

Fundamentos de Programação 1

Estrutura de decisão composta

Slides 7

Prof. SIMÃO

Jean Marcelo SIMÃO

Estrutura de Decisão ou Seleção Composta

se < condição for verdadeira > **entao**

um conjunto de comandos;

senao

outro conjunto de comando;

fimse

Exemplo de Algoritmo

Algoritmo “Analisar se um número inteiro é maior que 100 usando Se Senão”

// Parte Principal

Var

num : **Inteiro**

Inicio

escreval (“Digite um número: ”)

leia (num)

se (num > 100) **entao**

escreval (“Número maior que 100”)

senao

escreval (“Número menor ou igual a 100”)

fimse

FimAlgoritmo.

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.

Obs: Usar estrutura Se-Senão

Solução do Exercício

Algoritmo “Analisar se um número inteiro é maior que 100 e quantas vezes é maior – se *senão*”

Var

num : **Inteiro**

nvezes : **Real**

// Parte Principal

Inicio

escreval (“*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*”)

senao

escreval (“*Número menor ou igual a 100*”)

fimse

FimAlgoritmo

Segunda Versão

Algoritmo “Analise se um número inteiro é maior que 100 e quantas vezes é maior – Se Senão V2”

Var

num : **Inteiro**

// Parte Principal

Inicio

escreval (“*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”)

senao

escreval (“Número menor ou igual a 100”)

fimse

FimAlgoritmo

Terceira versão

Algoritmo “Analisar se um número inteiro é maior que 100 e quantas vezes é maior - Se Senão V.3”

Var

num : **Inteiro**

// Parte Principal

Inicio

escreval (“*Digite um número:* ”)

leia (num)

se (num > 100) **entao**

escreval (“*Número* ”, num , “ *é maior que 100*”)

escreval (“*O número* ”, num, “ *é maior que cem* ”, num / 100, “*vezes*”)

senao

escreval (“*Número* ”, num , “ *é menor ou igual a 100*”)

fimse

FimAlgoritmo

Exercício

Elaborar um algoritmo que analisa se um número inteiro é 1 , 2 ou diferente de 1 e 2

Obs: Usar estrutura Se-Senão

Solution

Algoritmo “Analisar se um número inteiro é o 1, 2 ou outro”

Var

num : **Inteiro**

// Parte Principal

Inicio

escreval (“ *Digite um número:* ”)

leia (num)

se (num = 1) **entao**

escreval (“ *É o número 1* ”)

senao

se (num = 2) **entao**

escreval (“ *É o número 2* ”)

senão

escreval (“ *Não é nem o número 1 nem o número 2* ”)

fimse

fimse

FimAlgoritmo

Solution errada

Esta solução está errada. Além de não 'rodar' em Visualg, o **senão** diz respeito apenas ao **se** imediatamente acima dele.

Algoritmo "Analisar se um número inteiro é o 1, 2 ou outro"

Var

num : **Inteiro**

// Parte Principal

Início

escreval (" *Digite um número:* ")

leia (num)

se (num = 1) **entao**

escreval (" *É o número 1* ")

se (num = 2) **entao**

escreval (" *É o número 2* ")

senão

escreval (" *Não é nem o número 1 nem o número 2* ")

fimse

FimAlgoritmo

Solution errada

Esta solução está errada. O **senão** diz respeito apenas ao **se** imediatamente acima dele.

Algoritmo “Analisar se um número inteiro é o 1, 2 ou outro”

Var

num : **Inteiro**

// Parte Principal

Início

escreval (“ *Digite um número:* ”)

leia (num)

se (num = 1) **entao**

escreval (“ *É o número 1* ”)

fimse

se (num = 2) **entao**

escreval (“ *É o número 2* ”)

senão

escreval (“ *Não é nem o número 1 nem o número 2* ”)

fimse

FimAlgoritmo

Exercício

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

Obs.: Usar a estrutura *Se Senão*

Uma primeira solução

Algoritmo 'Soma de dois ou três números'

Var

prim_num, seg_num : **Inteiro**
terc_num, soma, opcao : **Inteiro**

//Parte Principal

Início

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)

senao

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)

senao

escreval ("Opção inválida")

fimse

fimse

Fim.

Uma segunda versão

Algoritmo 'Soma de dois ou três números V2'

Var

prim_num, seg_num : **Inteiro**
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)

senao

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)

senao

escreval ("Opção inválida")

fimse

fimse

FimAlgoritmo

Melhorando a solução

**Seria possível uma solução melhor,
usando a estrutura *Se Senão* ?**

Segunda solução

Algoritmo Soma de dois ou três números

Var

prim_num, seg_num : **Inteiro**
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”) **ou** (opcao = “B”)) **entao**

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

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

soma ← prim_num + seg_num;

se (opcao = “A”) **entao**

escreval (“A soma é .:”, soma)

senao

escreval (“Digite o terceiro número: ”)
leia (terc_num)

soma ← soma + terc_num

escreval (“A soma é: ”, soma)

fimse

senao

escreval (“Opção inválida”)

fimse

FimAlgoritmo

Outra solução

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.

Obs. Usar *Se Senão*

Solução exercício.

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

Var

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

// Parte Principal

Início

```
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
```

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

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

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

senao

```
    se ( opcao = 2 ) entao
        result <- prim_num - seg_num
        escreval ( “O resultado da subtração é: ”, result )
    senao
        se ( opcao = 3 ) entao
            resul ← prim_num * seg_num
            escreval ( “O resultado da multiplic. é: ”, result )
        senao
            resul ← prim_num / seg_num
            escreval ( “O resultado da divisão é: ”, result )
        fimse
    fimse
```

fim-se

senão

```
    escreval ( “Opção inválida” )
```

fim-se

Fim.

Solução exercício – V2.

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

Var

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

// Parte Principal

Início

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) **entao**

result <- prim_num + seg_num

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

senao

se (opcao = 2) **entao**

result <- prim_num - seg_num

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

senão

se (opcao = 3) **entao**

result <- prim_num * seg_num;

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

senao

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

result <- prim_num / seg_num

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

senao

escreval (“Divisão por zero é impossível.”)

fim-se

fimse

fimse

fimse

senao

escreval (“Opção inválida”)

fimse

FimAlgoritmo

Solução exercício – V3.

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

Var

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

// Parte Principal

Início

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) **entao**

result <- prim_num + seg_num

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

senao

se (opcao = 2) **entao**

result <- prim_num - seg_num

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

senão

se (opcao = 3) **entao**

result <- prim_num * seg_num;

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

senao

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

result <- prim_num / seg_num

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

fimse

se (seg_num = 0) **entao**

escreval (“Divisão por zero é impossível.”)

fimse

fimse

fimse

fimse

senao

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.: Observação usar estrutura *Se Senão*

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).

Obs.: Observação usar estrutura *Se Senão*