Procedimento e função

Professor Edson Maia

Procedimento

Tem que ser declarado entre o final da seção de variáveis e o inicio do programa principal

é um subprograma que **não retorna nenhum valor**. Corresponde ao *procedure*.

Sintaxe

```
procedimento nome_procedimento(parâmetros)
// Seção de variáveis internas
inicio
// comandos do procedimento
    ações a serem feitas no procedimento
fimprocedimento
```

Objetivo: reutilizar códigos que se repetem.
Modularizar nossos programas.

Procedimento sem parâmetro

Parâmetros são variáveis que declaramos dentro dos parênteses. Eles deverão ser usadas obrigatoriamente no procedimento.

Exemplo 1

```
procedimento msg()

    inicio
        escreval("Olá Mundo!")
fimprocedimento
```

Procedimento sem parâmetro

Parâmetros são variáveis que declaramos dentro dos parênteses. Eles deverão ser usadas obrigatoriamente no procedimento.

Exemplo 2

```
procedimento sucesso()

    inicio
        escreval("Tarefa feita com sucesso!")
fimprocedimento
```

Evocar Procedimento sem parâmetro

No programa principal digite:

```
msg()
sucesso()
```

Procedimento com parâmetro

Parâmetros são variáveis que declaramos dentro dos parênteses. Eles deverão ser usadas obrigatoriamente no procedimento.

Exemplo 3

```
procedimento msg2 (msg : caractere)

    inicio
        escreval(msg)
fimprocedimento
```

Procedimento com parâmetros

Procedimento para somar 2 números inteiros.

Exemplo

```
procedimento somar2inteiros (n1, n2 : inteiro)
// Seção de variáveis internas
     resultado : inteiro

     inicio
// comandos do procedimento
          resultado <- n1 + n2
          escreval("Soma de " , n1 , " +" ,  n2, " =" , resultado)
fimprocedimento
```

Evocar Procedimento com parâmetros

No programa principal digite:

```
msg2("Curso de Algoritmos")
somar2inteiros(3,2)
```

Exemplo prático procedimentos

```
algoritmo "12-procedimento"
// Função : Criar procedimento
// Autor  : Edson Maia

var

// 1º PROCEDIMENTO SEM PARÂMETROS
procedimento msg()
   inicio
       escreval("Olá Mundo")
fimprocedimento

// 1º PROCEDIMENTO SEM PARÂMETROS
procedimento sucesso()
inicio
    escreval("Tarefa feita com sucesso!")
fimprocedimento


```

```

// 2º PROCEDIMENTO COM PARÂMETRO

procedimento msg2(msg : caractere)
   inicio
       escreval(msg)
fimprocedimento

// 3º PROCEDIMENTO COM PARÂMETROS

procedimento somar2inteiros(n1, n2 : inteiro)
// Seção de declarações internas
    var resultado : inteiro

    inicio
        resultado <- n1 + n2
        escreval("Soma de " , n1 , " +" ,  n2, " =" , resultado)
fimprocedimento

```

Função

Tem que ser declarada entre o final da seção de variáveis e o inicio do programa principal

é um subprograma que **retorna um valor**. Usa a palavra reservada **retorne**. Corresponde a *function*.

Sintaxe

```
funcao nome_funcao(parâmetros) : tipo retorno
// Seção de variáveis internas
inicio
// comandos da função
    ações a serem feitas na função
    retorne <conteúdo a ser retornado>
fimfuncao
```

Objetivo: reutilizar códigos que se repetem.
Modularizar nossos programas.

Exemplo prático funções

```
38 // FUNÇÕES
39
40 // 1° FUNÇÃO SEM PARÂMETROS
41 funcao mostra_msg() : caractere
42    inicio
43        retorne "Mensagem exibida por uma função"
44 fimfuncao
45
46 // 2° FUNÇÃO COM PARÂMETRO
47 funcao msg3(msg : caractere) : caractere
48    inicio
49        retorne msg
50 fimfuncao
51
52 // 3° FUNÇÃO COM PARÂMETROS
53 funcao somar2inteiros2(n1, n2 : inteiro) : inteiro
54 // Seção de declarações internas
55     var resultado : inteiro
56
57     inicio
58         resultado <- n1 + n2
59         retorne resultado
60 fimfuncao
```

```
61
62 inicio
63
64 // EVOCAR, CHAMAR OU EXECUTAR O PROCEDIMENTO
65 msg()
66
67 sucesso()
68
69 msg2("Curso de Algoritmos")
70
71 somar2inteiros(3,2)
72
73 escreval(mostra_msg())
74 escreval(msg3("Mensgem personalizada com Função"))
75 escreval(somar2inteiros2(4,6))
76
77 fimalgoritmo
```

Resumindo

Procedimentos são estruturas usadas para agruparmos códigos que se repetem no nosso programa. Ou funcionalidades úteis.

Servem para modularizarmos nossos programas.

Procedimentos ou *procedure* **não tem retorno**.

Eles podem ou não ter parâmetros.

Função é igual ao procedimento, a diferença é que **tem retorno**, usam a palavra **retorne** ou *return*. A estrutura é finalidade das funções é a “mesma”.

Parâmetros ou **argumentos são variáveis** que podemos **usar nos procedimentos** e **funções**.