



Programador de Sistemas

Lógica de Programação

Estrutura Condicional – 2ª parte

Prof. Mauricio Wieler Orellana

mauricioow@gmail.com

1. O custo ao consumidor de um carro novo é a soma do custo de fábrica com a porcentagem do distribuidor e com os impostos, ambos aplicados ao custo de fábrica. Sabe-se que as porcentagens são as mesmas que estão na tabela a seguir. Faça um programa que receba o custo de fábrica de um carro e mostre o custo ao consumidor.

CUSTO DE FÁBRICA	% DO DISTRIBUIDOR	% DOS IMPOSTOS
Até R\$ 12.000,00	5	isento
Entre R\$ 12.000,00 e R\$ 25.000,00	10	15
Acima de R\$ 25.000,00	15	20

Senac

```
algoritmo "Preço de Carros"
var
    custo_fab,dist,imp,custo_cons:real
inicio
    escreval("    P R E Ç O S   D E   C A R R O S    ")
    escreval("-----")
    escreval("Digite o custo de fábrica, do carro(em R$): ")
    leia(custo_fab)
    se (custo_fab <= 12000) entao
        dist <- custo_fab*0.05
        imp <- 0
    fimse
    se (custo_fab > 12000) e (custo_fab <= 25000) entao
        dist <- custo_fab*0.10
        imp <- custo_fab*0.15
    fimse
    se (custo_fab > 25000) entao
        dist <- custo_fab*0.15
        imp <- custo_fab*0.20
    fimse
    custo_cons <- custo_fab + dist + imp
    escreval("Custo do carro, ao consumidor: R$ ",custo_cons:5:2)
finalgoritmo
```

Faça um programa que calcule e imprima o valor da conta de água, a partir da leitura do consumo do mês anterior e do mês atual marcado no hidrômetro.

Sabe-se que a conta de água é formada pela tarifa de água somada à tarifa de esgoto (2,5% da conta de água) e à tarifa de conservação do hidrômetro (R\$ 5,00). O consumo de água é de acordo com a tabela mostra na sequência:

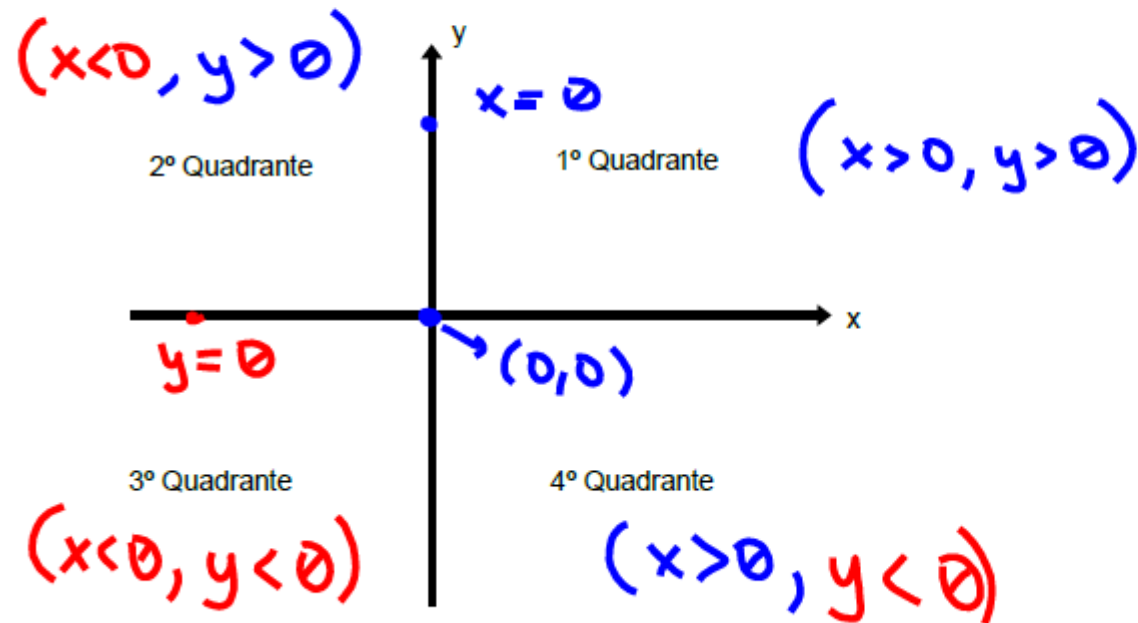
Consumo (m³)	Tarifa (R\$/m³)
0 – 10	0,69
11 – 15	1,17
16 – 25	1,48
Acima de 25	1,60

```

algoritmo "Conta de Água"
var
    conta,cons,cons_agua1,cons_agua2,tarifa_esgoto:real
inicio
    escreval("      C O N T A   D E   Á G U A      ")
    escreval("-----")
    escreval("Digite o consumo de água(em m³) do mês anterior: ")
    leia(cons_agua1)
    escreval("Digite o consumo de água(em m³) do mês atual: ")
    leia(cons_agua2)
    cons <- cons_agua1 + cons_agua2
    se (cons >= 0) e (cons <= 10) entao
        conta <- cons*0.69 + 5
    fimse
    se (cons >= 11) e (cons <= 15) entao
        conta <- cons*1.17 + 5
    fimse
    se (cons >= 16) e (cons <= 25) entao
        conta <- cons*1.48 + 5
    fimse
    se (cons > 25) entao
        conta <- cons*1.60 + 5
    fimse
    tarifa_esgoto <- conta*0.025
    conta <- conta + tarifa_esgoto
    escreval
    escreval("A conta de água será de R$ ",conta:4:2)
finalgoritmo

