Linked List To Queue 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 LLToQueueAdapter 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. add -> Should accept new data in FIFO manner  
 3.2. remove -> Should remove and return data in FIFO manner. If not available,   
 print "Queue underflow" and return -1.  
 3.3. peek -> Should return data in FIFO manner. If not available, print "Queue   
 underflow" and return -1.  
 3.4. size -> Should return the number of elements available in the queue  
4. Input and Output is managed for you.  
  
Note -> The intention is to use linked list functions to achieve the purpose of a queue. 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

add 10  
add 20  
add 30  
add 40  
add 50  
add 60  
peek  
remove  
peek  
remove  
peek  
remove  
peek  
remove  
peek  
remove  
peek  
remove  
quit

Sample Output

10  
10  
20  
20  
30  
30  
40  
40  
50  
50  
60  
60

import java.io.\*;

import java.util.\*;

public class Main {

public static class LLToQueueAdapter {

LinkedList<Integer> list;

public LLToQueueAdapter() {

list = new LinkedList<>();

}

int size() {

return list.size();

}

void add(int val) {

// write your code here

list.addLast(val);

}

int remove() {

if(size() == 0){

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

return -1;

}else{

return list.removeFirst();

}

}

int peek() {

if(size() == 0){

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

return -1;

}else{

return list.getFirst();

}

}

}

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

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

LLToQueueAdapter qu = new LLToQueueAdapter();

String str = br.readLine();

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

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

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

qu.add(val);

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

int val = qu.remove();

if (val != -1) {

System.out.println(val);

}

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

int val = qu.peek();

if (val != -1) {

System.out.println(val);

}

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

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

}

str = br.readLine();

}

}

}