

Fundamentos de Programação 1

Estrutura de decisão simples

Slides 6.

Prof. SIMÃO

Jean Marcelo SIMÃO

Estrutura de Decisão ou Seleção Simples

se (condição for verdadeira) **entao**

conjunto de comandos

fimse

Exemplo de Algoritmo

algoritmo "Analisar se um número inteiro é maior que 100"

Var

num : **Inteiro**

// Parte Principal

inicio

escreva ("Digite um número:")

leia (num)

se (num > 100) **entao**

escreva ("Número maior que 100")

fimse

se (num < 100) **entao**

escreva ("Número menor que 100")

fimse

se (num = 100) **entao**

escreva ("Número igual a 100")

fimse

fimalgoritmo

Operadores relacionais

algoritmo "Analisar se um número inteiro é maior que 100"

Var

num : **Inteiro**

// *Parte Principal*

inicio

escreva ("Digite um número:")

leia (num)

se (num > 100) **entao**

escreva ("Número maior que 100")

fimse

se (num < 100) **entao**

escreva ("Número menor que 100")

fimse

se (num = 100) **entao**

escreva ("Número igual a 100")

fimse

fimalgoritmo

Operadores relacionais:

>, >=, <, <=, =, <>

Obs.: <> é o diferente

Exercício

Elaborar um algoritmo que analisa se um número inteiro é maior que 100 e, se for maior, quantas vezes ele é maior que 100.

Por exemplo, o número 249 é 2,49 vezes maior que 100.

Solução do Exercício

algoritmo "Analisar se um número inteiro é maior que 100 é quantas vezes é maior".

var

num : **Inteiro**

nvezes : **Real**

// Parte Principal

inicio

escreva (" Digite um número inteiro: ")

leia (num)

se (num > 100) **entao**

escreval (" Número maior que 100.")

 nvezes <- num / 100

escreval (" O número é maior que cem ", nvezes, " vezes.")

fimse

se (num <= 100) **entao**

escreval (" Número menor ou igual a 100. ")

fimse

fimalgoritmo

Outra solução

algoritmo "Analise se um número inteiro é maior que 100 é quantas vezes é maior".

var

num : **Inteiro**

// Parte Principal

inicio

escreva (" Digite um número inteiro: ")

leia (num)

se (num > 100) **entao**

escreval (" Número maior que 100.")

escreval (" O número é maior que cem", num / 100, " vezes.")

fimse

se (num <= 100) **entao**

escreval ("Número menor ou igual a 100.")

fimse

fimalgoritmo

Conjunções e disjunções

Cada comando *se entao* pode avaliar várias expressões lógicas por meio de conectivos lógicos, em especial por meio das conjunções (*e*) e disjunções (*ou*).

Conjunções:

se ((expressão 1) e (expressão 2) então

... ..

fim-se

Disjunções:

se ((expressão A) ou (expressão B)) então

... ..

fim-se

Exercício

Elaborar um algoritmo que permita ao usuário somar dois ou três números inteiros, segundo seu desejo.

Uma primeira solução

```
algoritmo "Soma de dois ou três números"  
var  
  prim_num, seg_num, terc_num, soma, opcao :Inteiro  
  
// Parte Principal  
inicio  
  
  escreval ( " Digite 2 para somar 2 números." )  
  escreval ( " Digite 3 para somar 3 números." )  
  escreval ( " Qual é a sua opção?" )  
  leia ( opcao )  
  
  se ( opcao = 2 ) entao  
  
    escreval ( " Digite o primeiro número: " )  
    leia ( prim_num )  
  
    escreval ( " Digite o segundo número: " )  
    leia ( seg_num )  
  
    soma <- prim_num + seg_num  
    escreval ( " A soma é : ", soma )  
  
  fimse
```

```
    se ( opcao = 3 ) entao  
  
      escreval ( "Digite o primeiro número: " )  
      leia ( prim_num )  
  
      escreval ( " Digite o segundo número: " )  
      leia ( seg_num )  
  
      escreval ( " Digite o terceiro número: " )  
      leia ( terc_num )  
  
      soma <- prim_num + seg_num + terc_num  
      escreval ( " A soma é : ", soma )  
  
    fimse  
  
    se ( ( opcao <> 2 ) e ( opcao <> 3 ) ) entao  
  
      escreval ( " Opção inválida" )  
  
    fimse  
  
fimalgoritmo
```