```

3. Escreva um algoritmo para ler as coordenadas (X,Y) de um ponto no sistema cartesiano e escrever o quadrante ao qual o ponto pertence. Se o ponto estiver sobre os eixos, ou na origem, escrever NÃO ESTÁ EM NENHUM QUADRANTE. Considere que o usuário poderá informar qualquer valor para as coordenadas.



Ler

Digite a abscissa
(x):

Digite a ordenada
(y):


```

algoritmo "Quadrantes"
var
  x,y:real
inicio
  escreval("      Q U A D R A N T E S      ")
  escreval("----- ")
  escreval("Coordenadas de um ponto P(x,y):")
  escreval("Digite o valor da abcissa (x): ")
  leia(x)
  escreval("Digite o valor da ordenada (y): ")
  leia(y)
  se (x > 0) e (y > 0) entao
    escreval("P(",x,",",y,") pertence ao 1º quadrante!")
  senao
    se (x < 0) e (y > 0) entao
      escreval("P(",x,",",y,") pertence ao 2º quadrante!")
    senao
      se (x < 0) e (y < 0) entao
        escreval("P(",x,",",y,") pertence ao 3º quadrante!")
      senao
        se (x > 0) e (y < 0) entao
          escreval("P(",x,",",y,") pertence ao 4º quadrante!")
        senao
          se (x = 0) ou (y = 0) entao
            escreval("P(",x,",",y,") não pertence a nenhum quadrante!")
          fimse
        fimse
      fimse
    fimse
  fimse
fimse
fimse
fimse
finalgoritmo

```

4. Crie um algoritmo que ache as raízes reais de uma equação do 2º grau:
 $AX^2+BX+C=0$

Dica: $D = B^2 - 4 \cdot A \cdot C$

Se $D < 0$, não existe raiz real.

