

DOCUMENTAÇÃO DA LINGUAGEM COMPLICADORES

- Ter palavras-chave de início e fim de programa
- Um cabeçalho para declaração de variáveis (ex.: VAR a: <tipo>)
- Dois tipos numéricos: inteiro e real
- Um comando de escrita e outro de leitura de variáveis (não é necessário escrever mensagens de texto)
- Operações de adição e subtração
- Atribuição com um ou dois operandos (a à b + c)
- Um símbolo de pontuação ao final de cada comando

Sintaxe Básica:

*Observação: para todos os valores esperados, como <#start, var, #>, safe e show, poderão ser digitados tanto na forma maiúscula quanto na forma minúscula que o compilador irá aceitar, pois foi feito o processo de transformação caso entre com valores maiúsculos. Outro detalhe é em cada linha para o compilador saber diferenciar os tokens é necessário espaço entre eles.

Início: para iniciar o código, é necessária a presença do símbolo "<#" seguido da palavra **start**, como mostra o exemplo a baixo:

<#start

Erro palavra-chave Início: caso o usuário digite o termo de forma errada ou esqueça de digitar, o compilador irá expor o erro causado, como mostrado a seguir:

IMAGEM ERRO <#start

Declaração de Variáveis: a declaração de variáveis é feita utilizando a notação "**var**", seguido de um dos dois tipos numéricos presentes, int e/ou float, como mostrado no exemplo a baixo:

Ex: var float: a, int: b;

Também pode ser adiciona valor no cabeçalho de variaveis:

Ex: var float: a = 7.0, int: b =7;

Erro declaração de variáveis: se o usuário não entrar com a notação var, o compilador irá alerta-lo sobre a falta do mesmo, e caso o usuário digite *varfloat: a=3.5;* (sem a presença do espaço entre o var e o tipo da variável) também irá dar erro. O erro que irá aparecer é o seguinte:

IMAGEM ERRO DECLARAÇÃO DE VARIÁVEL

Comando de Escrita: O comando responsável pela escrita é o comando “show”, que deve ser escrito na forma minúscula, como mostra o exemplo:

```
show(resultado que será mostrado na tela);
```

Comando de Leitura: O comando responsável pela leitura dos dados é o comando “safe”, que também deve ser escrito na forma minúscula, conforme demonstrado no exemplo:

```
safe(p1);
```

Símbolo de encerrar comando: para que o compilador entenda que um comando foi finalizado, ele espera encontrar o símbolo “;” como mostra o exemplo a seguir:

```
show(resultado);
```

Erro no símbolo de encerramento: caso o compilador não encontre o símbolo “;” no final de qualquer comando, o seguinte erro irá aparecer:

IMAGEM ERRO ;

Fim de programa: para ser feita a finalização do código, o compilador espera a presença do símbolo, como no exemplo abaixo:

```
#>
```

Erro de fim de programa: caso o usuário digite o termo de forma errada ou esqueça de digitar, o compilador irá expor o erro causado, como mostrado abaixo:

IMAGEM ERRO #>

