Linked List To Stack Adapter

Try First, Check Solution later

1. You should first read the question and watch the question video.  
2. Think of a solution approach, then try and submit the question on editor tab.  
3. We strongly advise you to watch the solution video for prescribed approach.

Question

1. You are required to complete the code of our LLToStackAdapter class.   
2. As data members, you've a linkedlist available in the class.  
3. Here is the list of functions that you are supposed to complete  
 3.1. push -> Should accept new data in LIFO manner  
 3.2. pop -> Should remove and return data in LIFO manner. If not   
 available, print "Stack underflow" and return -1.  
 3.3. top -> Should return data in LIFO manner. If not available, print   
 "Stack underflow" and return -1.  
 3.4. size -> Should return the number of elements available in the   
 stack  
4. Input and Output is managed for you.  
  
Note -> The intention is to use linked list functions to achieve the purpose of a stack. All the functions should work in constant time.

Input Format

Input is managed for you

Output Format

Output is managed for you

Constraints

None

Sample Input

push 10  
push 20  
push 5  
push 8  
push 2  
push 4  
push 11  
top  
size  
pop  
top  
size  
pop  
top  
size  
pop  
top  
size  
pop  
top  
size  
pop  
top  
size  
pop  
top  
size  
pop  
quit

Sample Output

11  
7  
11  
4  
6  
4  
2  
5  
2  
8  
4  
8  
5  
3  
5  
20  
2  
20  
10  
1  
10

import java.io.\*;

import java.util.\*;

public class Main {

public static class LLToStackAdapter {

LinkedList<Integer> list;

public LLToStackAdapter() {

list = new LinkedList<>();

}

int size() {

// write your code here

return list.size();

}

void push(int val) {

list.addFirst(val);

}

int pop() {

if(list.size()==0)

{

System.out.println("Stack underflow");

return -1;

}else{

return list.removeFirst();

}

}

int top() {

if(list.size()==0)

{

System.out.println("Stack underflow");

return -1;

}else{

return list.getFirst();

}

}

}

public static void main(String[] args) throws Exception {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

LLToStackAdapter st = new LLToStackAdapter();

String str = br.readLine();

while(str.equals("quit") == false){

if(str.startsWith("push")){

int val = Integer.parseInt(str.split(" ")[1]);

st.push(val);

} else if(str.startsWith("pop")){

int val = st.pop();

if(val != -1){

System.out.println(val);

}

} else if(str.startsWith("top")){

int val = st.top();

if(val != -1){

System.out.println(val);

}

} else if(str.startsWith("size")){

System.out.println(st.size());

}

str = br.readLine();

}

}

}