com/problems/basic-calculator-ii/

By BODMAS rule we know that + and - will take place after \* and / have been performed.

So for + and - we store the numbers as pos or negative and keep track of last encountered operator.

So when we encounter \* and / we perform the operation with top value of stack and the present number.

So when are done with filling the stack with calculated values, now we need to perform + and - operators.

But since we have stored the results as +ve and -ve numbers, so just summing up all values of stack will do the job.

Code:  
class Solution {

public int calculate(String s) {

Stack<Integer> stck = new Stack<Integer>();

char op = '+';

int res=0, num=0, len = s.length();

for(int i=0; i<len; i++){

char c = s.charAt(i);

if(Character.isDigit(c)){

num\*=10;

num+=c-'0';

}

if((!Character.isDigit(c) && c!=' ')||i==len-1){

if(op == '+'){

stck.push(num);

}else if(op == '-'){

stck.push(-num);

}else if(op == '\*'){

stck.push(stck.pop()\*num);

}else if(op == '/'){

stck.push(stck.pop()/num);

}

num = 0;

op = c;

}

}

while(!stck.isEmpty()){

res+=stck.pop();

}

return res;

}

}

Github Link :<https://lnkd.in/ecwtJeaz>