

Teste de Mesa

Ao nos recordarmos dos passos necessários para a construção de um algoritmo veremos que após a elaboração de um algoritmo devemos testá-lo realizando simulações com o propósito de verificarmos se este está ou não correto.

Existem alguns softwares disponíveis que efetuam a interpretação de algoritmos representados em pseudocódigos ou em fluxogramas.

Porém, existe uma técnica denominada “teste de mesa” que permite a simulação do processo de interpretação de um algoritmo utilizando apenas um papel e uma caneta.

Teste de Mesa

Para acompanhar o desenvolvimento de um algoritmo é importante verificar o estado dos dados a cada instrução, verificando o conteúdo de todas as variáveis contidas no algoritmo.

Sendo assim deve-se enumerar as linhas do algoritmo e em seguida criar uma tabela onde, a cada linha, são mostrados os conteúdos das variáveis do algoritmo e o número da linha executada.

Para uma melhor visualização do processo adotaremos a seguinte convenção: nas linhas em que uma variável é lida (entrada), o valor da variável ficará entre colchetes [] e quando o conteúdo de uma variável for escrito (saída), ficará entre chaves {}.

Aplicaremos a técnica do teste de mesa sobre o algoritmo abaixo, o qual visa determinar o número de vértices de uma figura geométrica.

algoritmo “vértices”

var vertices, faces, arestas: inteiro

inicio

escreva (“Entre com o número de faces da figura geométrica: ”)

leia (faces)

escreva (“Entre com um número da arestas da figura geométrica:”)

leia (arestas)

 vertices <- arestas + 2 – faces

escreva (“O número de vértices do objeto especificado é: ”, vertices)

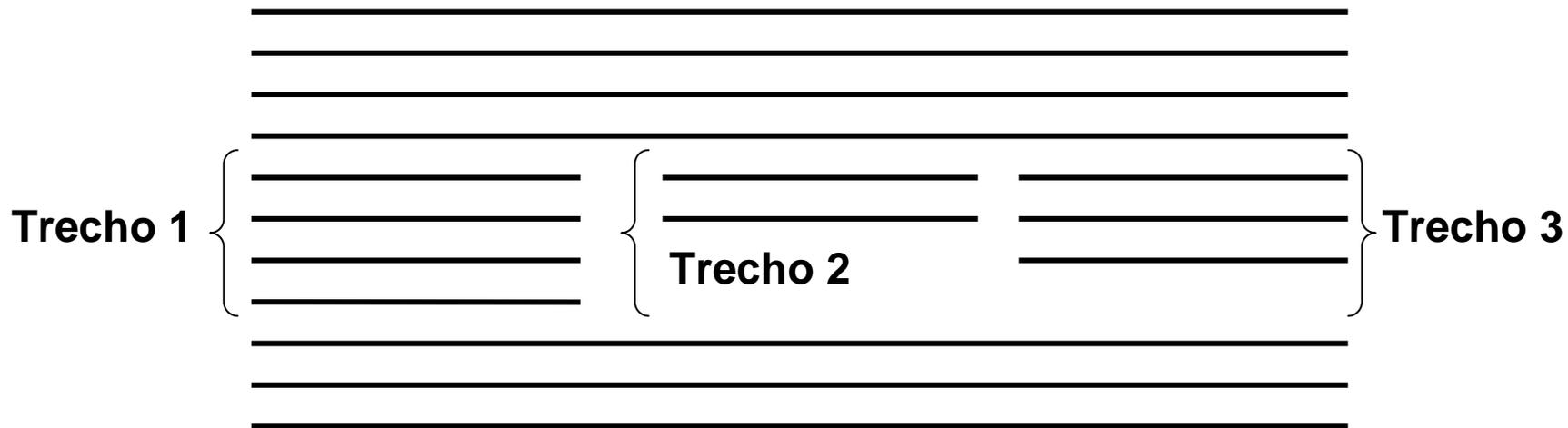
fimalgoritmo

Linha	Vertices	Faces	Arestas
1	?	?	?
2	?	[6]	?
3	?	6	?
4	?	6	[12]
5	8	6	12
6	{8}	6	12

Estruturas de Controle de Fluxo

2. Comando de Seleção Múltipla

Em algumas situações ao chegarmos a uma determinada instrução de um algoritmo devemos selecionar um dentre alguns trechos a seguir, tendo como base para esta escolha um conjunto de valores.



