

Subalgoritmos

Roteiro

- ❑ Subalgoritmos
 - ❑ Funções
 - ❑ Procedimentos
 - ❑ Funções recursivas

Subalgoritmos

- ❑ Trechos de algoritmos que realizam cálculos ou tarefas. Vantagens
 - ❑ Reduzem o tamanho do algoritmo.
 - ❑ Facilitam a compreensão e visualização do algoritmo.
 - ❑ São declarados no início do algoritmo e podem ser chamados em qualquer ponto após sua declaração.
 - ❑ Eles podem ser Funções que retorna algum valor ou Procedimento (Subrotina) que não retorna nada.

Funções

- ❑ Retornam um valor
- ❑ Funções Predefinidas do Visualg

| FUNÇÃO | DESCRIÇÃO |
|---|--|
| Abs (valor : real) : real | Valor absoluto |
| Arccos (valor : real) : real | Arco cosseno |
| Arcsen (valor : real) : real | Arco seno |
| Arctan (valor : real) : real | Arco tangente |
| Asc (s : caracter) : inteiro | Retorna o código ASCII |
| Compr (c : caracter) : inteiro | Retorna a dimensão do caractere |
| Copia (c : caracter , posini, posfin : inteiro) : caracter | Copia um determinado trecho do caractere |
| Cos (valor : real) : real | Cosseno |
| Cotan (valor : real) : real | Co-tangente |
| Exp (<base>, <expoente>) | Potenciação |

Funções

☐ Funções Predefinidas do Visualg

| | |
|---|---|
| Grauprad (valor : real) : real | Converte grau para radiano |
| Int (valor : real) : inteiro | Converte o valor em inteiro |
| Log (valor : real) : real | Logaritmo de base 10 |
| Logn (valor : real) : real | Logaritmo natural (ln) |
| Maiusc (c : caracter) : caracter | Converte em Maiúscula |
| Minusc (c : caracter) : caracter | Converte em Minúscula |
| Numpcarac (n : inteiro ou real) : caracter | Converte um numero inteiro ou real para caractere |
| Pi : real | Valor Pi |
| Pos (subc, c : caracter) : inteiro | Retorna a posição do caractere. |

Funções

☐ Funções Predefinidas do Visualg

| | |
|---|--|
| Quad (valor : real) : real | Elevado quadrado |
| Radpgrau (valor : real) : real | Converte Radiano para grau. |
| Raizq (valor : real) : real | Raiz quadrada |
| Rand : real | Gerador de números aleatórios entre 0 e 1 |
| Randi (limite : inteiro) : inteiro | Gerador de números inteiros aleatórios com um limite determinado |
| Sen (valor : real) : real | Seno |
| Tan (valor : real) : real | Tangente |

Funções

□ Ex.:

```
algoritmo "retorna sobrenome"  
// Seção de Declarações  
var  
nome, sobrenome : Caractere  
quant_caracteres, local_espcao : INTEIRO  
inicio  
// Seção de Comandos  
nome <- "Bruno Tonet"  
quant_caracteres <- Compr (nome)  
local_espcao <- POS (" ", nome)  
sobrenome <- Copia (nome, local_espcao + 1 , quant_caracteres)  
Escreva("Seu sobrenome é ", sobrenome)  
fimalgoritmo
```

Funções

□ Ex.:

```
algoritmo "retorna número inteiro"  
// Seção de Declarações  
var  
valorReal : REAL  
valorInteiro : INTEIRO  
inicio  
// Seção de Comandos  
valorReal <- 5.87978098980980989089898  
valorInteiro <- INT(valorReal)  
Escreva("Valor inteiro ", valorInteiro)  
fimalgoritmo
```

Criação de Funções

```
Algoritmo "<nome do algoritmo>"  
var  
    <declaração de variáveis globais>  
<definição da função>  
inicio  
    < lista de comandos>  
finalgoritmo
```

Sintaxe da Função

```
funcao <identificador> ([var]<parâmetros>) <tipo de retorno>  
var  
<declaração de variáveis locais>  
inicio  
<lista de comandos>  
retorne <variável de retorno>  
fimfuncao
```

□ Ex.:

algoritmo "semnome"

var

Valor_1, Valor_2, soma: real

FUNCAO FSoma(Recebe_valor1, Recebe_valor2: Real):Real

var

total : real

Inicio

total<-Recebe_valor1+Recebe_valor2

retorne total

fimfuncao

inicio

escreva ("Valor_1 : ")

leia(Valor_1)

escreva ("Valor_2 : ")

leia(Valor_2)

soma<-FSoma(Valor_1, Valor_2)

escreva("Soma das variáveis é ", soma)

fimalgoritmo

Procedimento (Sub_rotinas)

