package Stack;  
  
import java.util.EmptyStackException;  
public class Expression {  
 public boolean isValidExpr(String *expr*) {  
 Stack<Character> stack = new Stack<>();  
 char c;  
 for (int i = 0; i < *expr*.length(); i++) {  
 c = *expr*.charAt(i);  
 if( c == '('){  
 stack.push(c);  
 }  
 if (c == ')'){  
 if(!stack.isEmpty()){  
 stack.pop();  
 }  
 else{  
 return false;  
 }  
 }  
 }  
 return stack.isEmpty();  
 }  
  
  
 public static void main(String[] *args*) {  
  
 Expression expr = new Expression();  
 String f = "a+b-c(3+a)";  
 System.*out*.println(expr.isValidExpr(f));  
  
 }  
 class Stack<*T*> {  
 private Node<*T*> **top**;  
 private int **size**;  
  
 private class Node<*T*> {  
 private *T* **data**;  
 private Node<*T*> **next**;  
  
 private Node(*T data*) {   
 this.**data** = *data*;  
 }  
 }  
  
 public void push(*T data*) {  
 Node<*T*> node = new Node<>(*data*);  
 node.**next** = **top**;  
 **top** = node;  
 **size**++;  
 }  
  
 public *T* pop() {  
 if (isEmpty()) {  
 throw new EmptyStackException();  
 }  
 *T* data = **top**.**data**;  
 **top** = **top**.**next**;  
 **size**--;  
 return data;  
 }  
  
 public *T* peek() {  
 if (isEmpty()) {  
 throw new EmptyStackException();  
 }  
 return **top**.**data**;  
 }  
  
 public boolean isEmpty() {  
 return **top** == null;  
 }  
  
 public int size() {  
 return **size**;  
 }  
 }   
  
  
}