

Algoritmos e Programação

AULA 17: Subalgoritmos

UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CIÊNCIA DA COMPUTAÇÃO



Subalgoritmos

- São trechos de algoritmos que efetuam um ou mais cálculos determinados.
- Ao invés de escrever-se um algoritmo grande, escrevem-se vários algoritmos menores
 - Em conjunto, resolvem o problema proposto.
- É conveniente utilizá-los quando uma determinada tarefa é efetuada em diversos lugares no mesmo algoritmo.
 - Ao invés de escrever-se um trecho diversas vezes, escrevemos um subalgoritmo e o chamamos diversas vezes.

Subalgoritmos

- Vantagens:
 - Eles 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.

Subalgoritmos

- Importância de subalgoritmos:
 - subdivisão de algoritmos complexos, melhorando a compreensão do algoritmo
 - estruturação de algoritmos, permitindo que se detecte falhas com maior facilidade;
 - modularização de sistemas, que facilita a manutenção de softwares e a reutilização de subalgoritmos desenvolvidos em outros algoritmos.

Subalgoritmos

- Subalgoritmos podem ser:
 - Funções que resultam em um valor (são chamadas em expressões)
 - Procedimentos (Subrotinas) que não resultam em um valor (são chamados como comandos).

Declaração

- São definidos após as declarações das variáveis e antes do corpo principal do algoritmo

```
Algoritmo
```

```
{Definições de tipos}
```

```
{Declarações de variáveis}
```

```
{Definições de subalgoritmos}
```

```
{corpo do algoritmo}
```

```
fim_algoritmo
```

Componentes

- Parâmetros (ou argumentos) são dados
 - enviados ao subalgoritmo para serem processados;
 - recebidos do subalgoritmo como resultados do processamento.
 - Podem ser:
 - Parâmetros Formais: usados na definição do subalgoritmo
 - Parâmetros Reais: usados na chamada do subalgoritmo

Componentes

- Variáveis locais
 - variáveis declaradas dentro do subalgoritmo
 - só podem ser acessadas dentro do subalgoritmo
- Variáveis globais
 - variáveis declaradas fora do subalgoritmo, no algoritmo principal
 - podem também ser acessadas pelo subalgoritmo

Exemplo

Algoritmo

Declare a,b numérico

sub-rotina y(c,d numéricos)

Declare e,f numérico

{ comandos de y }

fim_sub-rotina

função numérico z(e numérico)

Declare f numérico

{ comandos de z }

fim_função

b:=2

y(1,b)

a := z(10)

fim_algoritmo

Exemplo

Algoritmo

```

Declare a,b numérico
sub-rotina y(c,d numéricos)
  Declare e,f numérico
  { comandos de y }
fim_sub-rotina
função numérico z(e numérico)
  Declare f numérico
  { comandos de z }
fim_função
b:=2
y(1,b)
a := z(10)
fim_algoritmo

```

- Onde:
 - $a; b$ – variáveis globais (acessadas pelo algoritmo e pelos dois subalgoritmos);
 - $e; f$ – variáveis locais a y (acessadas apenas pelo subalgoritmo y);
 - $c; d$ – parâmetros formais de y (usados na hora da chamada de y) declarados como numéricos;
 - f – variável local a z (acessada somente pelo subalgoritmo z , não possuindo qualquer relação com a outra variável f (local a y);
 - e – parâmetro formal de z , declarado como numérico;
 - $y(1; b)$ – chamada de y com parâmetros reais 1 e b (1 será atribuído a c e b será atribuído a d);
 - $z(10)$ – chamada de z .

