



Estrutura Condicional

Estrutura Condicional

- A estrutura condicional permite determinar qual a ação a ser tomada com base no resultado de uma expressão condicional.
- Isso significa que podemos selecionar entre ações alternativas dependendo de critérios desenvolvidos no decorrer da execução do algoritmo.



Estrutura Condicional Simples

Se *condição*

então $\left\{ \begin{array}{l} \textit{instrução_1;} \\ \textit{instrução_2;} \\ \textit{instrução_n;} \end{array} \right.$

As instruções de 1 até n só serão executadas se a condição for verdadeira.

Estrutura Condicional Simples

Algoritmo exemplo1;

Var

num: real;

Início

Imprima "Digite um número: ";

Leia num;

Se (num >= 0)

então { Imprima "Este número é positivo";

Fim.

Estrutura Condicional Simples

Algoritmo exemplo2;

Var

num: real;

Início

Imprima "Digite um número: ";

Leia num;

Se (num \geq 0)

então

{ Imprima "Este número é positivo";

Se (num $<$ 0)

então

{ Imprima "Este número é negativo";

Fim.

Estrutura Condicional Simples

Escreva um algoritmo que leia um número inteiro e mostre se o nº lido é: positivo, negativo ou nulo.

Algoritmo exemplo3;

Var

num: inteiro;

Início

Imprima "Digite um número: ";

Leia num;

Se (num > 0)

então { Imprima "Este número é positivo";

Se (num < 0)

então { Imprima "Este número é negativo";

Se (num = 0)

então { Imprima "Este número é nulo";

Fim.

Exercícios

1. Construa um algoritmo que leia 2 números reais e mostre o maior entre eles.
2. Faça um algoritmo que leia: saldo, total de débitos e total de créditos. Em seguida, calcule e imprima o saldo atual do cliente. Utilize a seguinte fórmula: $\text{saldo atual} = \text{saldo} - \text{débito} + \text{crédito}$. Verifique se saldo atual é maior ou igual a zero e escreva a mensagem 'Saldo Positivo', senão escreva a mensagem 'Saldo Negativo'.

Estrutura Condicional Composta

Se *condição*

então {
instrução_1;
instrução_2;
instrução_n;



senão {
instrução_3;
instrução_4;
instrução_m;



Se a *condição* for **verdadeira**, serão executadas as instruções de 1 até n; caso contrário, se a *condição* for **falsa**, serão executadas as instruções de 3 até m.

Exercícios

1. Faça um algoritmo em que o usuário precise digitar o valor do raio. A seguir calcule e mostre:

- a) O comprimento de uma esfera, sabe-se que: $C = 2 * \pi * R$;
- b) A área de uma esfera, sabe-se que: $A = \pi * R^2$;
- c) O volume de uma esfera, sabe-se que: $V = (4 * \pi * R^3)/3$.

$$\pi = 3.14159$$

Não permita que o usuário digite $R = 0$. Se isso ocorrer, mostre a mensagem “Raio não pode ser 0” e finalize o algoritmo.

2. Elabore um algoritmo que leia o sexo de uma pessoa. Se o sexo digitado for “M” ou “F”, imprimir “Sexo válido”, caso contrário imprima “Sexo inválido!”.

```
algoritmo um;
var
  R, C, A, V: real;
início
  Imprima "Digite o valor do Raio: ";
  Leia R;
  Se (R = 0)
    então { Imprima "Raio não pode ser zero";

    senão {
      C  $\leftarrow$  2 * 3.14 * R;
      A  $\leftarrow$  3.14 * R**2;
      V  $\leftarrow$  (4 * 3.14 * R ** 3)/3;
      Imprima "O comprimento da esfera:", C;
      Imprima "A área da esfera:", A;
      Imprima "O volume da esfera:", V;
    }
fim.
```

Resolução em C++

```
#include<math.h> //escrever os outros include
using namespace std;
int main() {
    float r, c, a, v;
    cout << "\nDigite o valor do Raio: ";
    cin >> r;
    if (r == 0){
        cout << "\nRaio nao pode ser 0";
    }
    else{
        c = 2 * 3.1415 * r;
        a = 3.1415 * pow(r, 2);
        v = (4 * 3.1415 * pow(r, 3))/3;
        cout << "\nO comprimento da esfera: " << c;
        cout << "\nA area da esfera: " << a;
        cout << "\nO volume da esfera: " << v;
    }
}
} //fim
```

Algoritmo dois;

Var

 sx: caractere;

Início

 Imprima “Digite o sexo: “;

 Leia sx;

Se (sx = “M” **OU** sx = “F”)

então {Imprima “Sexo válido”;

 senão {Imprima “Sexo inválido”;

Fim.

```
#include<conio.h>
#include<iostream>
#include <locale.h>
using namespace std;

main(){
    setlocale(LC_ALL, "");
    char sx;
    cout << "\nDigite o Sexo: ";
    cin >> sx;
    if (toupper(sx) == 'M' || toupper(sx) == 'F'){
        cout << "Sexo Válido!";}
    else{
        cout << "Sexo Inválido";
    }
}
```

3. Escreva um algoritmo que leia um número real e, a seguir, calcule e mostre:
 - a) O número multiplicado por 10;
 - b) O número dividido por um outro valor a ser digitado pelo usuário. Se o valor digitado for igual a 0, mostre mensagem “Valor não pode ser nulo” e finalize o algoritmo.
 - c) A raiz quadrada do número digitado. Se o valor digitado for negativo, mostre mensagem “Valor não pode ser negativo” e finalize o algoritmo. Para o cálculo da raiz quadrada, utilize a função `sqrt(num)`.

Análise do enunciado:

- Valor de entrada:
 - 1º valor a ser digitado.
 - 2º valor a ser digitado.
- Processamento:
 - Multiplicar o 1º valor por 10.
 - Dividir o 1º valor pelo 2º valor.
 - Restrição: verificar se o 2º valor é $\neq 0$.
 - Restrição: verificar se o 1º valor é < 0 .
 - A raiz quadrada do 1º valor.
- Saída:
 - Imprimir o 1º valor multiplicado por 10.
 - Imprimir a divisão do 1º valor pelo 2º valor.
 - Imprimir a raiz quadrada do 1º valor.

algoritmo três;

var

N1, N2, Resposta: real;

início

Imprima "Digite um número: ";

Leia N1;

Imprima N1, " multiplicado por 10: ", $N1 * 10$;

Imprima "Digite outro número: ";

Leia N2;

Se ($N2 = 0$)

então { Imprima "Valor não pode ser nulo";

senão { Resposta \leftarrow $N1 / N2$;

Imprima N1, " dividido por ", N2, " = ", Resposta;

Se ($N1 < 0$)

então { Imprima "Valor não pode ser negativo";

senão { Imprima "A raiz quadrada de ", N1, " é: ", $\text{sqrt}(N1)$;

fim.


```
#include<math.h> //escrever os outros include
using namespace std;
int main() {
    float n1, n2, resposta;
    cout << "\nDigite um numero: ";
    cin >> n1;
    cout << "\n" << n1 << " multiplicado por 10: " << n1*10;
    cout << "\nDigite outro numero: ";
    cin >> n2;
    if (n2 == 0){
        cout << "\nEsse valor nao pode ser nulo";
    }
    else{
        resposta = n1 / n2;
        cout << "\n" << n1 << " dividido por " << n2 << " = " << resposta;
    }
    if (n1 < 0){
        cout << "\nO 1º valor nao pode ser negativo";
    }
    else{
        cout << "\nA raiz quadrada de " << n1 << " e: " << sqrt(n1);
    }
}
```



FIM