

**Instituto Tecnológico de Nuevo León**

Lenguaje y Autómatas II

Unidad 1

Jesús Alexis Jiménez Reyna N. control: 15480465

Carrera: Ing. Sistemas Computaciones

Ing. Juan Pablo Rosas Baldazo

N.L. Guadalupe de noviembre del 2018

|  |
| --- |
| package arbol; |
|  |  |
|  | import java.util.Scanner; |
|  | import java.util.Stack; |
|  |  |
|  | public class ArbolExpresiones { |
|  |  |
|  | public static void main(String[] args) { |
|  |  |
|  | System.out.println("Escribe una expresión: "); |
|  |  |
|  | Scanner leer = new Scanner(System.in); |
|  | String exp = depurar(leer.nextLine()); |
|  | String[] arrayInfix = exp.split(" "); |
|  | Stack< String> En = new Stack< >(); |
|  | Stack< String> Pt = new Stack< >(); |
|  | Stack< String> Sa = new Stack< >(); |
|  |  |
|  | for (int i = arrayInfix.length - 1; i >= 0; i--) { |
|  |  |
|  | En.push(arrayInfix[i]); |
|  |  |
|  | } |
|  | try { |
|  | while (!En.isEmpty()) { |
|  | switch (pref(En.peek())) { |
|  |  |
|  | case 1: |
|  | Pt.push(En.pop()); |
|  | break; |
|  |  |
|  | case 4: |
|  | while (pref(Pt.peek()) >= pref(En.peek())) { |
|  | Sa.push(Pt.pop()); |
|  | } |
|  |  |
|  | Pt.push(En.pop()); |
|  |  |
|  | break; |
|  | case 2: |
|  | while (!Pt.peek().equals("(")) { |
|  | Sa.push(Pt.pop()); |
|  | } |
|  |  |
|  | Pt.pop(); |
|  | En.pop(); |
|  | break; |
|  |  |
|  | default: |
|  | Sa.push(En.pop()); |
|  | } |
|  | } |
|  |  |
|  | String infix = exp.replace(" ", ""); |
|  | String postfix = Sa.toString().replaceAll("[\\]\\[,]", ""); |
|  |  |
|  | System.out.println("Expresion Infija: " + infix); |
|  | System.out.println("Expresion Postfija: " + postfix); |
|  |  |
|  | } catch (Exception ex) { |
|  | System.out.println("Error en la expresión: "); |
|  | System.err.println(ex); |
|  | } |
|  | } |
|  | private static String depurar(String s) { |
|  |  |
|  | s = s.replaceAll("\\s+", ""); |
|  | s = "(" + s + ")"; |
|  | String simbols = "+-\*/()"; |
|  | String str = ""; |
|  |  |
|  | for (int i = 0; i < s.length(); i++) { |
|  | if (simbols.contains("" + s.charAt(i))) { |
|  | str += " " + s.charAt(i) + " "; |
|  | } else { |
|  | str += s.charAt(i); |
|  | } |
|  | } |
|  | return str.replaceAll("\\s+", " ").trim(); |
|  | } |
|  |  |
|  | private static int pref(String op) { |
|  |  |
|  | int prf = 99; |
|  | if (op.equals("^")) { |
|  | prf = 5; |
|  | } |
|  | if (op.equals("\*") || op.equals("/")) { |
|  | prf = 4; |
|  | } |
|  | if (op.equals("+") || op.equals("-")) { |
|  | prf = 3; |
|  | } |
|  | if (op.equals(")")) { |
|  | prf = 2; |
|  | } |
|  | if (op.equals("(")) { |
|  | prf = 1; |
|  | } |
|  | return prf; |
|  | } |
|  | } |