Para lidar com casos deste tipo foi criado o comando de seleção múltipla.

Estruturas de Controle de Fluxo

2. Comando de Seleção Múltipla (continuação)

Possuindo a sintaxe:

...

escolha (*<variável>*)

caso *<valor11>*, *<valor12>*, ..., *<valor1n>*

<seqüência-de-comandos-1>

caso *<valor21>*, *<valor22>*, ..., *<valor2m>*

<seqüência-de-comandos-2>

...

outrocaso

<seqüência-de-comandos-extra>

fimescolha

Estruturas de Controle de Fluxo

2. Comando de Seleção Múltipla (continuação)

Exemplo:

O pseudocódigo a seguir informa com base no nome de um time fornecido, se este trata-se de um time carioca, paulista ou de outro estado.

Estruturas de Controle de Fluxo

algoritmo " Exemplo Seleção Múltipla"

 var time: caractere

início

 escreva ("Entre com o nome de um time de futebol: ")

 leia (time)

 escolha (time)

 caso "Flamengo", "Fluminense", "Vasco", "Botafogo"

 escreva ("É um time carioca.")

 caso "São Paulo", "Palmeiras", "Santos", "Corinthians"

 escreva ("É um time paulista.")

 outrocaso

 escreva ("É um time de outro estado.")

 fimescolha

fimalgoritmo

Estruturas de Controle de Fluxo

2. Comando de Seleção Múltipla

Exercício:

Construa um algoritmo que receba dois números reais e um dos seguintes símbolos: +, -, * ou /, o qual designará qual operação será aplicada considerando os valores recebidos como seus operandos. O referido algoritmo deve retornar o resultado da operação selecionada com uma precisão de dois dígitos.

algoritmo "exercício"

var op1, op2: real

operador: caractere

inicio

escreva ("Entre com o primeiro operando: ")

leia (op1)

escreva ("Entre com o segundo operando: ")

leia (op2)

escreva ("Entre com um dos operadores (+, -, *, /): ")

leia (operador)

escolha (operador)

caso "+"

escreva (op1, " ", operador, op2, " =", op1+op2:10:2)

caso "-"

escreva (op1, " ", operador, op2, " =", op1-op2:10:2)

```
caso "*"
    escreva (op1," ",operador,op2," =", op1*op2:10:2)
caso "/"
    se (op2<>0) entao
        escreva (op1," ",operador,op2," =")
        escreval (op1/op2:10:2)
    senao
        escreva ("Não é possível efetuar a divisão!")
    fimse
outrocaso
    escreva ("Operação inválida! ")
fimescolha
fimalgoritmo
```

Estruturas de Controle de Fluxo

Em certos algoritmos percebemos que a execução de determinados trechos se faz necessária algumas vezes.

Um exemplo que nós já tratamos é o caso do cálculo da média aritmética entre alguns valores. Vimos que a operação de leitura é repetida de acordo com o número de valores que serviram de base para o cálculo da média. Outra forma de verificar a necessidade de repetição de um determinado trecho do algoritmo é imaginar que ao invés de se desejar apenas obter uma média, o objetivo fosse obter um conjunto de médias. Para o cálculo de cada uma das médias o mesmo conjunto de operações seria executado.

Este fato gerou a criação das estruturas de repetição, as quais veremos a seguir.

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição – enquanto

Sintaxe:

...

enquanto (*<expressão-lógica>*) faça
 <seqüência-de-comandos>

fimenquanto

...

