NOTACION POLACA

HERNANDEZ DEL NAGEL ANGEL IVAN
HERNANDEZ GERONIMO ELIEZER

LENGUAJES Y AUTOMATAS 2

CLASS = “CONVERSION”

```java
import java.util.Stack;
public class Conversion {  2 usages
    static StringBuilder conversionPrefijo(String expresion) {  // Nos facilita el uso de pilas  1 usage
        StringBuilder exprefija = new StringBuilder();          // Guarda la expresión prefija
        StringBuilder exinfija = new StringBuilder(expresion);  // Guarda la expresión infija (entrada)
        exinfija.reverse();                                     // Revierte el orden de la expresión
        Stack<Character> stack = new Stack<Character>();        // Se crea una pila de tipo caracter
        char[] carexp = new String(exinfija).toCharArray();     // Arreglo de caracteres

        for (int i = 0; i < carexp.length; i++) {               // Se utiliza para corregir los paréntesis al hacer el reverse
            if(carexp[i] == '(') {
                carexp[i] = ')';
                i++;
            } else if (carexp[i] == ')') {
                carexp[i] = '(';
                i++;
            }
        }

        for (int i = 0; i < carexp.length; i++) {               // Se comienza a llenar la pila
            char car = carexp[i];

            if (jerarquia(car) > 0) {                           // Compara operadores
                while (stack.isEmpty() == false && jerarquia(stack.peek()) >= jerarquia(car)) {
                    exprefija.append(stack.pop());              // Si se cumple, se concatena el elemento
                }
                stack.push(car);
            } else if (car == ')') {
                char aux = stack.pop();
                while (aux != '(') {
                    exprefija.append(aux);
```

CLASS = “CONVERSION”

```
32                      aux = stack.pop();
33                  }
34              } else if (car == '(') {
35                  stack.push(car);
36              } else {
37                  exprefija.append(car);
38              }
39          }
40
41          for (int i = 0; i <= stack.size(); i++) {              // Vacía la pila y genera la expresión prefija
42              exprefija.append(stack.pop());
43          }
44          return exprefija.reverse();
45      }
46
47      static int jerarquia(char car) {                          // Se evalua el nivel de jerarquía de los operadores
48          switch (car) {
49              case '+':
50              case '-':
51                  return 1;
52              case '*':
53              case '/':
54                  return 2;
55              case '^':
56                  return 3;
57          }
58          return -1;
59      }
60  }
```

CLASS = “MAIN”

```java
import java.util.Scanner;

public class NotacionPolaca {
    public static void main (String[] args) {
        System.out.print("Ingresa la expresion que deseas convertir: ");
        Scanner scan = new Scanner(System.in);
        String expresion = scan.nextLine();
        String prefija = Conversion.conversionPrefijo(expresion).toString();
        System.out.println("Expresion prefija generada: " + prefija);
    }
}
```

CONSOLE.OUTPUT

```
Ingresa la expresion que deseas convertir: x-(a+b)*d
Expresion prefija generada: -x*+abd

Process finished with exit code 0
```

```
Ingresa la expresion que deseas convertir: x*m*p+1
Expresion prefija generada: +*x*mp1

Process finished with exit code 0
```

CONSOLE.OUTPUT

```
Ingresa la expresion que deseas convertir: (a-1)*(2+b)
Expresion prefija generada: *-a1+2b

Process finished with exit code 0
```