

UNIVERSIDADE DA FRONTEIRA SUL (UFFS)

LINGUAGEM INTERPRETER

DALTON LUIZ PIMMEL

Chapecó 2015

Apresentação da Linguagem de Programação Interpreter

Interpreter é uma linguagem que pode - se gerar vários modelos programas. A linguagem possui três tipos variáveis, possui laços, condições, leitura de informação do teclado, impressão de informações na tela, cálculos matemáticos e lógicos, atribuição de variáveis de mesmo tipo, ou da mesma natureza, sendo que o tipo string aceita qualquer valor, pode-se haver espaços nas linhas, e pode-se usar comentários em qualquer parte do código.

SINTAXE

Cada programa deve estar dentro de um escopo principal, declaração de variáveis pode ser com atribuição ou sem, os laços e as condições tem que tem dois valores, podendo ser numérico ou uma variável, desde que esteja declarada e um operador lógico entre eles, retornando verdadeiro ou falso, pode haver linhas em branco durante o código, os comentários deve-se iniciar com duas barras, podendo estar em qualquer parte da linha. Pode - se utilizar laços dentro laços, condições dentro de condição.

ESCOPO PRINCIPAL

O programa deve esta dentro do escopo principal:

Inicio programa()

Comando

Comando

Fim programa

DECLARAÇÃO DE VARIÁVEIS E ATRIBUIÇÃO

Ex:

Int a = 10 // declaração do tipo inteiro com atribuição.

Int b // declaração de variável sem atribuição

Double c = 3.3 // declaração de ponto flutuante com atribuição

Double d // sem atribuição

String e = 'teste' // declaração de string com atribuição.

String f // sem atribuição.

Atribuição de variáveis:

A = b , é possível atribuição entre int e Double ou inverso, sendo que quem faz a conversão é sempre a primeira variável. As variáveis int e Double não aceitam variáveis do tipo string, já há string aceita qualquer valor.

A = 1 + b , atribuição de variáveis com operadores.

Str = s , atribuição de strings, neste caso substitui o valor antigo.

Str = + 'teste' , atribuição de string somente acrescentando teste, sem apagar o conteúdo anterior, podendo também usar uma variável de qualquer tipo.

a-- // diminui o valor de uma variável mesma coisa a = a - 1

a++ // aumento em 1 o valor da variável

CONDIÇÃO SE

Exemplo de utilização da condição se:

Operadores lógicos que podem ser utilizados:

➤ Maior que

< menor que

== igual a

!= diferente de

Inicio programa()

Se 1 > 0

Comando

Comando

Fim se

Fim programa

Exemplo com variáveis e a condição senao, que se o se der falso cai na condição se senao.

Inicio programa()

Int a = 10

Se a > 0

Comando

Fim se

Senao

Comando

Fim senao

Fim programa

UTILIZAÇÃO DE LAÇO

Exemplo de utilização de laço:

Inicio programa()

 Enquanto 1 > 0

 Comando

 Comando

 Fim enquanto

Fim programa()

IMPRESSÃO DE DADOS NA TELA

Exemplo de impressões de informações na tela:

Inicio programa()

 Imprime 'teste' // impressão de uma string

 Imprime 1 // impressão de um numero

 Imprime a // impressão de uma variável

 Imprime a + b // impressão de variáveis e operadores

 Imprime 1 + 3 // impressão de números e operadores

Fim programa

COMENTÁRIOS

Exemplo de utilização de comentários:

// teste com comentários

Inicio programa()

Int a = 5 // pode-se utilizar comentários em qualquer parte do código.

// se o comentário for no início da linha ignora toda ela.

Fim programa

CALCÚLOS

Pode - se usar, multiplicação, divisão, subtração, soma, além do cálculo da raiz quadrada, e a função mod.

Exemplos:

A = a + 3 // soma

A = a * 1 // multiplicação

A = a - 1 // subtração

A = a / 2 // soma

A = # 25 // calcula raiz quadrada, podendo usar-se variável no valor.

Para utilizar o mod, é preciso estar dentro de uma condição, podendo ser com variáveis ou com números:

Se 2 % 2 == 0

Imprime 'par'

Fim se

LER INFORMAÇÕES DO TECLADO

Exemplo para ler informações do teclado e armazenar em uma variável.

Inicio programa()

Int a

String s

Leia a // ira ler uma informação do teclado do int

Leia s // ira ler uma string

Fim programa

Se for digitado um valor que não seja do tipo da variável ou da mesma natureza, ocasionara em um erro.

ESPAÇAMENTOS

Exemplos de espaçamentos:

Inicio programa()

Int a = 10

Se 1 > 0

Comando

Fim se

Fim programa

FUNÇÃO BREAK E CONTINUE

Deve-se utilizar somente dentro dos laços.

Break: termina a laço que esta rodando.

Continue: retorna para o inicio do laço que esta rodando.

Ex:

Inicio programa()

 Enquanto 1 > 0

 Comando

 Break

 Fim enquanto

 Enquanto 1 > 0

 comando

 continue

 fim enquanto

fim programa

PALAVRAS RESERVADAS

Inicio programa() , Fim programa

Se, senão, fim se, fim senão

Enquanto, fim enquanto

Imprime

Leia

Int, Double, string

Continue

Break