Obs.: As instruções contidas no enquanto serão executadas durante o tempo em que o resultado da avaliação da expressão lógica resultar verdadeiro. O fato da avaliação da expressão lógica encontrar-se no início do laço faz com que a seqüência de comandos só venha a ser executada se ao menos uma vez a avaliação da expressão resultar em verdadeiro.

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição – enquanto (continuação)

Exemplo:

O pseudocódigo e os fluxogramas a seguir escrevem na saída padrão os números inteiros contidos no intervalo [1, 10].

Estruturas de Controle de Fluxo

algoritmo "exemplo 1 laço enquanto"

var

 valor: inteiro

inicio

 valor <- 1

 enquanto (valor <= 10) faça

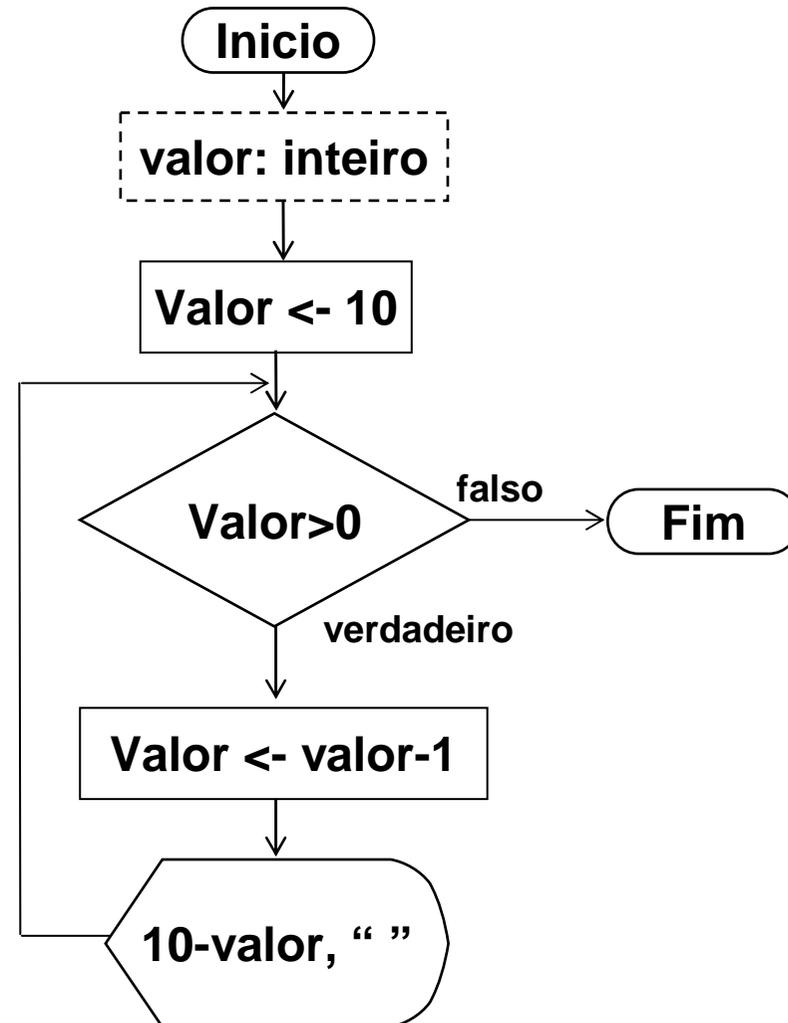
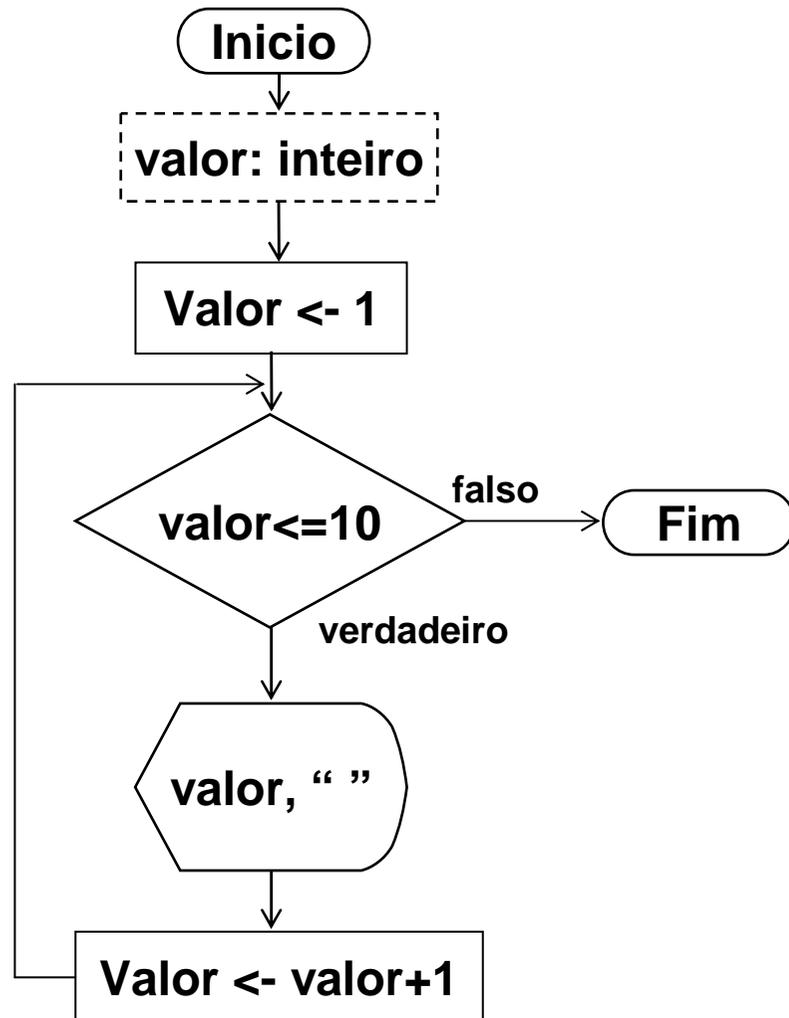
 escreval (valor)

