

Algoritmos

Estruturas de controle

Prof. Fabio Takeshi Matsunaga

SENAI Londrina

6 de março de 2019

Objetivos

Missão da aula

- As principais estruturas de controle de um algoritmo.
 - Estrutura sequencial.
 - Estrutura de decisão.
- Apresentar a aplicabilidade das estruturas de controle no dia-a-dia.

Estruturas de controle

- Sabe-se que em algoritmos existem comandos de entrada, saída, atribuições e expressões aritméticas.
- Estes comandos representam um conjunto de ações para resolver um problema.
- É necessário existir uma relação lógica entre cada uma das ações a serem executadas.
- Essas relações são representadas pelas **estruturas de controle** do algoritmo.

Estruturas de controle

- Sequencial.
- Decisão.
- Repetição.

Estrutura sequencial

- A estrutura sequencial de um algoritmo consiste na execução linear do conjunto de comandos e ações.
- A sequência é lida do primeiro comando até o último, isto é, de cima para baixo e da esquerda para direita.

Algoritmo sequencial

Estrutura de um algoritmo sequencial

Algoritmo "Sequencial"

Var

// Declaração de variáveis

Inicio

Ação 1

Ação 2

Ação 3

⋮

Ação n

Fimalgoritmo

Exemplo de algoritmo sequencial

Construa um algoritmo que calcule a quantidade de latas de tinta necessárias e o custo para pintar tanques cilíndricos de combustível, em que são fornecidos a altura e o raio deste cilindro. Sabe-se que:

- A lata de tinta custa \$ 50,00.
- Cada lata tem 5 litros.
- Cada litro de tinta pinta 3 metros quadrados.

Planejamento reverso

- Planejamento reverso é a técnica para definir as etapas do processamento a partir das informações de entrada e saída.
- Técnica útil para construir algoritmos.

Algoritmo sequencial

Solução

Algoritmo "Tinta"

Var

H, R: **real**

C, QTD, ÁREA, LITRO: **real**

Inicio

leia(H,R)

AREA <- $(3,14 * \text{Exp}(R,2)) + (2 * 3,14 * R * H)$

LITRO <- ÁREA / 3

QTD <- LITRO / 5

C <- QTD * 50

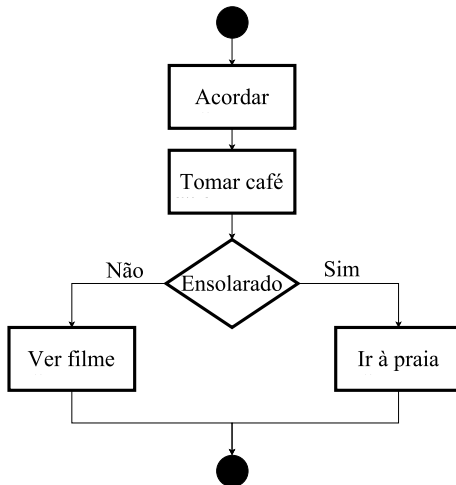
escreva(C,QTD)

Fimalgoritmo

Estrutura de decisão

- Até agora vimos que um problema pode ser resolvido através de um conjunto de ações executadas na sequência.
- Porém, nem todas as ações do nosso dia-a-dia são feitas de modo linear.
- Em muitos momentos, fazemos ações que dependem de uma determinada condição para ser executada.

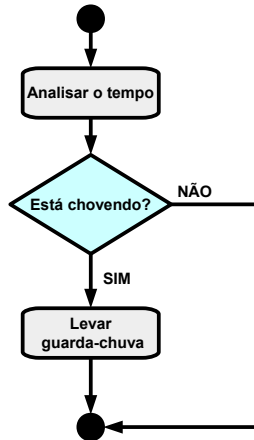
Exemplo de decisão



Estrutura de decisão

- Em muitos momentos da vida, precisamos de uma condição antes de executar uma ação.
- Não é sempre que a gente sai de casa com guarda-chuva.
- Para ir para a faculdade/trabalho com blusa ou jaqueta, temos que analisar o ambiente.
- Um aluno é aprovado em uma matéria somente se cumprir com todos os requisitos.

