



INTRODUÇÃO À PROGRAMAÇÃO

Estrutura de Decisão

Simples

Profº. Sérgio Roberto Costa Vieira, M.Sc.

Cursos de Computação

1º. Período

Introdução à Programação

Estrutura de Decisão ou Seleção

Estruturas de Seqüência

Está essencialmente inserida no conceito de algoritmos e foi o que vimos até agora.

A menos que seja especificado de outra forma, as instruções serão executadas segundo a estrutura de seqüência, isto é, uma após a outra, de cima para baixo.

Introdução à Programação

Estrutura de Decisão ou Seleção

Estruturas de Seqüência

Existem casos em que é preciso que um conjunto de instruções seja executada somente se determinada condição for satisfeita.

Para isso utiliza-se Estruturas de Seleção ou Estruturas de Controle Condicional ou Estruturas de Decisão.

Introdução à Programação

Estrutura de Decisão ou Seleção

Estruturas de Controle

- ✓ São utilizadas para especificar a ordem em que as instruções devem ser executadas.

Tipos de estruturas de controle:

1. Estrutura de seqüência
2. Estruturas de seleção
3. Estruturas de repetição

Introdução à Programação

Estrutura de Decisão ou Seleção

Estruturas de Seleção

São estruturas de controle de fluxo, que executam um ou vários comandos se a condição testada for verdadeira.

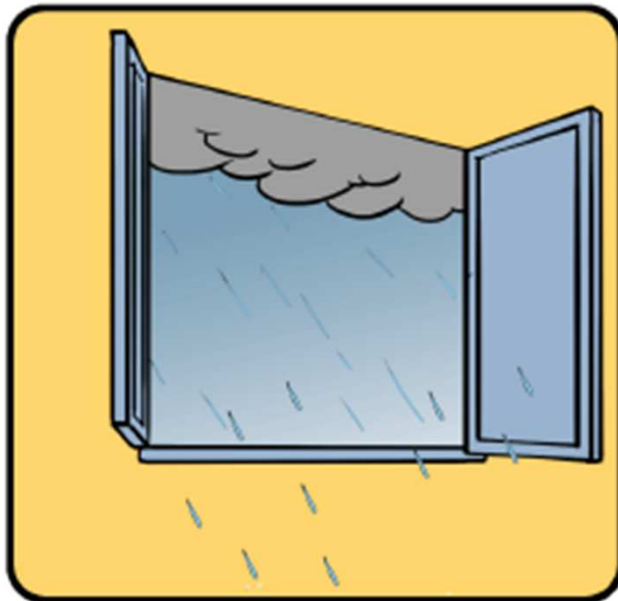
Permitem a tomada de decisão sobre qual o caminho a ser escolhido, de acordo com o resultado de uma expressão lógica.

Introdução à Programação

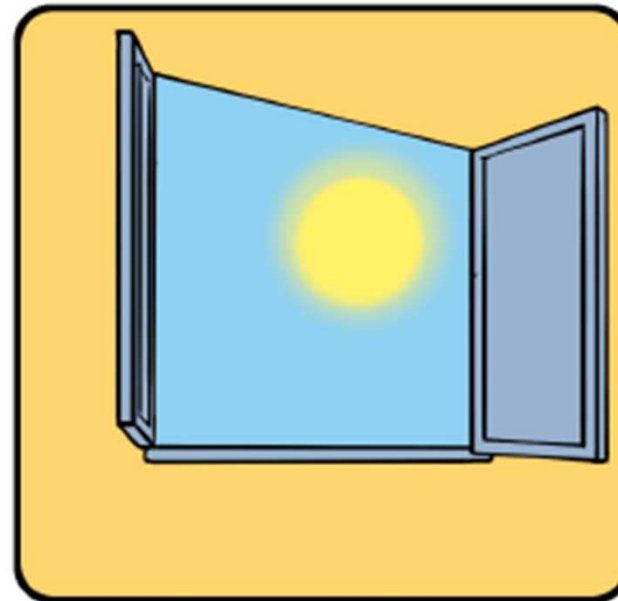
Estrutura de Decisão ou Seleção

Estruturas de Seleção – Exemplo

Se estiver chovendo **então** tenho
que fechar as janelas do quarto.