 valor <- valor+1

 fimenquanto

fimalgoritmo

Estruturas de Controle de Fluxo



Estruturas de Controle de Fluxo

2. Estrutura ou laço de repetição – enquanto (continuação)

Exemplo:

Dada uma reta $ax+by+c=0$ e cinco pontos, faça um algoritmo para calcular, para cada ponto, o seguinte: se o ponto estiver no primeiro quadrante calcule e informe a distância do ponto a reta caso contrário escreva uma mensagem informando que o ponto não pertence ao primeiro quadrante.

Estruturas de Controle de Fluxo

```
algoritmo "exemplo laço enquanto "  
var a,b,c,x,y: real  
    contador: inteiro  
inicio  
    contador <- 1  
    escreval ("Equação da reta: ax+by+c=0")  
    escreva ("Coeficiente a da reta = ")  
    leia (a)  
    escreva ("Coeficiente b da reta = ")  
    leia (b)  
    escreva ("Coeficiente c da reta = ")  
    leia (c)
```

```

enquanto (contador<=5) faça
  escreval ("Coordenadas do ponto ",contador," :")
  escreva ("Coordenada x do ponto = ")
  leia (x)
  escreva ("Coordenada y do ponto = ")
  leia (y)
  se (x>=0) e (y>=0) então
    escreval ("A distância do ponto ",contador,
" a reta é: ",((a*x+b*y+c)^2)^0.5/(((a^2)+(b^2))^0.5))
  senão
    escreval ("O ponto não está no primeiro
quadrante! ")
  fimse
  contador <- contador + 1
fimenquanto
fimalgoritmo

```

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição – repita

Sintaxe:

...

repita

<seqüência-de-comandos>

ate (*<expressão-lógica>*)

...