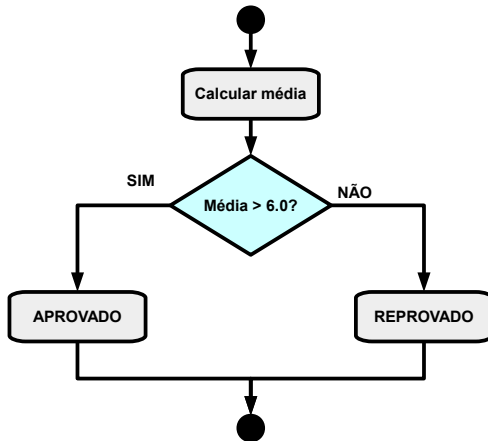
Exemplo de decisão



Exemplo de decisão



Exemplo de decisão



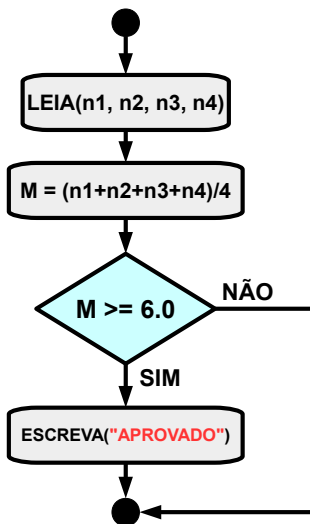
Seleção simples

- Serve para testar uma única condição antes de executar uma ação.
- A <condição> é uma expressão lógica que é testada, cujo resultado pode ser apenas **verdadeiro** ou **falso**.
- Se a condição for verdadeira, então a sequência de comandos S_1 será executada.

Estrutura básica de um comando de seleção simples

```
SE <condição> ENTÃO  
    // Sequência de comandos  $S_1$   
FIMSE
```


Exemplo de decisão



Operadores relacionais

- **Relacionais** – efetua comparação entre variáveis, retornando 1 se a expressão for verdadeira ou 0 se for falsa
 - Maior que ($x > y$): Verifica se x é maior que y ;
 - Menor que ($x < y$): Verifica se x é menor que y ;
 - Maior ou igual a ($x \geq y$): Verifica se x é maior ou igual que y ;
 - Menor ou igual a ($x \leq y$): Verifica se x é menor ou igual que y ;
 - Igual ($x = y$): Verifica se x é igual a y ;
 - Diferente ($x \neq y$): Verifica se x é diferente de y ;

Operadores lógicos

- **Lógicos** – utilizado em expressões que retornam valores verdadeiro ou falso de acordo com a lógica booleana (considera-se que os dois operandos x e y sejam do tipo `bool` ou provenientes de expressões relacionais ou lógicas)
 - AND (x E y): resulta um valor VERDADEIRO (1) se os dois valores das variáveis forem VERDADEIROS (1) e FALSO (0) caso contrário;
 - OR (x OU y): resulta um valor VERDADEIRO (1) se pelo menos um dos valores forem VERDADEIROS (1) e FALSO (0) caso contrário;
 - NOT (NÃO x): operador lógico unário que inverte os valores, isto é, se for VERDADEIRO (1) retorna FALSO (0), e vice-versa;

Operadores lógicos

- Os operadores relacionais e lógicos sempre manipulam ou resultam dados booleanos, isto é, valores que assumem somente dois estados: 1 (verdadeiro ou `true`) ou 0 (falso ou `false`).
- Em muitos algoritmos, ambos os operadores são utilizados em conjunto.

Operação E (AND) lógico

Somente resulta em verdadeiro (1) se ambos os operandos forem verdadeiros.

Tabela: Tabela-verdade da operação E lógico.

A	B	R
0	0	0
0	1	0
1	0	0
1	1	1

Operação OU (OR) lógico

É verdadeiro (1) quando pelo menos um dos operandos for verdadeiro.

Tabela: Tabela-verdade da operação OU lógico.

A	B	R
0	0	0
0	1	1
1	0	1
1	1	1

Operação NÃO (NOT) lógico

Inverte o resultado de uma expressão lógica: verdadeiro (1) torna-se falso (0) e falso (0) torna-se verdadeiro (1).

Tabela: Tabela-verdade da operação NÃO.

A	R
0	1
1	0

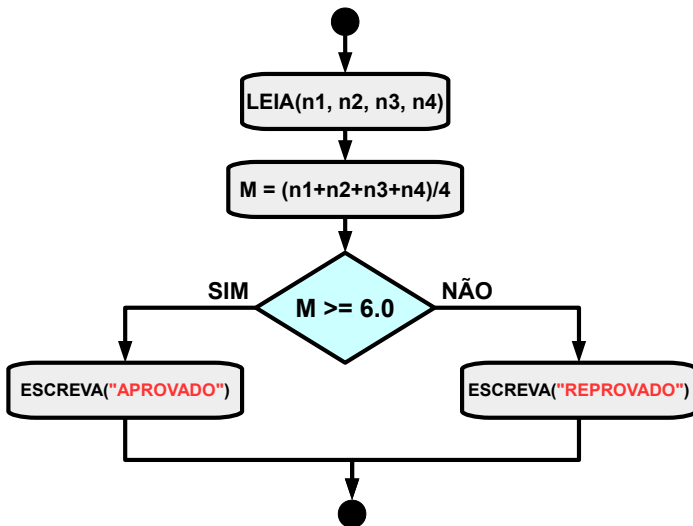
Seleção composta

- Algumas situações possuem duas alternativas ou opções de saída para uma mesma condição.
- Nestas opções, uma é para a saída verdadeira e outra para a saída falsa.
 - Se a <condição> for verdadeira, a sequência de comandos S_1 será executado.
 - Se a <condição> for falsa, a sequência de comandos S_2 será executado.