Caso contrário (**senão**)
as janelas podem ficar abertas.



Introdução à Programação

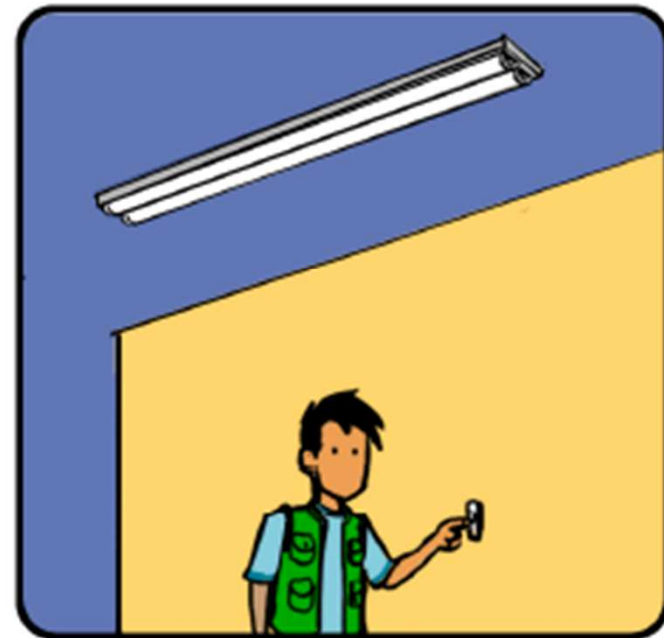
Estrutura de Decisão ou Seleção

Estruturas de Seleção – Exemplo

Se estiver escuro **então** tenho
que ligar as lâmpadas de casa.



Caso contrário (**senão**) as lâmpadas
podem ficar desligadas.



Introdução à Programação

Estrutura de Decisão ou Seleção

Estruturas de Seleção

Têm como característica diferenciada das estruturas sequenciais, a **avaliação da veracidade de uma condição** (representada por expressões com operadores relacionais e operadores lógicos), a qual **permite ou não um bloco de comandos ser executado**.

Introdução à Programação

Estrutura de Decisão ou Seleção

Definindo uma Condição

Entende-se qualquer expressão relacional (ou seja, que use os operadores $>$, $<$, $>=$, $<=$, $=$ ou $<>$, OU, E, Não) que resulte em uma resposta do tipo verdadeira ou falsa.

Exemplo: para a condição $(X > 0)$ temos que:

- ✓ Se o valor de X for um valor POSITIVO, a condição será considerada VERDADEIRA.
- ✓ Se o valor de X for igual a ZERO ou NEGATIVO, a condição será considerada FALSA.

Introdução à Programação

Estrutura de Decisão ou Seleção

Definindo uma Condição

Uma expressão condicional é qualquer expressão que resulte em uma resposta do tipo verdadeiro ou falso.

Utilizando os operadores:

- ✓ Matemáticos: +, -, *, /, %
- ✓ Relacionais: >, <, >=, <=, =, <>
- ✓ Lógicos: e, ou, não

Introdução à Programação

Estrutura de Decisão ou Seleção

Definindo uma Condição

Uma expressão condicional pode utilizar operadores dos tipos matemático, relacional e/ou lógico.

- ✓ X é maior ou igual a Y? $X \geq Y$
- ✓ X é maior do que Y+2? $X > Y+2$
- ✓ X – 5 é diferente de Y+3? $X-5 \neq Y+3$
- ✓ X é maior que Y e menor que Z? $(X > Y) \text{ e } (X < Z)$
- ✓ X é menor que 1 ou maior que 10? $(X < 1) \text{ ou } (X > 10)$

Introdução à Programação

Estrutura de Decisão ou Seleção

Os comandos de decisão ou desvio fazem parte das técnicas de programação que conduzem a estruturas de programas que não são totalmente seqüenciais.