Uma segunda versão

algoritmo "soma de dois ou três números"

var

prim_num, seg_num, terc_num, soma : **inteiro**
opcao : **caracter**

// Parte Principal

inicio

escreval ("Digite A para somar 2 números.")

escreval ("Digite B para somar 3 números.")

escreval ("Qual é a sua opção?")

leia (opcao)

se (opcao = "A") **entao**

escreval ("Digite o primeiro número: ")

leia (prim_num)

escreval ("Digite o segundo número: ")

leia (seg_num)

soma <- prim_num + seg_num

escreval ("A soma é :", soma)

fimse

se (opcao = "B") **entao**

escreval ("Digite o primeiro número: ")

leia (prim_num)

escreval ("Digite o segundo número: ")

leia (seg_num)

escreval ("Digite o terceiro número: ")

leia (terc_num)

soma <- prim_num + seg_num + terc_num

escreval (" A soma é :", soma)

fimse

se ((opcao <> "A") e (opcao <> "B")) **entao**

escreval (" Opção inválida ")

fimse

fimalgoritmo

Uma terceira versão

algoritmo "Soma de dois ou três números"

var

prim_num, seg_num, terc_num, soma : **inteiro**
opcao : **caracter**

// Parte Principal

inicio

escreval ("Digite **1** para somar 2 números.")

escreval ("Digite **2** para somar 3 números.")

escreval ("Qual é a sua opção?")

leia (opcao)

se (opcao = "**1**") **entao**

escreval ("Digite o primeiro número: ")

leia (prim_num)

escreval ("Digite o segundo número: ")

leia (seg_num)

soma <- prim_num + seg_num

escreval ("A soma é :", soma)

fimse

se (opcao = "**2**") **entao**

escreval (" Digite o primeiro número: ")

leia (prim_num)

escreval (" Digite o segundo número: ")

leia (seg_num)

escreval (" Digite o terceiro número: ")

leia (terc_num)

soma <- prim_num + seg_num + terc_num

escreval ("A soma é :", soma)

fimse

se ((opcao <> "**1**") e (opcao <> "**2**")) **entao**

escreval (" Opção inválida. ")

fimse

fimalgoritmo

Melhorando a solução

Seria possível uma solução melhor ?

Segunda solução

algoritmo “Soma de dois ou três números”

var

prim_num, seg_num : **Inteiro**

terc_num, soma, opcao : **Inteiro**

// Parte Principal

inicio

escreval (“*Digite 2 para somar 2 números.*”)

escreval (“*Digite 3 para somar 3 números.*”)

escreval (“*Qual é a sua opção?*”)

leia (opcao)

se ((opcao = 2) **ou** (opcao = 3)) **entao**

escreval (“*Digite o primeiro número:* ”)

leia (prim_num)

escreval (“*Digite o segundo número:* ”)

leia (seg_num)

soma ← prim_num + seg_num

se (opcao = 2) **entao**

escreval (“*A soma é .:*”, soma)

fimse

se (opcao = 3) **entao**

escreval (“*Digite o terceiro número:* ”)

leia (terc_num)

soma ← soma + terc_num

escreval (“*A soma é:*”, soma)

fimse

fimse

se ((opcao <> 2) **e** (opcao <> 3)) **entao**

escreval (“*Opção inválida*”)

fimse

fimalgoritmo

Outra solução ainda

```
algoritmo “Outra Solução”  
var  
    num1, num2, num3, result : Inteiro  
// Parte Principal  
inicio  
  
    escreval ( “Digite o primeiro número: ” )  
    leia ( num1 )  
  
    escreval ( “Digite o segundo número: ” )  
    leia ( num2 )  
  
    escreval ( “Digite o terceiro número: ” )  
    escreval ( “Obs. Para somar dois números, zere o terceiro!” )  
    leia ( num3 )  
  
    result ← num1 + num2 + num3  
  
    escreval ( “O resultado é .:”, result)  
  
fimalgoritmo
```

Exercícios.