- ❑ Não retorna nada, apenas realiza uma tarefa

Sintaxe Procedimento:

```
procedimento <identificador> ([var]<parâmetros>)  
  
var  
  
<declaração de variáveis locais>  
  
inicio  
  
<lista de comandos>  
fimprocedimento
```

□ Ex.:

```
algoritmo "semnome"
var
A,B,C,D,CONT,AUX:Inteiro
```

```
Procedimento TROCA(var x, y: inteiro)
```

```
var
Aux : inteiro
```

```
INICIO
```

```
Aux <- x
```

```
x <- y
```

```
y <- Aux
```

```
FIMProcedimento
```

```
INICIO
```

```
LEIA (A,B,C,D)
```

```
Enquanto NAO((A<=B) e (B<=C) e (C<=D)) faça
```

```
  se (D<C) então
```

```
    TROCA(D,C)
```

```
  FIMSE
```

```
  SE C<B ENTÃO
```

```
    TROCA(C,B)
```

```
  FIMSE
```

```
  SE B<A ENTÃO
```

```
    TROCA(A,B)
```

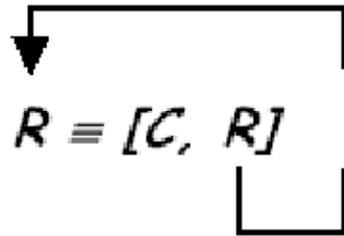
```
  FIMSE
```

```
FIMENQUANTO
```

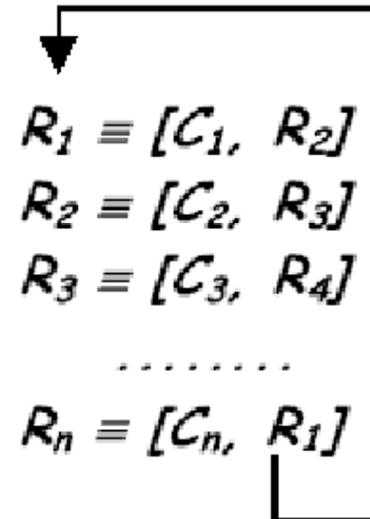
```
escreva (A," ",B," ",C," ",D)
```

```
fimalgoritmo
```

Funções definidas recursivamente



Recursão direta



Recursão indireta

Funções definidas recursivamente

- ❑ Ex. Cálculo do fatorial:
 - ❑ A definição de fatorial é:
 - ❑ $F(n) = 1$ se $n = 0$ ou $n = 1$;
 - ❑ $F(n) = n.F(n-1)$, se $n > 1$
 - ❑ onde n é um numero inteiro positivo.
 - ❑ Note que:
 - ❑ $n! = n . (n-1)!$

Funções definidas recursivamente

□ Assim, fatorial de 4 é

□ $F(4) = 4.F(4-1)$ //Caso de geral

□ $F(3) = 3.F(3-1)$ //Caso de geral

□ $F(2) = 2.F(2-1)$ //Caso de geral

□ $F(1) = 1.F(1-1)$ //Caso de geral

□ $F(0) = 1$ //Caso de base

□ $F(1) = 1.1$

□ $F(2) = 2.1$

□ $F(3) = 3.2$

□ $F(4) = 4.6$

Funções definidas recursivamente

- ❑ caso de base (base de recursão)
 - ❑ o problema é resolvido diretamente (sem chamada recursiva)
- ❑ caso geral
 - ❑ o problema é resolvido com uma chamada recursiva
- ❑ caso geral
 - ❑ o tamanho do problema é menor a cada chamada

Funções definidas recursivamente

□ Forma:

se "condicao para o caso de base" entao
 resolucao direta para o caso de base
senao
 uma ou mais chamadas recursivas
fimse

Funções definidas recursivamente

□ Ex. Cálculo do fatorial:

$$F(4) = 4.F(4-1)$$

$$F(3) = 3.F(3-1)$$

$$F(2) = 2.F(2-1)$$

$$F(1) = 1.F(1-1)$$

$$F(0) = 1 \text{ ----- Caso Base}$$

$$F(1) = 1.1$$

$$F(2) = 2.1$$

$$F(3) = 3.2$$

$$F(4) = 4.6$$

□ Ex. Cálculo do fatorial:

algoritmo "fatorial"

var

A, Fatorial: Inteiro

Funcao Fat (x: Inteiro): Inteiro

inicio

se $x=0$ entao

retorne 1

senao

retorne $x * \text{Fat}(x-1)$

Fimse

FimFuncao

inicio

Leia (A)

Fatorial \leftarrow Fat (A)

Escreva ("Fatorial ", A, " é ", Fatorial)

FimAlgoritmo