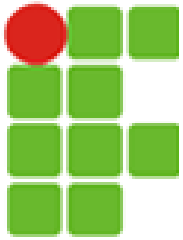

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

Linguagem de Programação Estruturada – LPES1

Tecnologia em Análise e Desenvolvimento de Sistemas

Estruturas Condicionais

Professor:

Fernando Vieira Duarte
fernandoduarte@ifsp.edu.br

Aula 3 - 09/03/2020

Estruturas de Decisão

Uma estrutura de decisão permite ao programador **alterar a sequência de execução do programa** em função do resultado da avaliação de uma ou mais condições.

Uma **condição** é uma expressão lógica, que retorna V ou F.

Classificação das Estruturas de Decisão em linguagem C:

Estrutura de Decisão **Simples** (***if***...)

Estrutura de Decisão **Composta** (***if*** ...***else***...)

Estrutura de Decisão **Múltipla** do Tipo Escolha (***switch***... ***case***...)

Estrutura de Decisão Simples (*if...*)

Nesta estrutura uma **única condição** (expressão lógica) é avaliada.

Dependendo do resultado desta avaliação, um comando ou conjunto de comandos serão executados (se a avaliação for verdadeira) ou não serão executados (se a avaliação for falsa).

Sintaxe da Estrutura de Decisão Simples

Em algoritmo:

se (condicao) *então*

<comandos>

fim_se

Sintaxe da Estrutura de Decisão Simples

```
if (condicao){
    // comandos;
}
```

Exemplo 1 - Estrutura de Decisão Simples

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
  float nota;
  printf("Digite a nota do aluno: ");
  scanf("%f", &nota);

  if (nota >= 6){
     printf("\nAprovado!\n\n");
  }
  if (nota <6){
     printf("\nReprovado!\n\n");
  }
  system("PAUSE");
  return 0;
}
```

Exemplo 2 - Estrutura de Decisão Simples

```
#include <stdio.h>
#include <stdlib.h>
int main( )
{
   int numero;

   printf (“Digite um numero: ” );
   scanf (“%d”, &numero);

   if  (  (numero % 2) == 0){
       printf (“O numero %d eh par”, numero);
   }

   system("PAUSE");
   return 0;
}
```

Estrutura Decisão Composta (*if..else*)

Nesta estrutura uma **única condição** (expressão lógica) é avaliada.

Se o resultado desta avaliação for verdadeiro, um comando ou conjunto de comandos serão executados. Caso contrário, ou seja, **quando o resultado da avaliação for falso, um outro comando ou um outro conjunto de comandos será executado**.

Sintaxe da Estrutura de Decisão Composta

Em algoritmo:

se (condicao) *então*

    <comandos>

fim_se

senão

    <comandos>

fim_senão

Sintaxe da Estrutura de Decisão Composta

```
if ( condicao ){
        // comandos;
}else{
        // comandos;
}
```

Exemplo - Estrutura de Decisão Composta

```
#include <stdio.h>
#include <stdlib.h>
int main( ){
      int numero;
      printf (“Digite um numero: ”);
      scanf (“%d”, &numero);

      if  ((numero % 2) == 0){
         printf(“O numero %d eh par”, numero);
      }else{
         printf(“O numero %d eh impar”, numero);
      }
      system("PAUSE");
      return 0;
}
```

Estrutura de Decisão Concatenadas

Os programas podem ser baseados em **estruturas concatenadas** de duas maneiras:

- uma em sequência a outra, ou
- em estruturas aninhadas (ou encadeadas) uma dentro da outra.

Exemplo de Estrutura de Decisão: em sequências

```
#include <stdio.h>
#include <stdlib.h>
main(){
        int x,y;

        scanf("%d%d", &x, &y);

        if((x == 0) && (y == 0)){
                printf("Ponto origem");
        }

        if((x == 0) && (y != 0)){
                printf("Ponto eixo y");
        }

        if((x != 0) && (y == 0)){
                printf("Ponto eixo x");
        }
        if((x > 0) && (y > 0)){
                printf("Quadrante 1");
        }

        if((x < 0) && (y > 0)){
                printf("Quadrante 2");
        }

        if((x < 0) && (y < 0)){
                printf("Quadrante 3");
        }

        if((x > 0) && (y < 0)){
                printf("Quadrante 4");
        }
        system("PAUSE");
        return 0;
}
```

Exemplo de Estrutura de Decisão: aninhadas

```
#include <stdio.h>
#include <stdlib.h>
main(){
   int x,y;
   scanf("%d%d", &x, &y);
   if((x == 0) && (y == 0)){
      printf("Ponto origem");
   }else {
       if((x == 0) && (y != 0)){
            printf("Ponto eixo y");
       } else {
            if((x != 0) && (y == 0)){
            printf("Ponto eixo x");
            } else {
            if((x > 0) && (y > 0)){
                printf("Quadrante 1");
            } else {
                if((x < 0) && (y > 0)){
                   printf("Quadrante 2");
                } else {
                   if((x < 0) && (y < 0)){
                      printf("Quadrante 3");
                   } else {
                      printf("Quadrante 4");
                   }
                }
                }
            }
       }
   }
   system("PAUSE");
   return 0;
}
```

Exemplo 2 de Estrutura de Decisão: aninhadas