Estrutura básica de um comando de seleção composta

```
SE <condição> ENTÃO
    // Sequência de comandos  $S_1$ 
SENÃO
    // Sequência de comandos  $S_2$ 
FIMSE
```

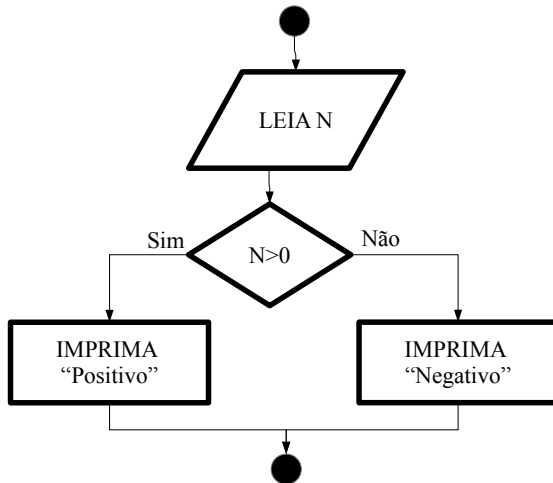

Exemplo de decisão



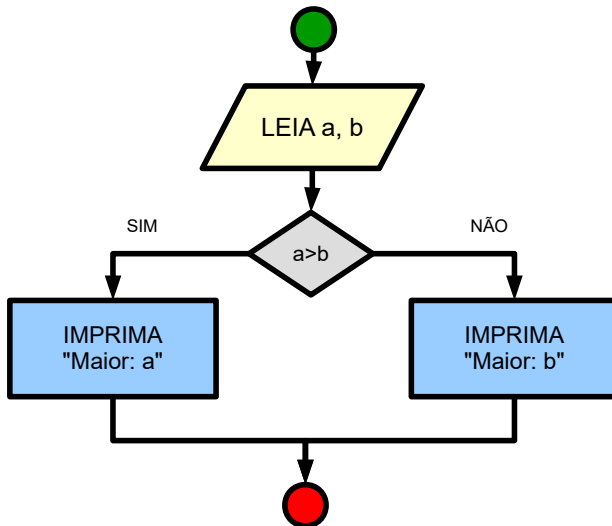
Exemplos

- 1 Faça um algoritmo que leia um número e determine se esse número é positivo ou negativo.
- 2 Faça um algoritmo que leia dois números. Em seguida, determine qual dos dois números é o maior.
- 3 Faça um algoritmo que leia um número e determina se o número é par ou ímpar.

Exemplo 1



Exemplo 2



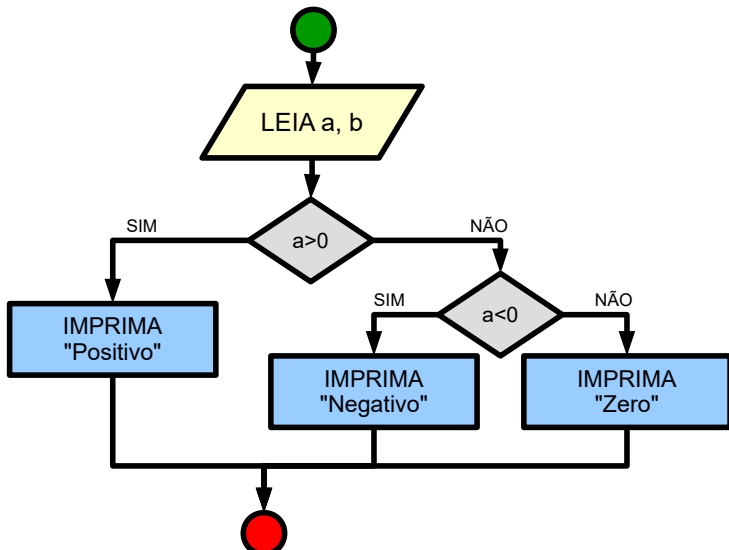
Seleção encadeada

- Existem situações que são agrupadas várias seleções por terem múltiplas opções (mais de duas) para um problema.
- São combinadas múltiplas condições de modo que apenas uma é satisfeita.

Estrutura básica de um comando de seleção composta

```
SE <condição 1> ENTÃO
    // Sequência de comandos  $S_1$ 
SENÃO
    SE <condição 2> ENTÃO
        // Sequência de comandos  $S_2$ 
    SENÃO
        // Sequência de comandos  $S_3$ 
    FIMSE
FIMSE
```

Exemplo 1 com estrutura composta



Considerações finais

- Os fluxos de decisão também são aplicados na construção de programas de computadores.
- Para isso existem os algoritmos de decisão ou seleção.
- Estes algoritmos procuram executar uma ação com base em uma condição definida.

OBRIGADO!

Contato

- **Fabio Takeshi Matsunaga**
- **E-mail:** fabio.matsunaga@sistemafiep.org.br