Exemplo

Algoritmo

Declare `a, b` numérico

sub-rotina `y(c, d numéricos)`

Declare `e, f` numérico

{ comandos de `y` }

fim_sub-rotina

função numérico `z(e numérico)`

Declare `f` numérico

{ comandos de `z` }

fim_função

`b := 2`

`y(1, b)`

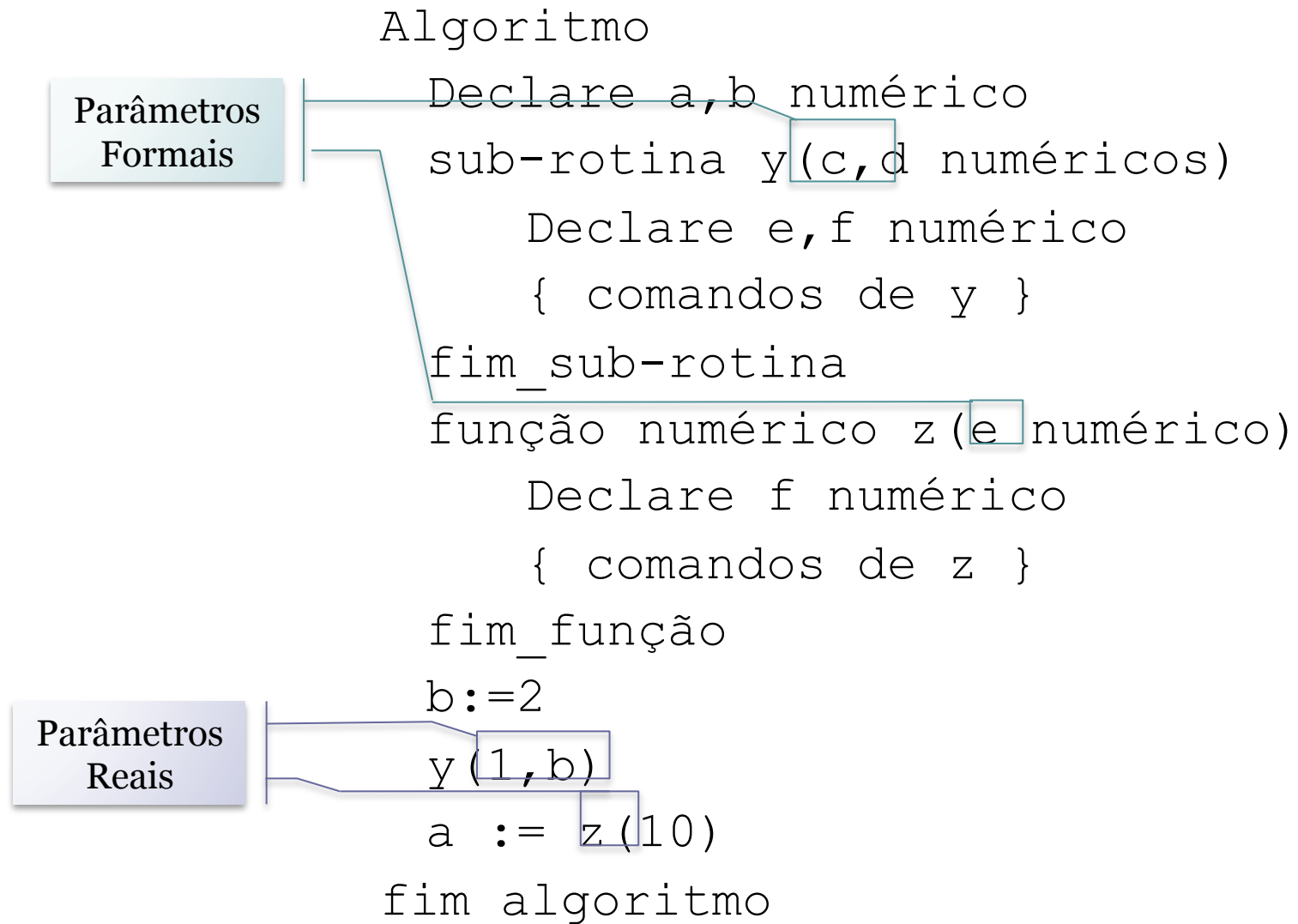
`a := z(10)`

fim_algoritmo

Globais

Locais

Exemplo



Passagem de Parâmetros

- Existem diferentes tipos:
 - **Passagem por valor** (só entrada): na chamada os valores dos parâmetros reais são copiados para os parâmetros formais para serem usados dentro do sub-algoritmo sem que se modifiquem os valores armazenados nos parâmetros reais.
 - **Passagem copia-restaura** (entrada e saída): na chamada os valores dos parâmetros reais são copiados para os parâmetros formais e usados, ao final os valores dos parâmetros formais são copiados de volta nos parâmetros reais
 - os parâmetros reais deverão ser variáveis

Passagem de Parâmetros

- O tipo copia-restaura é indicado, colocando-se a palavra reservada CR antes dos parâmetros deste tipo.

- Exemplo:

```
Sub-rotina teste(CR a numérico; b,c literal)
    {Declarações de teste}
    {comandos de teste}
fim_sub-rotina
```

```
teste(x, 'ok', y)
```

Sub-rotinas



Sub-rotinas

- Sub-rotinas são subalgoritmos que não resultam em um valor (são chamados como comandos)
- Podem retornar valores apenas pelos argumentos (caso sejam definidos como copia-restaura).