Obs.: As instruções contidas no repita serão executadas enquanto o resultado da avaliação da expressão lógica resultar em falso. O fato da avaliação da expressão lógica encontrar-se no final do laço faz com que, mesmo no caso da expressão lógica nunca resultar em falso, a seqüência de comandos seja executada ao menos uma vez.

Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição – repita (continuação)

Exemplo:

O pseudocódigo e os fluxogramas a seguir escrevem na saída padrão os números inteiros contidos no intervalo [1, 10].

Estruturas de Controle de Fluxo

algoritmo "exemplo"

var

 valor: inteiro

inicio

 valor <- 0

 repita

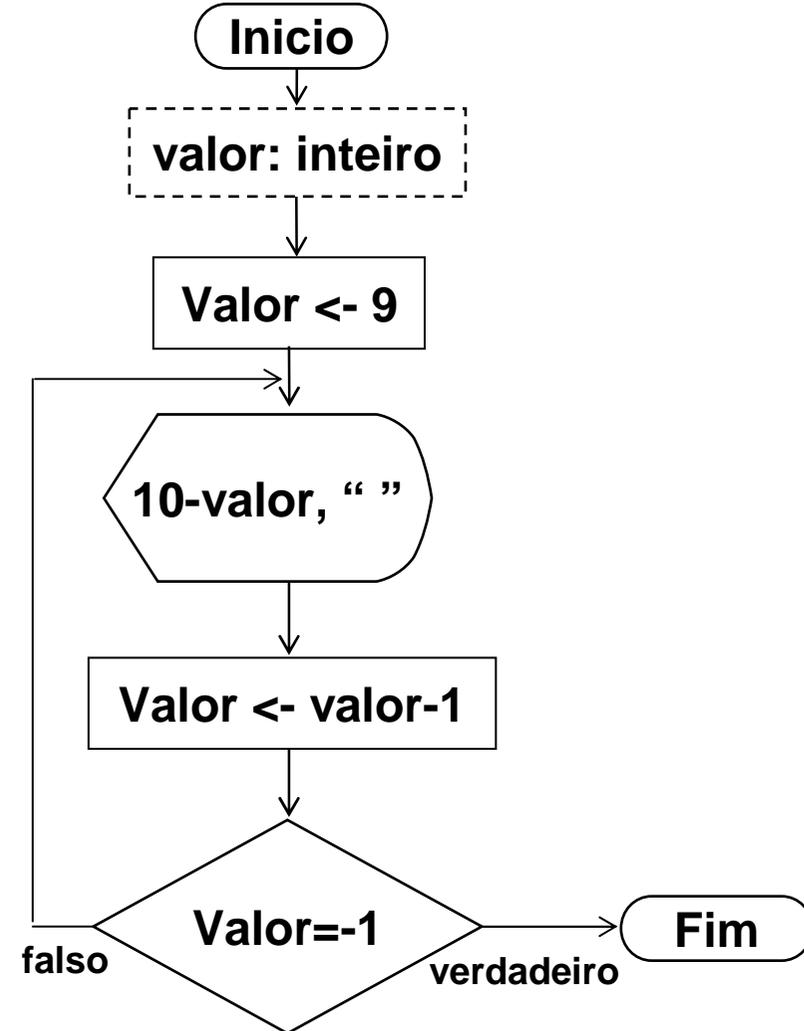
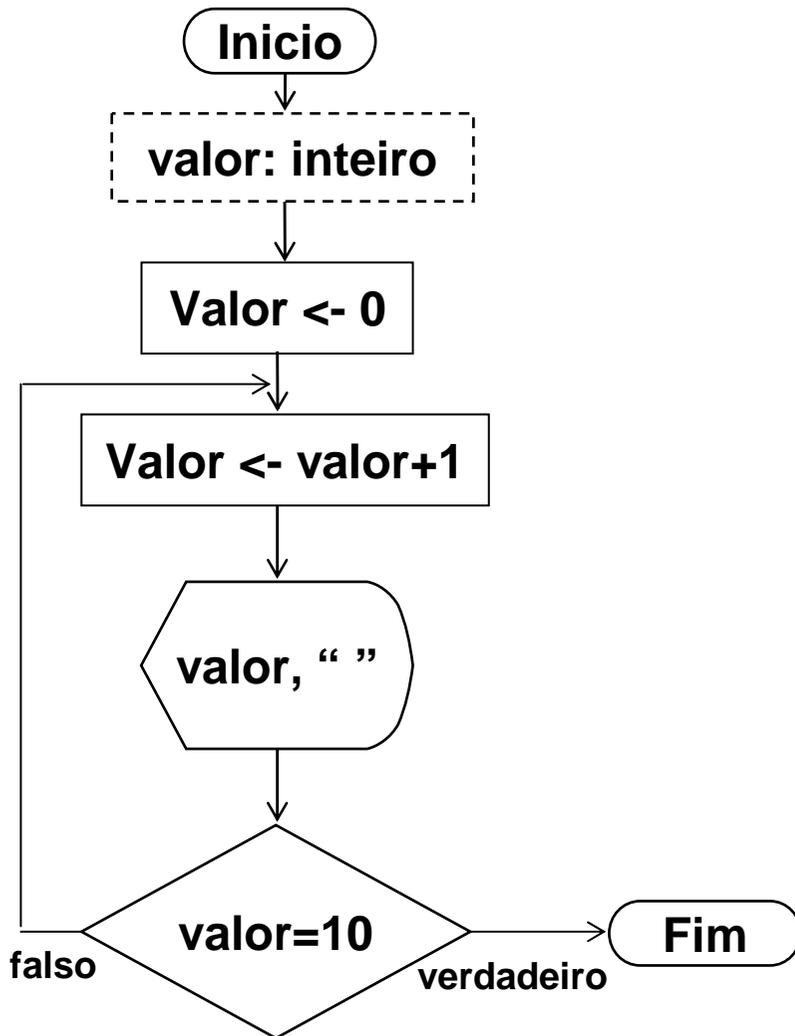
 valor <- valor+1

