

Funções



PROFESSOR: DIEGO RICARDO KROHL
`diego.krohl@ifc-videira.edu.br`

Funções



- A complexidade dos algoritmos está intimamente ligada à da aplicação a que se destina;
- Em geral, problemas complicados exigem algoritmos extensos para sua solução;
- É possível dividir um problema grande em problemas menores e de solução mais simples.

Funções



- Podemos codificar a solução de problemas simples utilizando os chamados sub algoritmos ou sub rotinas;
- Sub algoritmo é um nome dado a um trecho de um algoritmo mais complexo e que, em geral, encerra em si mesmo um pedaço da solução de um problema maior a que ele está subordinado.

Funções - Importância



- Subdivisão de algoritmos complexos, facilitando o seu entendimento;
- Estruturação de algoritmos, facilitando principalmente a detecção de erros e a documentação de sistemas;
- Modularização de sistemas, que facilita a manutenção de softwares e a reutilização de sub algoritmos já implementados.

Funções - Sintaxe



```
funcao AREA (varX, varY :real) : real
```

```
var
```

```
varAux :real
```

```
inicio
```

```
varAux <- varX * varY
```

```
retorne varAux
```

```
fimfuncao
```

Funções - Exemplo



```
algoritmo "Função soma"

var
  a,b, soma: inteiro //declaração das variáveis globais

funcao fsoma(a1, b1: inteiro):inteiro //declaração da função
var
  resultado : inteiro //declaração de variáveis locais
Inicio
  resultado<-a1+b1 //instruções
  retorne resultado //valor de retorno
fimfuncao

inicio
  Escreval ("Digite o primeiro Número")
  leia (a)
  Escreval ("Digite o segundo Número")
  leia (b)
  soma <- fsoma(a,b) //chamada da função
  escreva ("soma: ", soma)      //Corpo do algoritmo principal

fimalgoritmo
```

Procedimentos



- São sub programas que se distinguem das funções pelo fato de não apresentarem qualquer valor de retorno;
- Podem apresentar ou não uma lista de parâmetros que definem a forma como o procedimento irá ser executado;

Procedimentos - Sintaxe



```
procedimento <nome do procedimento> (<parâmetros>)  
var  
<declaração das variáveis locais>  
inicio  
<lista de comandos>  
fimprocedimento
```


Procedimentos – Exemplo 1 (Dobro)



```
algoritmo "Procedimento"  
  
var  
numero: inteiro  
  
procedimento Dobro (valor: inteiro)  
var  
total: inteiro  
inicio  
total<-valor*2  
escreval("O dobro de",valor," é:",total)  
fimprocedimento  
  
inicio  
    escreval ("Digite um número:")  
    leia (numero)  
    Dobro(numero)  
fimalgoritmo
```

Procedimentos – Exemplo 2 (Troca)



```
algoritmo "TrocaValores"  
var  
x: inteiro  
y: inteiro  
  
procedimento troca (x, y: inteiro)  
var  
aux: inteiro  
inicio  
    aux <- x  
    x <- y  
    y <- aux  
    Escreval ("Os valores de x e y depois da troca são:", x, " ", y)  
fimprocedimento  
  
inicio  
    x <- 5  
    y <- 8  
    Escreval ("Os valores de x e y antes da troca são:", x, " ", y)  
    troca (x, y)  
fimalgoritmo
```