Declaração

```
sub-rotina <nome>(<lista de parâmetros formais>)  
  { declarações locais }  
  { comandos }  
fim_sub-rotina
```

- Onde:
 - subrotina e fim_subrotina :são palavras reservadas para demarcar o início e o fim da sub-rotina.
 - <nome>: é o identificador da sub-rotina que será utilizado na chamada da sub-rotina
 - <lista de parâmetros formais>: descrição dos parâmetros usados na definição do subalgoritmo e de seus tipos

Chamada de subrotina

`<nome>(<lista de parâmetros reais>)`

- A chamada de uma sub-rotina é feita através de seu nome (identificador) e a indicação dos parâmetros reais a serem usados na sub-rotina
- Deve-se ter o mesmo número de parâmetros reais que os parâmetros formais utilizados na declaração da sub-rotina
- Os parâmetros formais e reais devem ser do mesmo tipo

Sub-rotinas

- Observe que:
 - podemos definir uma sub-rotina sem parâmetros formais (para qual não precisamos passar nenhum parâmetro na chamada)
 - é possível definir um sub-algoritmo dentro de outro (aninhamento). Sub-algoritmos declarados dentro de outros são locais a estes (desconhecidos fora).

Exemplo 1

Algoritmo

Declare Nota1, Nota 2, Media numérico

sub-rotina LerNotas

 Escreva "Digite a primeira nota", \n

 Leia Nota1

 Escreva "Digite a segunda nota", \n

 Leia Nota2

fim_sub-rotina

LerNotas

Media:= (Nota1 + Nota2)/2

Escreva "Média:", Media

fim_algoritmo

Exercício 1

- Faça um algoritmo que represente uma calculadora. O algoritmo deve apresentar um menu de opções pra cada uma das quatro operações básicas:
 - 1 - Adição
 - 2 - Subtração
 - 3 - Multiplicação
 - 4 - Divisão
- Cada operação deve ser realizada em uma sub-rotina sem parâmetros.

Exercício 1 - Solução

Algoritmo

Declare opcao numerico

sub-rotina Adicao

Declare X, A, B numérico

Escreva “Entre com o primeiro valor”,\n

Leia A

Escreva “Entre com o segundo valor”,\n

Leia B

$X:=A+B$

Escreva “Adição:”, X

fim_sub-rotina

...

Exercício 1 - Solução

...

sub-rotina Subtracao

Declare X, A, B numérico

Escreva “Entre com o primeiro valor”, \n

Leia A

Escreva “Entre com o segundo valor”, \n

Leia B

$X := A - B$

Escreva “Subtração:”, X

fim_sub-rotina

...

Exercício 1 - Solução

...

sub-rotina Multiplicacao

Declare X, A, B numérico

Escreva “Entre com o primeiro valor”, \n

Leia A

Escreva “Entre com o segundo valor”, \n

Leia B

X:=A*B

Escreva “Multiplicação:”, X

fim_sub-rotina

...

Exercício 1 - Solução

...

sub-rotina Divisao

Declare X, A, B numérico

Escreva “Entre com o primeiro valor”,\n

Leia A

Escreva “Entre com o segundo valor”,\n

Leia B

$X:=A/B$

Escreva “Divisão:”, X

fim_sub-rotina

...

Exercício 1 - Solução

...

Repita

```
Escreva ("Menu de Opções: ")
Escreva ("1 - Adição")
Escreva ("2 - Subtração")
Escreva ("3 - Multiplicação")
Escreva ("4 - Divisão")
Escreva ("5 - Sair")
Escreva ("Opção: ")
```

Leia opcao

Se opção = 1

então Adicao

fim_se

Se opção = 2

então Subtracao

fim_se

Se opção = 3

então Multiplicacao

fim_se

Se opção = 4

então Divisao

fim_se

até opção = 5

fim algoritmo

Exemplo 2

Algoritmo

```
Declare a,b numérico
```

```
sub-rotina troca(x,y numérico)
```

```
    Declare aux numérico
```

```
    aux:=x
```

```
    x:=y
```

```
    y:=aux
```

```
fim_sub-rotina
```

```
a:=10
```

```
b:=5
```

```
troca(a,b)
```

```
fim_algoritmo
```

- **Funciona? Porque?**

Exemplo 2

Algoritmo

```
Declare a,b numérico
```

```
sub-rotina troca (CR x,y numérico)
```

```
    Declare aux numérico
```

```
    aux:=x
```

```
    x:=y
```

```
    y:=aux
```

```
fim_sub-rotina
```

```
a:=10
```

```
b:=5
```

```
troca(a,b)
```

```
fim_algoritmo
```

- Deve-se definir a passagem de parâmetro copia-restaura

Exercício 2

- Faça um algoritmo para ler três valores a , b e c e colocá-los em ordem crescente, utilizando uma sub-rotina para que seja realizada a ordenação.

Exercício 2 - Solução

Algoritmo

Declare a,b,c numérico

sub-rotina troca(CR x,y numérico)

Declare aux numérico

aux:=x

x:=y

y:=aux

fim_sub-rotina

Escreva "Digite três valores", \n

Leia a,b,c

Se a>b

então troca(a,b)

fim_se

Se a>c

então troca(a,c)

fim_se

Se b>c

então troca(b,c)

fim_se

Escreva a,b,c

fim_algoritmo