Se $D = 0$, $X_1 = X_2 = -B / (2 \cdot A)$

Se $D > 0$:

$$X_1 = \frac{-B + \sqrt{\Delta}}{2A} \quad X_2 = \frac{-B - \sqrt{\Delta}}{2A}$$


```

algoritmo "Equação de Segundo Grau "
var
    a,b,c, delta,x1,x2: real
inicio
    escreval("Equação de Segundo Grau - Báscara")
    escreval("na forma  $ax^2 + bx + c = 0$ ")
    escreval("Digite o valor de a: ")
    leia(a)
    escreval("Digite o valor de b: ")
    leia(b)
    escreval("Digite o valor de c: ")
    leia(c)
    escreval
    escreval("A equação é: ",a,"x2","+",b,"x+",c," = 0")
    escreval
    delta <- b^2 - 4*a*c
    se (delta > 0) entao
        x1 <- (-b + RaizQ(delta))/(2*a)
        x2 <- (-b - RaizQ(delta))/(2*a)
        escreval(" e suas raízes são: ",x1, " e ",x2)
    fimse
    se (delta = 0) entao
        x1 <- -b/(2*a)
        escreval("possui apenas uma raiz: ",x1)
    fimse
    se (delta < 0) entao
        escreval("Não existem raízes reais para esta equação !")
    fimse
finalgoritmo

```

Estrutura de Seleção de Múltipla Escolha

- O algoritmo anterior pode ser escrito da seguinte forma, com a *estrutura de seleção múltipla*:

ESCOLHA X

CASO V1:

C1

CASO V2:

C2

CASO V3:

C3

CASO V4:

C4

FIM-ESCOLHA

select op
case 'j':

↓
VB / Delphi

switch(op)
case 'j':

↓
C / C++ / C# / Java.....

Estrutura de Seleção de Múltipla Escolha

ESCOLHA <variável>

CASO valor1:

instrução 1A

·
instrução nA

CASO valor2:

instrução 1B

·
instrução nB

...
CASO valorn:

instrução 1N

·
instrução nN

CASO-CONTRÁRIO:

instrução 1P

·
instrução nP

FIM-ESCOLHA

O tipo de dado para
<variável> pode ser
inteiro e caracter
somente.

O tipo de dado de
<variável> deve ser igual
ao tipo de valor1, valor2,
...

O bloco do identificador
CASO-CONTRÁRIO é
executado caso nenhuma
comparação com os
valores anteriores tenha
resultado em verdadeiro

Estrutura de Seleção de Múltipla Escolha

ESCOLHA <variável>

CASO valor1:

instrução 1A

instrução nA

CASO valor2:

instrução 1B

instrução nB

...

CASO valorn:

instrução 1N

instrução nN

CASO-CONTRÁRIO:

instrução 1P

instrução nP

FIM-ESCOLHA

Quando o conteúdo de <variável> for igual ao valor apresentado em um CASO, então a bloco correspondente é executado. Caso contrário, serão inspecionados os outros casos até ser encontrada uma igualdade ou terminarem os casos.

O bloco do identificador CASO-CONTRÁRIO (opcional) é executado caso nenhuma comparação com os valores anteriores tenha resultado em verdadeiro

Seleção de Múltipla Escolha

- Observações:
 - O tipo de dado de **<variável>** só pode ser inteiro ou caracter (em java: só **int** e **char**, além dos inteiros **byte** e **short**);
 - O tipo de dado de **<variável>** deve ser do mesmo tipo que os valores testados (valor1.. valor_n);
 - O bloco **default** é opcional;

CASO-CONTRÁRIO

5. Criar um algoritmo que leia a quantidade total de calorias de uma refeição a partir do usuário que deverá informar o prato, a sobremesa e a bebida (veja a tabela a seguir).