```
#include <stdio.h>
#include <stdlib.h>
int main( )
{
  float nota = 0;

  printf ("Digite a nota: ");
  scanf ("%f", &nota);
if  ((nota >= 8) && (nota <= 10))
      printf ("\n Conceito A! \n\n");
   else
      if  ((nota >= 6) && (nota < 8))
         printf ("\n Conceito B! \n\n");
      else
         if  ((nota >= 4) && (nota <= 5))
            printf ("\n Conceito C! \n\n");
          else
             if  ((nota >= 2) && (nota <= 3))
                printf ("\n Conceito D! \n\n");
             else
                if  ((nota >= 0) && (nota <= 1))
                   printf ("\n Conceito E! \n\n");
                else
                   printf ("\n Nota invalida! \n\n”);
   system("PAUSE");
   return 0;
}
```

Comentários sobre as Estrutura de Decisão

As **estruturas concatenadas** tem a vantagem de tornar o algoritmo mais legível, facilitando a correção do mesmo em caso de erros.

As estruturas **aninhadas** tem a vantagem de tornar o algoritmo mais rápido pois são efetuados menos testes e menos comparações, o que resulta num menor número de passos para chegar ao final do mesmo.

Estrutura de Decisão de Múltipla Escolha

Na estrutura de decisão do tipo **Múltipla Escolha** pode haver uma ou mais condições a serem testadas e um comando diferente associado a cada uma destas condições.

Estrutura de Decisão de Múltipla Escolha

Em algoritmo:

```
escolha (nomeVariavel)
início
        caso valorVariavel:
        inicio
            <comandos>
        fim_caso
        caso valorVariavel:
        início
            <comandos>
         fim_caso
        ...
        senão:
        início
            <comandos>
        fim_senão
fim_escolha
```

Estrutura de Decisão de Múltipla Escolha

```
switch (nomeVariavel)
{
        case valorVariavel:
        {
                // comandos;
                break;
        }
        case valorVariavel:
        {
                // comandos;
                break;
        }
        ...
        default:
        {
           // comandos;
            break;
        }
}
```

Exemplo de Estrutura de Decisão de Múltipla Escolha

```
...
  float salario, novoSalario;
  int profissao;

  printf("Digite sua profissao >> [1] Medico;
[2] Enfermeiro; [3] Farmaceutico: ");
  scanf("%d", &profissao);

  printf("Digite seu salario: R$");
  scanf("%f", &salario);

  switch(profissao)
  {
    case 1:
    {
        novoSalario = salario * 1.3;
        printf("Salario com aumento de 30/100
        = R$%.2f\n\n", novoSalario);
        break;
    }
    case 2:
    {
        novoSalario = salario * 1.45   ;
        printf("Salario com aumento de 45/100
        = R$%.2f\n\n", novoSalario);
        break;
    }
    case 3:
    {
        novoSalario = salario * 1.2   ;
        printf("Salario com aumento de
               20/100 = R$%.2f\n\n",
               novoSalario);
        break;
    }
    default:
    {
        printf("Profissao nao
        cadastrada!\n\n");
        break;
    }
  }//fim do switch

...
```

Operadores

Operadores de incremento e decremento

O símbolo do operador de incremento na linguagem C é **++**. Podendo ser pós-fixado (variavel++) ou pré-fixado (++variavel).

O símbolo do operador de decremento na linguagem C é **--**. Podendo ser pós-fixado (variável--) ou pré-fixado (--variavel).

Exemplo:

```
...
   int a = 10; int b = 10; int c = 10; int d = 10;

   a++;
   printf("A = %i n", a);

   ++b;
   printf("B = %i n", b);

   a--;
   printf("A = %i n", a);

