Algoritmos e Lógica de Programação

80 horas // 4 h/semana

Estrutura Condicional: simples e composta

Aula 05

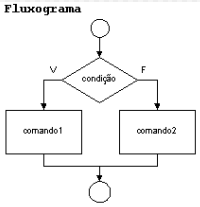
Prof. Piva

Estruturas condicionais...

- Existem comandos que, a partir de uma **condição**, permitem que o programa siga por um caminho ou por outro.
- Em programação o uso de **condições para permitir a escolha** de executar ou não um trecho de programa é muito utilizado, principalmente quando precisamos incluir no programa **condições de controle**, para evitar situações não permitidas, que podem resultar em erros. Por exemplo, para evitar divisões por zero.

Para começar...

Fluxograma



Forma Textual

```
se condição então
        comando-1
senão
        comandos-2
fimse
```

análise do comando:

- as palavras **se**, **então** e **senão** representam o comando condicional;
- a *condição* deve ser uma expressão lógica;

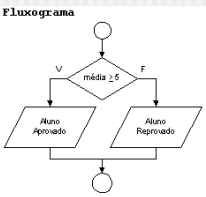
Para começar...

Imagine que se quer escrever um algoritmo/programa para ler duas notas de um aluno, calcular a média aritmética e além de imprimir os dados lidos e a média calculada, deseja-se que se imprima se, com a média obtida, o aluno está aprovado (maior ou igual a 5) ou reprovado (menor do que 5).

- Vejam que agora existe uma escolha a ser feita: o aluno está aprovado ou reprovado?
- **A resposta é**: depende da média obtida. **SE** a média for maior ou igual a 5 **ENTÃO** o aluno está aprovado **SENÃO**, se a média for menor do que 5 então está reprovado.
- Reparem nas palavras escritas em letras maiúsculas, elas compõem a sintaxe dos comandos condicionais.

Para começar...

Fluxograma



Forma Textual

```
se media ≥ 5 então
      Escreva "Aluno APROVADO"
senão
      Escreva "Aluno REPROVADO"
fimse
```

Algoritmo do cálculo da média do aluno

Para começar...

- **Expressões lógicas:** são as expressões cujos operadores são lógicos e cujos operandos são: relações, constantes e/ou variáveis lógicas.
- Pode-se construir um comando condicional contendo uma expressão lógica muito simples contendo apenas uma variável lógica até uma contendo vários operadores lógicos e relacionais.

Para começar...

Exemplo 01: supor a variável *alfa* do tipo lógico – significa que poderá receber os valores V (verdade) ou F (falso).

se *alfa* = V **então**
  comando$_1$
senão
  comando$_2$
fimse

Assim, se *alfa* for verdade então é executado o comando$_1$. Se *alfa* for falso **não** executará o comando$_1$ e executará o comando$_2$.

Observação: não é necessário comparar alfa com V, bastaria construir a condição com: **se** alfa **então.**

Para começar...

Exemplo 02: supor as variáveis *alfa* e *beta* do tipo lógico e as variáveis *x* e *y* do tipo inteiro.

se *alfa* = V e *beta* = F e (*x* = 0 ou *y* = 0) **então**
  comando$_1$
senão
  comando$_2$
fimse

Assim, **se** *alfa* for verdade, *beta* falso e, *x* ou *y* igual a zero **então** é executado o ***comando*$_1$ senão**, se qualquer coisa diferente ocorrer, pulará o ***comando*$_1$** e executará o ***comando*$_2$.**

Para começar...

Do exemplo anterior, na expressão lógica:

alfa = V **e** *beta* = F **e** (*x* = 0 **ou** *y* = 0)

- cada uma das expressões: *alfa* = V, *beta* = F, *x* = 0, *y* = 0; são chamadas de **expressões relacionais;**
- o sinal de igualdade (=) que aparece nas expressões relacionais, é chamado de **operador relacional** e indica uma comparação. Em, *x* = 0, está comparando x com zero;
- ***e*** e ***ou***, são os **conectores**, que ligam as expressões relacionais entre si, formando as expressões lógicas;
- V e F são constantes lógicas;
- *alfa* e *beta*, que são comparadas com as constantes lógicas V e F, são variáveis lógicas.

Para começar...

Operadores Relacionais

=	igual a
≠	diferente de (não igual)
>	maior que
≥	maior ou igual a
<	Menor que
≤	menor ou igual a

Conectores

e – para a conjunção
ou – para a disjunção
não – para a negação

Estrutura Condicional

- Uma estrutura de decisão ou estrutura condicional permite a escolha do grupo de ações a ser executado quando determinadas condições, representadas por expressões lógicas, são ou não satisfeitas.