 escreval (valor)

 ate (valor = 10)

fimalgoritmo

Estruturas de Controle de Fluxo



Estruturas de Controle de Fluxo

3. Estrutura ou laço de repetição – **repita** (continuação)

Exemplo:

Dada uma reta $ax+by+c=0$ e cinco pontos, faça um algoritmo para calcular, para cada ponto, o seguinte: se o ponto estiver no primeiro quadrante calcule e informe a distância do ponto a reta caso contrário escreva uma mensagem informando que o ponto não pertence ao primeiro quadrante.

Estruturas de Controle de Fluxo

algoritmo "exemplo"

var a,b,c,x,y: real

contador: inteiro

inicio

contador <- 1

escreval ("Equação da reta: $ax+by+c=0$ ")

escreva ("Coeficiente a da reta = ")

leia (a)

escreva ("Coeficiente b da reta = ")

leia (b)

escreva ("Coeficiente c da reta = ")

leia (c)

repita

 escreval ("Coordenadas do ponto ",contador," :")

 escreva ("Coordenada x do ponto = ")

 leia (x)

 escreva ("Coordenada y do ponto = ")

 leia (y)

 se ((x>=0) e (y>=0)) entao

 escreval ("A distância do ponto ",contador,

 " a reta é: ",

$((a*x+b*y+c)^2)^{0.5}/(((a^2)+(b^2))^{0.5})$)

 senao

 escreval ("O ponto ",contador," não está no
primeiro quadrante! ")

 fimse

 contador <- contador + 1

ate (contador>5)

fimalgoritmo