Prato	Calorias	Sobremesa	Calorias	Bebida	Calorias
Vegetariano	180 cal	Abacaxi	75 cal	Chá	20 cal
Peixe	230 cal	Sorvete <i>diet</i>	110 cal	Suco de laranja	70 cal
Frango	250 cal	Mouse <i>diet</i>	170 cal	Suco de melão	100 cal
Carne	350 cal	Mouse chocolate	200 cal	Refrigerante <i>diet</i>	65 cal

Sugestão: enumere cada opção de prato, sobremesa e bebida. Ou seja: Prato:

1 – Vegetariano, 2 – Peixe, 3 – Frango, 4 – Carne;

Sobremesa: 1 – Abacaxi, 2 – Sorvete diet, 3 – Mousse diet, 4 – Mousse de chocolate;

Bebida: 1 – Chá, 2 Suco de laranja, 3 – Suco de melão, 4 – Refrigerante diet.

```
qtde_cal <- qtde_cal + 75
caso 2
qtde_cal <- qtde_cal + 110
caso 3
qtde_cal <- qtde_cal + 170
caso 4
qtde_cal <- qtde_cal + 200
outrocaso
escreval("Opção Inválida !!!")
qtde_cal <- 0
fimescolha
limpatela
escreval("BEBIDAS:")
escreval("-----")
escreval("1- Chá")
escreval("2- Suco de Laranja")
escreval("3- Suco de Melão")
escreval("4- Refri Diet")
escreval("-----")
escreval("Digite a sua opção: ")
leia(op)
escolha op
caso 1
qtde_cal <- qtde_cal + 20
caso 2
qtde_cal <- qtde_cal + 70
caso 3
qtde_cal <- qtde_cal + 100
caso 4
qtde_cal <- qtde_cal + 65
outrocaso
escreval("Opção Inválida !!!")
qtde_cal <- 0
fimescolha
limpatela
escreval("A quantidade total de calorias consumida foi de: ",qtde_cal," cal")
finalgoritmo
```


Faça um algoritmo que mostre o menu de opções a seguir, receba a opção do usuário e os dados necessários

para executar cada operação.

Menu de opções:

1. Somar dois números.
2. Subtrair dois números
3. Multiplicar dois números
4. Dividir dois números
5. Raiz quadrada de um número

Digite a opção desejad

```
Algoritmo "Opções"
var
op :inteiro
res,res1, num1,num2: real
inicio
escreval("Calculadora A
Paragua")
escreval("Digite um número inteiro:")
leia(num1)
escreval("Digite outro número inteiro:")
leia(num2)
limpatela
escreval(" Menu de Opções ")
escreval("")
escreval("1Soma
dois números ")
escreval("2Subtrai
dois números ")
escreval("3Multiplica
dois números ")
escreval("4Divide
dois números ")
escreval("5Raiz
Quadrada de um número ")
escreval("")
escreval("Digite a sua opção: ")
```

```

algoritmo "Opç8es"
var
    op : inteiro
    res, real, num1, num2: real
inicio
    escreval("Calculadora - A. Faragua")
    escreval("Digite um número inteiro:")
    leia(num1)
    escreval("Digite outro número inteiro:")
    leia(num2)
    limpatela
    escreval("          Menu de Opç8es          ")
    escreval("-----")
    escreval("1- Soma dois números          ")
    escreval("2- Subtrai dois números       ")
    escreval("3- Multiplica dois números    ")
    escreval("4- Divide dois números        ")
    escreval("5- Raiz Quadrada de um número ")
    escreval("-----")
    escreval("Digite a sua opção: ")
    leia(op)
    escolha op
        caso 1
            res <- num1 + num2
            escreval(num1, "+", num2, " = ", res)

        caso 2
            res <- num1 - num2
            escreval(num1, "-", num2, " = ", res)

        caso 3
            res <- num1 * num2
            escreval(num1, "*", num2, " = ", res)

        caso 4
            res <- num1 / num2
            escreval(num1, "/", num2, " = ", res)

        caso 5
            res <- RaizQ(num1)
            real <- RaizQ(num2)
            escreval("RaizQuadrada(", num1, ") = ", res)
            escreval("RaizQuadrada(", num2, ") = ", real)

        outrocaso
            escreval("Opção Inválida !!!")

    fimsecolha
finalgoritmo

```

Estruturas de Repetição:

Usadas em algoritmos e em linguagens de programação para realizarmos tarefas exaustivas, repetitivas, evitando muitas linhas de instruções ou de código.

enquanto (l. de Programação \Rightarrow while)

Usada para assignar tarefas de repetição e, nas linguagens de programação é conhecida como while

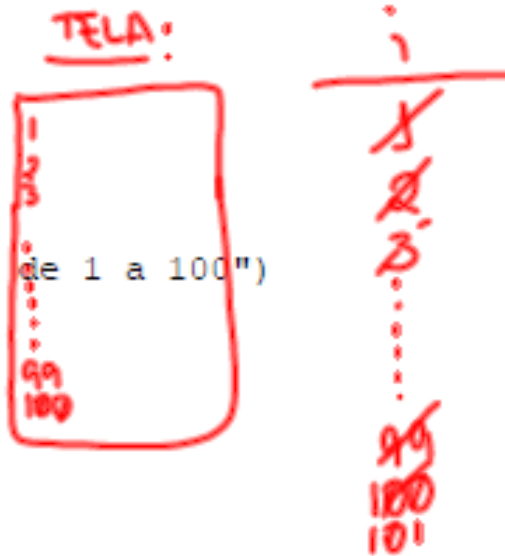
Sintaxe: enquanto <condição> faca inicia com
|
<instruções>
fim-enquanto

Obs: Para o enquanto
precisamos usar
um contador

Exemplo:

① Crie um algoritmo que escreva na tela os n^{os} inteiros entre 1 e 100;

```
algoritmo "Números de 1 a 100"
var
    i: inteiro
inicio
    escreval("Números inteiros de 1 a 100")
    i <- 1 //Inicialização
    enquanto (i <= 100) faça
        escreval(i)
        i <- i + 1 //Contador
    fimenquanto
fimalgoritmo
```



2. Crie um algoritmo que imprima os números inteiros de 100 a 1

```
algoritmo "Números de 100 a 1"
var
    i: inteiro
inicio
    escreval("Números inteiros de 100 a 1")
    i <- 100
    enquanto (i >= 1) faca
        escreval(i)
        i <- i - 1//i--;
    fimenquanto
fimalgoritmo
```

3. Crie um algoritmo que escreva os 100 primeiros números pares positivos na tela:

```
algoritmo "100 primeiros números pares"
var[L]SEP i: inteiro
inicio
    escreval("os 100 primeiros números pares são: ")
    i <- 2
//    i <- 1
    enquanto (i <= 200) faça
        //se (i % 2 = 0) então //
        //i%2 = 0 significa que i é PAR
        escreval(i)
        //fimse
        i <- i + 2
    fimenquanto
finalgoritmo
```


4. Crie um algoritmo que imprima na tela os números múltiplos de 5, entre 1 e 600:

```
algoritmo "Múltiplos de 5 entre 1 e 600"
var
    i: inteiro
inicio
    escreval("Os múltiplos de 5, entre 1 e 600, são: ")
    i <- 1
    enquanto (i <= 600) faca
        se (i % 5 = 0) entao //i%5 = 0 significa que i é MÚLTIPLO de 5
            escreval(i)
        fimse
        i <- i + 1
    fimenquanto
fimalgoritmo
```

5. Crie um algoritmo que imprima o quadrado dos números de 1 a 20:

$$1^2 = 1$$

$$2^2 = 4$$

$$3^2 = 9$$

.

.