Estrutura Condicional

São de 3 tipos

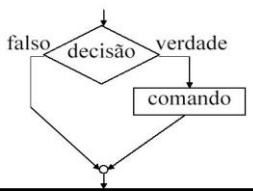
- ESTRUTURA CONDICIONAL SIMPLES
- ESTRUTURA CONDICIONAL COMPOSTA
- SELEÇÃO ENTRE DUAS OU MAIS SEQÜÊNCIAS DE COMANDOS

Estrutura Condicional

Estrutura Condicional Simples

Formas de Representação no Algoritmo

Fluxograma



Linguagem Algoritmica

se <condição>
  então <comando>
fim-se

Estrutura Condicional

Estrutura Condicional Simples

- O comando só será executado se a condição for verdadeira
- a condição deve ser uma expressão lógica
- se mais de um comando deve ser executado quando a condição for verdadeira, esses comandos devem ser transformados em um comando composto.

Estrutura Condicional

Comando Composto

- Um conjunto de comandos que devem ser executados em uma ordem específica.
- Os comandos devem ficar reunidos entre as palavras chaves **início** e **Fim**
- **início**
  comando 1;
  comando 2;
  comando 3;
  Fim

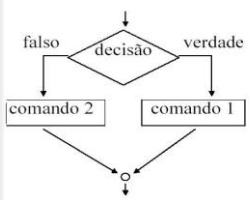
Comando composto

Estrutura Condicional

Estrutura Condicional Composta

Formas de Representação no Algoritmo

Fluxograma



Linguagem Algoritmica

se <condição>
  então <comando1>
  senão <comando2>
fim-se

Estrutura Condicional

Estrutura Condicional Composta

- Se condição for verdadeira será executado o comando 1 e não será executado o comando 2.
- Se condição for falsa será executado o comando 2 e não será executado o comando 1.
- a condição deve ser uma expressão lógica

Estrutura Condicional

Estrutura Condicional Composta

- se mais de um comando deve ser executado quando a condição for verdadeira ou quando a condição for falsa, esses comandos devem ser transformados em comandos compostos.

VisuALG – Condicional simples

sintaxe:

se *<condição>* **entao**
<sequência-de-comandos>
fimse

análise do comando:

- **se**, **entao** (sem o til) e **fimse** são palavras-chave do comando condicional;
- *condição* é uma expressão lógica que sendo verdadeira então a *sequência-de-comandos* é executada e sendo falsa o controle passará para o próximo comando após a palavra-chave **fimse**;

VisuALG – Condicional composta

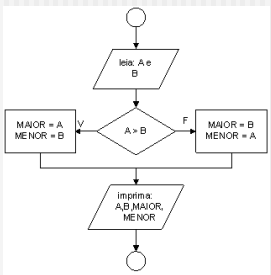
sintaxe:

se *<condição>* **entao**
<sequência-de-comandos-1>
senao
<sequência-de-comandos-2>
fimse

análise do comando:

- **senao** (sem o til), palavra-chave, além das já existentes no comando condicional simples;
- *condição* é uma expressão lógica que sendo verdadeira então a *sequência-de-comandos-1* é executada e sendo falsa a *sequência-de-comandos-2* é executada. Somente após executada uma das sequências o controle passará para o próximo comando, que segue a palavra-chave **fimse**;

Exemplo...



Fluxograma para encontrar o Maior e o Menor de dois números lidos

Exemplo... Em VisuALG

```
var
   A,B,MAIOR,MENOR: inteiro
inicio
   escreval("Digite 2 números inteiros")
   leia(A,B)
   se A>B  entao
     MAIOR <- A
     MENOR <- B
   senao
     MAIOR <- B
     MENOR <- A
   fimse
   escreval("Números lidos: ", A, " e ",B)
   escreval("Maior = ", MAIOR)
   escreval("Menor = ", MENOR)
fimalgoritmo
```

Estruturas de Decisão em C

- Permitir testes para decidir ações alternativas:
  - if
  - if - else
  - switch
  - (?:) Operador Condicional

Comando if

```
if (condição)
    instrução;
```

```
#include <stdio.h>
   main ( )
   {
    char ch;
    ch = getchar ( );
    if  (ch == 'p')
          printf ("você pressionou a
tecla p");
}
```

```
if (condição) {
    instrução1;
    intrução2;
   }
```

```
#include <stdio.h>
 main ( )
 {
   if  (getchar()) == 'p' ) {
      printf (" você digitou p");
      printf (" pressione outra tecla ");
              getchar( );
   }
 }
```

Comando if-else

- O comando **if** só executa a instrução caso a condição de teste seja verdadeira, nada fazendo se a expressão for falsa.
- O comando **else** executará uma instrução ou um conjunto de instruções se a expressão for falsa.