   --b;
   printf("B = %i n", b);
```

Operadores

Os **operadores de atribuição** são usados para cálculo de expressões em linguagem C. Os operadores de atribuição são listados no quadro a seguir:

OPERADORES DE ATRIBUIÇÃO DA LINGUAGEM C				
Operação		**Operador**	**Exemplo**	**Equivalente**
Adição		+=	num += 7;	num = num + 7;
Subtração		-=	num -= 17;	num = num - 17;
Multiplicação		*=	num *= 28;	num = num * 28;
Divisão		/=	num /= 37;	num = num / 37;
Módulo		%=	num %= 53;	num = num % 53;

Os **Operadores de igualdade** são utilizados para estabelecer equivalência e podem ser empregados para avaliar se uma determinada variável é igual, ou diferente, à outra variável. Os operadores de igualdade são listados no quadro a seguir:

OPERADORES DE IGUALDADE			
Operador	**Símbolo**	**Exemplo**	**Resultado**
Igualdade	==	7 == 17	Falso
Diferença	!=	7 != 17	Verdadeiro

Operadores (continuação)

Os **operadores relacionais** são empregados na comparação entre dados do mesmo tipo. Os operadores relacionais são listados no quadro a seguir:

OPERADORES RELACIONAIS			
Operador	**Símbolo**	**Exemplo**	**Resultado**
maior que	>	3 > 122	Falso
menor que	<	15 < 7	Falso
maior ou igual que	>=	13 >= 13	Verdadeiro
menor ou igual que	<=	12 <= 34	Verdadeiro

Os **operadores lógicos** são utilizados para concatenar diversas expressões lógicas. O quadro a seguir ilustra os operadores lógicos, e sua notação, na linguagem C.

PRINCIPAIS OPERADORES LÓGICOS			
Operador	**Representação na Lógica Matemática**	**Representação na Linguagem C**	**Exemplo**
Conjunção	E (∧)	&&	A && B
Disjunção	OU (∨)	||	A || B
Negação	Negação (~)	!	!A

Operadores (continuação)

As tabelas verdade possuem os mesmos resultados da lógica matemática, conforme ilustra o quadro a seguir:

A	***B***	***A && B***	***A || B***
1	0	0	1
1	1	1	1
0	1	0	1
0	0	0	0

ORDEM DE EXECUÇÃO DOS OPERADORES	
Prioridade	**Operador**
1º	parênteses mais internos
2º	potenciação, raiz
3º	*** /** (multiplicação, divisão)
4º	**+ -** (soma, subtração)

Dúvidas

Exercícios

1. Faça um programa em Linguagem C que leia os valores A, B, C e diga se a soma de A + B é menor que C.
2. Faça um programa em Linguagem C que leia dois valores inteiros A e B. Se os valores forem iguais deverá ser calculada a soma dos dois valores, caso contrário multiplicar A por B e armazenar o resultado em uma variável C.
3. Faça um programa em Linguagem C que leia o nome e a três notas (lista = 25%, prática = 25% e prova = 50%) de uma disciplina de uma aluno e ao final escreva o nome do aluno, sua média ponderado e se ele foi aprovado ou reprovado. A média mínima para aprovação é 6.
4. Faça um programa em Linguagem C que leia 3 números inteiros e imprima o menor e o maior deles.
5. Dados três valores inteiros e distintos, A, B e C, fazer um programa em Linguagem C que, após a leitura destes dados coloque-os em ordem crescente e depois em ordem decrescente.
6. Dado três valores X, Y, Z, verificar se eles podem ser os comprimentos dos lados de um triângulo, e se forem, verificar se é um triângulo equilátero, isósceles ou escaleno. Se eles não formarem um triângulo, escrever uma mensagem. O comprimento de cada lado de um triângulo é menor do que a soma dos comprimentos dos outros dois lados.
    - Definição 1 - Chama-se triângulo equilátero os que tem os comprimentos dos três lados iguais,
    - Definição 2 - Chama-se triângulo isósceles ao triângulo que tem os comprimentos de dois lados iguais.
    - Definição 3 - Chama-se triângulo escaleno ao triângulo que tem os comprimentos dos três lados diferentes.

Exercícios

7. Faça um programa em C para verificar se o ano digitado pelo usuário é bissexto.
8. Faça um programa em C para calcular a conta de energia elétrica de uma casa. O usuário deve digitar a quantidade de kWh consumido no mês. O valor de um kWh é R$ 0,32. Entretanto, quando a casa é de uma pessoa aposentada, a conta tem um desconto de 15%.
9. Faça um programa em C para ler o consumo de água de uma residência e calcular o valor da conta de água, considerando a seguinte tabela de gastos:

M^3	Cada M^3
0 – 10	R$ 1,20
11 – 20	R$ 1,50
Acima de 20	R$ 2,00

10. Faça um programa em C que receba como entrada a altura e o sexo de uma pessoa e que calcule e mostre o seu peso ideal, utilizando as seguintes fórmulas:
    - Para homens: (72.7 * altura) – 58;
    - Para mulheres: (62.1 * altura) – 44.7.

Exercícios

11. Implemente um programa em C para calcular quantos reais serão necessários para encher o tanque de um veiculo para se realizar uma viagem. O usuário deverá informar o tipo de combustível do veículo, o número total de km a ser percorrido e o consumo médio do veículo. A tabela de preços dos combustíveis utilizada no cálculo é apresentada abaixo:

Combustível	Preço
Gasolina	R$ 4,49
Álcool	R$ 2,89
Diesel	R$ 3,08

12. Suponha que um caixa disponha apenas notas de R$100, R$10 e R$1. Considerando que alguém está pagando uma compra, faça uma aplicação para determinar o número mínimo de notas que o caixa deve fornecer como troco. Imprima também o valor da compra, o valor do troco e a quantidade de cada tipo de nota a ser fornecido como troco. Suponha que o sistema monetário não utilize centavos.
13. Uma agência bancária possui dois tipos de investimentos, conforme o quadro abaixo. Faça um programa em C que receba o tipo do investimento e o valor do investimento e que calcule e mostre o valor corrigido de acordo com o tipo de investimento.

Tipo	Descrição	Rendimento Mensal
1	Poupança	3%
2	Fundos de renda Fixa	4%