Existem três formas básicas desse tipo de estrutura:

- **Estrutura Condicional Simples**
- **Estrutura Condicional Composta**
- **Estrutura Condicional de Múltipla Escolha**

Introdução à Programação

Estrutura de Decisão ou Seleção

A classificação das estruturas de decisão são feitas de acordo com o número de possibilidades de decisões que devem ser testadas para que se decida qual o caminho a ser seguido.

Existem 3 tipos de estruturas de decisão:

- Estrutura de Decisão Simples
 - *(Se ... Então ... Fimse)*
- Estrutura de Decisão Composta
 - *(Se ... Então ... Senão ... Fimse)*
- Estrutura de Decisão Múltipla do Tipo Escolha
 - *(Escolha ... Caso ... Senão ... FimEscolha)*

Introdução à Programação

Estrutura de Decisão ou Seleção

Estrutura de Decisão Simples

É representada por um comando que avalia uma expressão lógica (**condição**), resultando um valor que pode ser *true* (**verdadeiro**) ou *false* (**falso**).

se (**condição**) então

sequência de instruções

fimse

if (**condição**) {

sequência de comandos

}

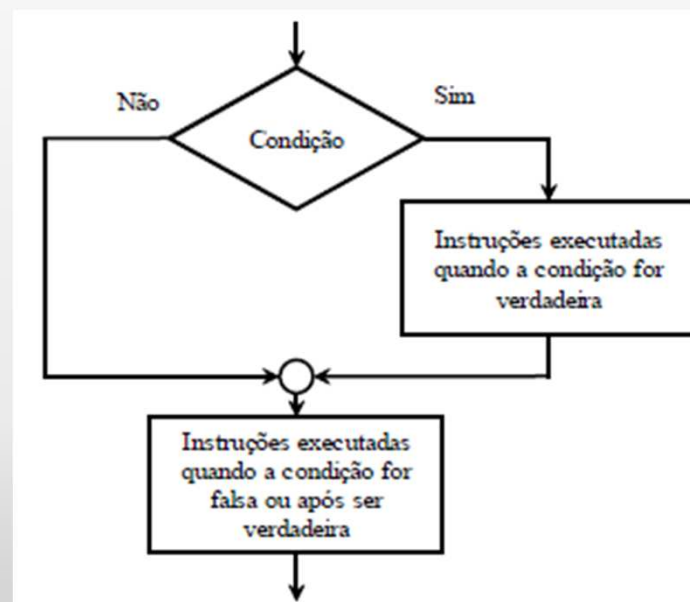
Isso significa que a execução das instruções dependem do fato da condição ser verdadeira

Introdução à Programação

Estrutura de Decisão ou Seleção

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).



Introdução à Programação

Estrutura de Decisão ou Seleção

A estrutura de decisão “**SE**” normalmente vem acompanhada de um comando, ou seja, se determinada condição for satisfeita pelo comando **SE** então execute determinado comando.

Imagine um algoritmo que determinado aluno somente estará aprovado se sua média for maior ou igual a 5.0, veja no exemplo de algoritmo como ficaria.

se (media >= 5.0) então

 escreva(“Aluno Aprovado”);

fimse

if (media >= 5.0) {

 printf(“Aluno Aprovado”);

}

Introdução à Programação

Exemplo 1

Algoritmo Aluno;

Declare p1, p2, media : real;

media \leftarrow 0;

escreva(" entre com 1ª. nota");

leia (p1);

escreva(" entre com 2ª. nota");

leia (p2);

media \leftarrow (p1+p2) / 2;

se (media \geq 5) então

escreva ("Aluno Aprovado!");

fimse

FimAlgoritmo

Declaração
de Variáveis

Comandos
de Entrada

Processamento

Decisão ou
Seleção

Introdução à Programação

Exemplo 1

Algoritmo Aluno;

