

## MANUAL DO USUÁRIO:

### JGJAVAINTERPRETOR

#### SINTAXE:

- Todos os comandos e variáveis devem ser separadas por espaços (um ou mais) sendo no mínimo um.
- Não é necessário uso de caractere de finalização de linha.
- A linguagem não lê caracteres de parênteses “( )” e nem escopo {}. Pode ser usado if em comando aninhado.
- Os operadores são os mesmo usados na linguagem C /Java. EX; % resto de divisão.

- **Operadores válidos:**

- **Adição:** var1 + var2 // Atribui valor
- **Subtração:** var 1 - var2 // Atribui valor
- **Multiplicação:** var 1 \* var2 // Atribui valor
- **Divisão:** var1 / var2 // Atribui valor
- **Verif. De Igualdade:** var 1 == var2 // Retorna true ou false
- **Verif. Diferente:** var 1 != var2 // Retorna true ou false
- **Maior:** var1 > var2 // Retorna true ou false
- **Menor:** var1 < var2 // Retorna true ou false
- **Resto:** var1 % var2 // Atribui valor
- **Incremento:** var 1 ++ // Incremento Var1
- **Decremento:** var 1 -- //Decrementa var1

- **Declaração de variáveis:**

#DEC NomeVariavel

#DEC NomeVariave l = 0

- **Atribuição de valor à variável:**

NomeVariavel = A + 10

- **Controle de fluxo (if):**

#IFF CONDIÇÃO

Código

#ENDIFF

- **Comenado de saída:**

#OUT (texto \$Valorvariavelc);

A função # OUT não realiza funções só imprime texto e o valor das variáveis.

***OBS.:*** Para imprimir o valor das variáveis usa-se \$variável.

## **RELAÇÃO DE ERROS**

ERRO 12 – O nome da variável não foi passado corretamente ou não existe.

## **PARA DESENVOLVEDORES:**

O interpretador foi desenvolvido seguindo a seguinte metodologia:

1º O arquivo de texto é passado para um vetor de linhas;

2º A análise é feita linha a linha;

3º Quebra um linha de cada vez em um SubStrings chamada pilha, que por sua vez é desempilhada;

4º A função desempilha é responsável por verificar a se há atribuição ou não na pilha, fazendo assim as demais operações primeiro, e por último verifica a existência de atribuição, e atribui.