Elaborar um algoritmo para o cálculo da soma, subtração, multiplicação ou divisão de dois números reais fornecidos pelo usuário, segundo sua opção.

Solução exercício.

algoritmo “Operações elementares sobre dois números cf. opção do usuário”

prim_num, seg_num : **real**
soma, sub, mul, div, opcao : **real**

// Parte Principal

inicio

escreval (“Operações sobre 2 números reais.”)
escreval (“Digite 1 para soma,”)
escreval (“ 2 para subtração, ”)
escreval (“ 3 para multiplicação ou”)
escreval (“ 4 para divisão.”)
leia (opcao)

se ((opcao = 1) **ou** (opcao = 2) **ou**
(opcao = 3) **ou** (opcao = 4)) **entao**

escreval (“Digite o primeiro número: ”)
leia (prim_num)

escreval (“Digite o segundo número: ”)
leia (seg_num)

se (opcao = 1) **então**
soma <- prim_num + seg_num
escreval (“O resultado da soma é: ”, soma)
fimse

se (opcao = 2) **entao**
sub <- prim_num - seg_num
escreval (“O resultado da subtração é: ”, sub)
fimse

se (opcao = 3) **entao**
mul <- prim_num * seg_num
escreval (“O resultado da multiplic. é: ”, mul)
fimse

se (opcao = 4) **entao**
div <- prim_num / seg_num
escreval (“O resultado da divisão é: ”, div)
fimse

fimse

se ((opcao <> 1) **e** (opcao <> 2) **e**
(opcao <> 3) **e** (opcao <> 4)) **entao**

escreval (“Opção inválida”)

fimse

fimalgoritmo

Solução exercício V2.

algoritmo “Operações elementares sobre dois números cf. opção do usuário V2”

var

prim_num, seg_num : **real**
soma, sub, mul, divs, opcao : **real**

// Parte Principal

inicio

escreval (“Operações sobre 2 números reais.”)
escreval (“Digite 1 para soma,”)
escreval (“ 2 para subtração, ”)
escreval (“ 3 para multiplicação ou”)
escreval (“ 4 para divisão.”)
leia (opcao)

se ((opção = 1) **ou** (opção = 2) **ou**
(opção = 3) **ou** (opção = 4)) **entao**

escreva (“Digite o primeiro número: ”)
leia (prim_num)

escreva (“Digite o segundo número: ”)
leia (seg_num)

se (opcao = 1) **entao**
soma <- prim_num + seg_num
escreval (“O resultado da soma é: ”, soma)
fimse

se (opcao = 2) **entao**
sub <- prim_num - seg_num
escreval (“ O resultado da subtração é: ”, sub)
fimse

se (opcao = 3) **entao**
mul <- prim_num * seg_num
escreval (“ O resultado da multiplic. é: ”, mul)
fimse

se (opcao = 4) **entao**
se (seg_num <> 0) **entao**
divs <- prim_num / seg_num;
escreval (“O resultado da divisão é: ”, divs)
fimse
se (seg_num = 0) **entao**
escreval (“Divisão por zero é impossível.”)
fimse
fimse

fimse

se ((opção <> 1) **e** (opção <> 2) **e**
(opção <> 3) **e** (opção <> 4)) **entao**

escreval (“Opção inválida”)

fimse

fimalgoritmo

Solução exercício V3.