Declare p1, p2, media : real;

media \leftarrow 0;

escreva(" entre com 1ª. nota");

leia (p1);

escreva(" entre com 2ª. nota");

leia (p2);

media \leftarrow (p1+p2) / 2;

se (media \geq 5) então

escreva ("Aluno Aprovado!");

fimse

se (media < 5) então

escreva ("Aluno Reprovado!");

fimse

FimAlgoritmo

PORTUGOL

**Decisão ou
Seleção**

Introdução à Programação

Exemplo 1

```
int main(){
    float p1, p2, media=0 ;
    printf(" entre com 1ª. nota");
    scanf ("%f", &p1);
    printf(" entre com 2ª. nota");
    scanf ("%f", &p2);
    media = (p1+p2) / 2;
    if (media >= 5) {
        printf ("Aluno Aprovado!");
    }
    if (media < 5) {
        printf ("Aluno Reprovado!");
    }
    return 0;
}
```

LINGUAGEM C

**Decisão ou
Seleção**

Introdução à Programação

Exemplo 2

Algoritmo para ler dois números e escrever “Números iguais” se os dois forem iguais.

Algoritmo NUM;

Declare A, B : inteiro;

Declaração
de Variáveis

escreva(“ entre com 1º numero ”);

leia (A);

Comandos
de Entrada

escreva(“ entre com 2º numero ”);

leia (B);

se (A = B) então

Decisão ou
Seleção

escreva (“Números iguais”);

fimse

FimAlgoritmo

Introdução à Programação

Exemplo 2

Algoritmo NUM;

PORTUGOL

Declare A, B : inteiro;

escreva(“ entre com 1º numero ”);

leia (A);

escreva(“ entre com 2º numero ”);

leia (B);

se (A = B) então

escreva (“Números iguais”);

fimse

se (A <> B) então

escreva (“Números diferentes”);

fimse

FimAlgoritmo

**Decisão ou
Seleção**

Introdução à Programação

Exemplo 2

```
int main() {  
    int A, B ;  
    printf(" entre com 1º numero ");  
    scanf ("%d", &A);  
    printf(" entre com 2º numero ");  
    scanf ("%d", &B);  
    if (A == B) {  
        printf ("Números iguais");  
    }  
    if (A != B) {  
        printf ("Números diferentes");  
    }  
    return 0; }
```

LINGUAGEM C



**Decisão ou
Seleção**

Introdução à Programação

Exemplo 3

Algoritmo que lê um número inteiro e escreva “Número Válido” se este for um número que estiver entre 10 e 30.

Algoritmo NUM_10_30;

Declaração
de Variáveis

Declare A : inteiro;

escreva(“ entre com 1º numero ”);

Comandos
de Entrada

leia (A);

se (A >= 10) e (A <=30) então

escreva (“Número Válido”);

fimse

Decisão ou
Seleção

FimAlgoritmo

Introdução à Programação

Exemplo 3

Algoritmo NUM_10_30;

PORTUGOL

Declare A : inteiro;

escreva(“ entre com 1º numero ”);

leia (A);

se (A >= 10) e (A <= 30) então

escreva (“Número Válido”);

fimse

se (A < 10) ou (A > 30) então

escreva (“Número Inválido”);

fimse

FimAlgoritmo

Introdução à Programação

Exemplo 3

LINGUAGEM C

```
int main() {  
    int A ;  
    printf(" entre com 1º numero ");  
    scanf ("%d", &A);  
    if (A >= 10) && (A <= 30) {  
        printf ("Número Válido");  
    }  
    if (A < 10) || (A > 30) {  
        printf ("Número Inválido");  
    }  
    return 0;  
}
```

Introdução à Programação

Exemplo 4

```
Algoritmo_Idade;  
Declare ID : inteiro;  
    escreva(" Informe a sua idade ");  
    leia (ID);  
    se (ID > 0) e (ID < 18) então  
        escreva ("Menor de Idade");  
    fimse  
    se (ID >= 18) e (ID < 65) então  
        escreva ("Maior de Idade");  
    fimse  
    se (ID >= 65) então  
        escreva ("Idoso");  
    fimse  
FimAlgoritmo
```

Algoritmo que leia a idade de uma pessoa e informar se ela é maior de idade, menor de idade ou idosa.