.

$$19^2 = 361$$

$$20^2 = 400$$

```
algoritmo "Quadrados de 1 a 20"
var
    i,num:inteiro
inicio
    i <- 1
    escreval("Digite um número inteiro: ")
    leia(num)
    escreval("Quadrados dos números de 1 a ",num,":")
    enquanto (i <= num) faça
        escreval(i,"² = ",i^2)
        i <- i + 1
    fimenquanto
fimalgoritmo
```

for

Estrutura do para:

para valor de valorinicial ate valorfinal passo x faca
|
<instruções>
hipara

para i de (1) até 100 faça
 escreva (i)
fimpara

para i de 100 até 1 passo -1 faça
 escreva (i)
fimpara

(Quando não há "passo")
 passo = 1

1
2
3
4
5
...
97
98
99
100

100
99
...
3
1

Senac

6. Crie um algoritmo que leia dez números e imprima os ^LSE_P quadrados desses números:

```
algoritmo "Quadrados de 10 números"
var
    i,num:inteiro
inicio
    escreval("Cálculo do quadrado de 10 números")
    para i de 1 ate 10 faca
        escreval("Digite o ",i,"º número: ")
        leia(num)
        escreval(num,"² = ",num^2)
    fimpara
fimalgoritmo
```

7. Crie um algoritmo que imprima todos os números de 1 até 200 e imprima a soma deles.

$$S = 1 + 2 + 3 + 4 + \dots + 197 + 198 + 199 + 200$$

$S \leftarrow 0$
 $S \leftarrow S + i$

i	S
1	0
2	$0 + 1 = 1$
3	$1 + 2 = 3$
4	$3 + 3 = 6$
...	$6 + 4 = 10$
200	


```
algoritmo "Soma de números de 1 a 200"
var
    i,soma:inteiro
inicio
    soma <- 0
    escreval("Números de 1 a 200:")
    para i de 1 ate 200 faca
        escreval(i)
        soma <- soma + i
    fimpara
    escreval("A soma dos números de 1 a 200, vale: ",soma)
fimalgoritmo
```

8. Crie um algoritmo que faça uma tabela de conversão de [SEP]polegadas para centímetros. Deseja-se que na tabela constem [SEP]valores desde 1 polegada até 40 polegadas.

Sabe-se que $1 \text{ pol} = 2.54 \text{ cm}$

```
algoritmo "Tabela de Polegadas de 1 a 40"
var
    pol:inteiro
inicio
    escreval("Tabela de Conversão de Polegadas de 1 a 40:")
    para pol de 1 ate 40 faca
        escreval(pol," pol = ",pol*2.54," cm")
    fimpara
fimalgoritmo
```



```
algoritmo "Tabela de Conversão de Celsius a Farenheit"
var
    aux, ti, tf, F:inteiro
inicio
    escreval("Tabela de Conversão de F a C:")
    escreval("Digite a 1ª temperatura em °F: ")
    leia(ti)
    escreval("Digite a 2ª temperatura em °F: ")
    leia(tf)
    se (ti>tf) entao
        aux <- ti
        ti <- tf
        tf <- aux
    fimse
    para F de ti ate tf faca
        escreval(F , "°F = ", ((5/9) * (F-32)) : 5 : 1, "°C")
    fimpara
fimalgoritmo
```

10. Crie um algoritmo que mostre os números entre 1000 e 2000 (inclusive) que, quando divididos por 11 dão resto igual a 5

```
algoritmo "Resto 5 ao dividir por 11"
var
    i: inteiro
inicio
    escreval("Números entre 1000 e 2000 que divididos por 11 dão resto 5")
    para i de 1000 ate 2000 faca
        se (i%11 = 5) entao
            escreval(i)
        fimse
    fimpara
fimalgoritmo
```

11. Crie um algoritmo que leia um número n inteiro e $\boxed{\boxed{L}}_{\boxed{SEP}}$ calcule a soma S, dada por :

$$S = 1 + 1/2 + 1/3 + \dots + 1/n$$

$$= \frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{n}$$

$$S = \sum_{i=1}^n \frac{1}{i}$$

```
algoritmo "Soma de frações"
var
    S: real
    i,n:inteiro
inicio
    escreval("Digite um número inteiro
positivo:")
    leia(n)
    S <- 0
    para i de 1 ate n faca
        S <- S + 1/i
    fimpara
    escreval("A soma louca vale:
",S:3:2)
finalgoritmo
```