```java
public class InfixaPosfixa {
    // retorna a precedência dos operadores. Quanto maior, maior é a prec
endência
    public int precedencia(char carac) {
        switch (carac) {
            case '+':
            case '-':
                return 1;
            case '*':
            case '/':
                return 2;

            case '^':
                return 3;
        }

        return -1;
    }

    // esse aqui é o bichão que converte infixa para posfixa;
    public String infixaParaPosfixa(String expressao) {
        // initializing empty String for result
        String resultado = new String("");

        // initializing empty stack
        Pilha pilha = new Pilha();

        for (int i = 0; i < expressao.length(); ++i) {
            char c = expressao.charAt(i);

            // se o caractere escaneado for um operando,a dicione ele no
output.
            if (Character.isLetterOrDigit(c)) resultado += c;

            // se o caractere escaneado for igual a (, jogue-
o dentro da pilha.
            else if (c == '(') pilha.push(c);

            // se o caractere escaneado for um ), popa a pilha, armazenan
do os valores no output, até encontrar o (.
            else if (c == ')') {
                while (!pilha.isEmpty() &&
                        pilha.peek() != '(')
                    resultado += pilha.pop();

                    pilha.pop();
            }
            // se um operador for encontrado.
            else {
```

```
                while (!pilha.isEmpty()
                       && precedencia(c) <= precedencia(pilha.peek())) {
                    resultado += pilha.pop();
             }
                pilha.push(c);
            }

        }

        // popar todos os operadores da pilha
        while (!pilha.isEmpty()){
            if(pilha.peek() == '(')
                return "EXPRESSÃO INVÁLIDA";
            resultado += pilha.pop();
         }
        return resultado;
    }


}
```