Introdução à Programação

Exemplo 4

Algoritmo_Idade;

PORTUGOL

Declare ID : inteiro;

escreva(“ Informe a sua idade ”);

leia (ID);

se (ID <= 0) então

escreva (“Idade Inválida”);

fimse

se (ID > 0) e (ID < 18) então

escreva (“Menor de Idade”);

fimse

se (ID >= 18) e (ID < 65) então

escreva (“Maior de Idade”);

fimse

se (ID >= 65) então

escreva (“Idoso”);

fimse

FimAlgoritmo

Introdução à Programação

Exemplo 4

LINGUAGEM C

```
int main() {  
    int ID ;  
    printf(" Informe a sua idade ");  
    scanf ("%d", &ID);  
    if (ID <= 0) {  
        printf ("Idade Inválida");  
    }  
    if (ID > 0) && (ID < 18) {  
        printf ("Menor de Idade");  
    }  
    if (ID >= 18) && (ID < 65) {  
        printf ("Maior de Idade");  
    }  
    if (ID >= 65) {  
        printf ("Idoso");  
    }  
    return 0; }
```


Introdução à Programação

Expressões de Algoritmos

- **Expressões Relacionais**

- são aquelas que realizam a comparação entre dois valores de um mesmo tipo.

Como operadores relacionais temos:

Descrição	Símbolo
Igual a	=
Diferente de	<> ou #
Maior que	>
Menor que	<
Maior ou igual a	>=
Menor ou igual a	<=

Introdução à Programação

Expressões de Algoritmos

- **Expressões Lógicas**

- expressões booleanas, utiliza operadores lógicos e suas relações entre variáveis são do tipo lógico.

Como operadores lógicos temos:

E	AND
OU	OR
NÃO	NOT

1º Valor	Operador	2º Valor	Resultado
T	AND	T	T
T	AND	F	F
F	AND	T	F
F	AND	F	F
T	OR	T	T
T	OR	F	T
F	OR	T	T
F	OR	F	F
T	NOT		F
F	NOT		T

Introdução à Programação

Expressões de Algoritmos

- Expressões Lógicas**

Prioridade entre os operadores

3°

2°

1°

A	B	A ou B	A e B	não A
falso	falso	falso	falso	verdade
falso	verdade	verdade	falso	verdade
verdade	falso	verdade	falso	falso
verdade	verdade	verdade	verdade	falso

Introdução à Programação

Exercício de Fixação

- 1 – Faça um algoritmo que leia um número inteiro e escreva “MAIOR” se ele for maior que 10;
- 2 – Faça um algoritmo que leia um salário de um funcionário, se este for menor que R\$ 700.00, calcule e mostre o percentual de 8% referente ao salário família :
- 3 – Faça um algoritmo que leia o nome de uma pessoa, altura e seu sexo, verifique se é do sexo masculino e calcule seu peso ideal, informe no final o peso. ($\text{peso_ideal} = (72.7 * \text{altura}) - 58$):
- 4 – Faça um algoritmo que leia três notas de um aluno, calcule a média e informe se o aluno foi Aprovado (≥ 7.5), se o aluno foi Reprovado (< 5.0) ou se o aluno irá para a Recuperação.
- 5 – Faça um algoritmo que leia um nome e escreva se for igual a “Maria”. Veja as várias formas de escrever este nome:

Para Próxima Aula

NOVOS EXERCÍCIOS

em Portugal e Linguagem C

Introdução à Programação

Exercício de Fixação

- 1 – Faça um algoritmo que leia um número inteiro e, somente se, ele for um número par, imprimir sua metade;
- 2 – Faça um algoritmo que leia a quantidade de horas trabalhadas no mês de um funcionário, se for maior que 180 calcular e verificar quantas horas extras ele fez e informar quanto receberá no final do mês de horas extras e seu salário, a hora dele custa R\$ 14,20:
- 3 – Faça um algoritmo que leia duas notas de um aluno, calcule a média desse aluno. Se a média for menor que 5, acrescentar um ponto extra de 0.3 para cada um dos 3 exercícios que ele fez, no final informe a média:
- 4 – Faça um algoritmo que leia o valor de um veículo à venda, se o comprador for funcionário público, aplique um desconto de 15% no valor do veículo. No final mostre o valor do veículo e o desconto recebido:

Introdução à Programação

Exercício de Fixação

- 5 – Dados dois números A e B, digitados pelo usuário, escreva-os em ordem crescente:
- 6 – Dados dois números X e Y, digitados pelo usuário, se estiverem em ordem crescente, trocar o valor de X por Y e vice-versa:
- 7 – O usuário deverá digitar um número e o programa informará se o número é positivo ou negativo (não considere o 0 como positivo):
- 8 – Maças possuem o valor de custo normal de R\$ 0,45 , mas custam R\$ 0,30 se forem compradas mais que uma dúzia. Escreva um programa em pseudocódigo que leia o número de maçãs compradas, calcule e escreva na tela o valor total da compra:

Introdução à Programação

Exercício de Fixação

- 9 – Crie um algoritmo que leia uma senha, se a senha estiver correta exibir uma mensagem de boas vindas. Caso esteja incorreta exibir uma mensagem de erro (considere a senha 1234):
- 10 – Um vendedor ganha um prêmio por vendas realizadas. A cada 200 vendas no mês, o mesmo ganha o prêmio de 6% como comissão. Faça um algoritmo que calcule e escreva quanto é o valor do prêmio:
- 11 – Faça um algoritmo que leia um número inteiro e verifique se o número é múltiplo de 2, se for escreva o número e o seu dobro:
- 12 – Faça um algoritmo que leia uma letra informada pelo usuário, se for uma vogal, escreva a mensagem que “é uma vogal”:

Introdução à Programação

Exercício de Fixação

- 13 – Crie um algoritmo que leia a quantidade de horas trabalhadas por um funcionário e seu turno de trabalho, se for do turno noturno, acrescentar o valor de 15% sobre seu salário. No final exibir o salário:
- 14 – Faça um algoritmo que leia um número inteiro, se este estiver entre 25 e 60, calcule a diferença pelo valor 100. No final exibir o número:
- 15 – Faça um algoritmo que leia o preço de um medicamento, se este ultrapassar o valor de R\$ 100, aplicar o desconto de 12% sobre o valor. Imprimir o valor final a ser pago:
- 16 – Escreva um algoritmo para ler o nome de 2 times e o número de gols marcados (suponha valores diferentes). Escrever o nome do vencedor:

Introdução à Programação

Exercício de Fixação

- 17 – Faça um algoritmo que leia um número qualquer e verifique se este número é divisível por 5 ou não:
- 18 – Ler o ano atual e o ano de nascimento de uma pessoa. Escrever uma mensagem que diga se ela poderá ou não votar este ano (não é necessário considerar o mês em que a pessoa nasceu):
- 19 – Ler o salário fixo e o valor das vendas efetuadas pelo vendedor de uma empresa. Sabendo-se que ele recebe uma comissão de 3% sobre o total das vendas até R\$ 1.500,00 mais 5% sobre o que ultrapassar este valor, calcular e escrever o seu salário total:
- 20 – Faça um algoritmo para ler: número da conta do cliente, saldo, débito e crédito. Após, calcular e escrever o saldo atual (saldo atual = saldo - débito + crédito). Também testar se saldo atual for maior ou igual a zero escrever a mensagem 'Saldo Positivo', senão escrever a mensagem 'Saldo Negativo':.



INTRODUÇÃO À PROGRAMAÇÃO

Estrutura de Decisão

Simples

Profº. Sérgio Roberto Costa Vieira, M.Sc.

Cursos de Computação

1º. Período