Algoritmo “Operações elementares sobre dois números cf. opção do usuário V2”

Var

prim_num, seg_num, result, opcao : **Real**

// Parte Principal

Inicio

escreval (“ Operações sobre 2 números reais.”)

escreval (“ Digite 1 para soma,”)

escreval (“ 2 para subtração, ”)

escreval (“ 3 para multiplicação ou”)

escreval (“ 4 para divisão.”)

leia (opcao)

se ((opção = 1) **ou** (opção = 2) **ou**
 (opção = 3) **ou** (opção = 4)) **entao**

escreva (“Digite o primeiro número: ”)

leia (prim_num)

escreva (“Digite o segundo número: ”)

leia (seg_num)

se (opcao = 1) **entao**

result <- prim_num + seg_num;

escreval (“O resultado da soma é: ”, result)

fimse

se (opcao = 2) **entao**

result <- prim_num - seg_num

escreval (“ O resultado da subtração é: ”, result)

fimse

se (opcao = 3) **entao**

result <- prim_num * seg_num;

escreval (“O resultado da multiplic. é: ”, result)

fimse

se (opcao = 4) **entao**

se (seg_num <> 0) **entao**

result <- prim_num / seg_num;

escreval (“O resultado da divisão é: ”, result)

fim-se

se (seg_num = 0) **entao**

escreval (“Divisão por zero é impossível.”)

fim-se

fimse

fimse

se ((opção <> 1) **e** (opção <> 2) **e**
 (opção <> 3) **e** (opção <> 4)) **entao**

escreval (“Opção inválida”)

fimse

FimAlgoritmo

Exercício

- Elaborar um algoritmo onde o usuário possa escolher entre:
 - o cálculo do perímetro de uma circunferência cujo valor de raio é fornecido pelo usuário.
 - o cálculo da área de um retângulo cujos valores dos lados são fornecidos pelo usuário.
 - o cálculo da área de um triângulo retângulo cujos valores dos lados são fornecidos pelo usuário.
 - o cálculo do volume de um cubo cujo valor do lado...
 - o cálculo da área e do volume de uma esfera cujo valor do raio...

Obs.: Atenção com a prioridade de operadores... pág. 11 da apostila...

Início da Solução

```
algoritmo “Cálculo de Áreas e Volumes”  
var  
    opcao : caracter  
  
// Parte Principal  
inicio  
  
    escreval ( “Algoritmo de Cálculo de Áreas e Volumes. Escolha uma opção:” )  
    escreval ( “A – Perímetro de uma circunferência:” )  
    . . .  
  
    leia ( opcao )  
  
    se ( opcao = “A” ) então  
        // Algoritmo ‘Cálculo do Perímetro da Circunferência’  
    fimse  
  
    se ( opção = “B” ) então  
        // Algoritmo ‘Cálculo da Área do Retângulo’  
    fimse  
    . . .  
  
fimalgoritmo
```

Exercício

- Elabore um algoritmo que receba quatro notas de um aluno e calcule a média aritmética delas, dizendo se o aluno é aprovado, reprovado ou em exame:
 - Aprovação com média maior ou igual a sete.
 - Reprovação com média menor ou igual a cinco.
 - Exame nota entre cinco e sete (exclusive).

Uma solução

```
algoritmo "média"  
// Função : calcular média de alunos  
// Autor : Nilton Trento  
// Data : 15-Aug-08  
// Seção de Declarações  
var  
    n1, n2, n3, n4, media           : real  
    aluno, letra                   : caracter  
  
inicio  
    escreval ( " O nome do aluno: " )  
    leia ( aluno )  
    escreval ( " Digite suas notas: " )  
    leia ( n1, n2, n3, n4 )  
    media <- ( n1 + n2 + n3 + n4 ) / 4  
    escreval ( "Sua media foi", media )  
  
    se ( media >= 7 ) entao  
        escreval ( " O aluno ", aluno, " esta aprovado!" )  
    fimse  
  
    se ( ( media < 7 ) e ( media >= 5 ) ) entao  
        escreval ( " O aluno ", aluno, " está em exame." )  
    fimse  
  
    se ( media < 5 ) entao  
        escreval ( " O aluno ", aluno, " está reprovado." )  
    fimse  
fimalgoritmo
```

Exercício

- Pesquisar sobre Fluxogramas.
- Refazer todos os exercícios anteriores por meio de fluxogramas.

Obs.: Aplicar estes dois tópicos a todas as aulas de algoritmos.