Comando if-else

```
if (condição)
   instrução;
else
   instrução;
```

```
#include <stdio.h>
main ( )
{
   if (getchar ( ) == 'p')
      printf (" você digitou p");
   else
      printf (" você não digitou p");
}
```

Exemplo: Evitar divisões por Zero, usando recursos do comando if-else.

```
include <stdio.h>
main()
{
int a,b;
printf("Digite 2 números: ");
scanf("%d %d",&a,&b);
if (b)
  printf("%f",a/b);
else
  printf("Nao posso dividir por zero\n");
}
```

Alguns detalhes Linguagem C

- Pré e Pós Incremento
- Comparações e testes
- Size of
- Conversões de Tipos

Comparações e Testes

- Observemos antes de mais nada que ++x é diferente de x++!

```
Se
   x = 10;
   y = ++x;
  /* x=x+1;   y=x; */
então
   x = 11 e
   y = 11
```

```
porém Se
   x = 10;
   y = x++;
   /* y=x;   x=x+1 */
então
    x = 11 e
    y = 10
```

Programa Exemplo – Pre/Pos Incremento

```
#include<stdio.h>
main()
{
int cont = 0, loop;
loop=++cont;
printf("Loop=%d, Cont=%d\n", loop, cont);

loop=cont++;
printf("Loop=%d, Cont=%d\n", loop, cont);
```

Quais são as saídas deste programa ?

Comparações e Testes

```
Se
  x = 1;
  y = 2;
  printf("%d == %d e' %d\n", x, y, x==y )
```

Qual seria a saída deste comando?
resultaria em `1 == 2  0`
(pois a expressão é falsa)

Comparações e Testes

```
if (10 > 4 && !(10 < 9) || 3 <= 4)
```

Como seria avaliado esta instrução?
resultaria em Verdadeiro, pois dez é maior que quatro E dez não é menor que nove OU três é menor ou igual a quatro

Operador Sizeof

- Este operador retorna o tamanho da variável ou tipo que está em seu operando.
- Por exemplo "sizeof(char)" resultaria em 1.

Conversões de Tipos

- Quando forem misturadas variáveis de diferentes tipos, o compilador C converterá os operandos para o tipo de operando maior, de acordo com as regras descritas a seguir:
- 1-Todo char e short int é convertido para int. Todo float é convertido para double.
- 2-Para os demais pares de operandos valem as seguintes regras em seqüência:
  - 2.1- Se um operando for long double, o outro também o será.
  - 2.2- Se um operando for double, o outro também o será.
  - 2.3- Se um operando for long, o outro também o será.
  - 2.4- Se um operando for unsigned, o outro também o será.

Conversões de Tipos - NOTA

- **Nota:** Devemos observar que o compilador C é bastante flexível e pouco vigilante, comportando-se de maneira muito diferente de um compilador Clipper ou Pascal, sempre vigilantes com relação aos tipos das variáveis. De fato aqueles compiladores podem gerar executáveis misturando tipos, porém a ocorrência de erros de execução é quase inevitável. Ao contrário destes compiladores, os compiladores C "ajeitam" as coisas para o programa funcionar da "melhor maneira possível", o que não significa em hipótese alguma que os resultados serão os esperados por programadores "relapsos". Assim esta boa característica dos compiladores C, pode transformar-se numa autêntica "bomba relógio" para programas não muito bem elaborados.

EXERCÍCIOS

EXERCÍCIO 1

FAZER UM ALGORITMO QUE LEIA DOIS NÚMEROS E ESCREVA (DEVOLVA COMO RESULTADO) O MENOR DELES.

EXERCÍCIO 2

Faça um algoritmo que receba três notas de um aluno, calcule e mostre a média aritmética e a mensagem que segue a tabela abaixo. Para alunos de exame, calcule e mostre a nota que deverá ser tirada no exame para aprovação, considerando que a média do exame é 6,0 (e que a média após o exame é resultado da média aritmética entre a nota do exame e a média antes do exame).

	Mensagem
0,0 e <3,0	*Reprovado*
>= 3,0 e < 7,0	*Exame*
>=7,0 e <=10,0	*Aprovado*

EXERCÍCIO 3

Faça um programa que receba um número inteiro e verifique se esse número é par ou ímpar.

EXERCÍCIO 4

(FORBELLONE; EBERSPÄCHER, 2000 - pág. 46) Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um algoritmo que calcule seu peso ideal, utilizando as seguintes fórmulas:

- **para homens: (72.7 * h) – 58;**
- **para mulheres: (62.1 * h) – 44.7.**

Fonte: Slides da Aula 6 do Prof. Dr. José Nelson Falavinha Junior 40

EXERCÍCIO 5

(FORBELLONE; EBERSPÄCHER, 2000 - pág. 46) Faça um algoritmo que leia o ano de nascimento de uma pessoa, calcule e mostre sua idade e, também, verifique e mostre se ela já tem idade para votar (16 anos ou mais) e para conseguir a Carteira de Habilitação (18 anos ou mais).

Fonte: Slides da Aula 6 do Prof. Dr. José Nelson Falavinha Junior 41