

Constantes, Variáveis, Tipos de Dados e Operadores



Estrutura Sequencial

Definição de Estrutura Sequencial

A estrutura sequencial é aquela na qual uma ação (instrução) segue a outra em seqüências. As tarefas se sucedem de tal modo que a saída de uma é a entrada da seguinte e assim sucessivamente até o fim do processo.

Exemplo: *Inicio*

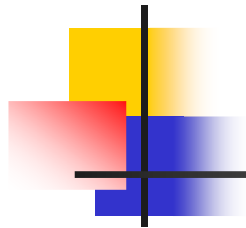
Ação 1

Ação 2

Ação 3

Ação n

Fim



Constantes

Entendemos que um dado é constante quando não sofre nenhuma variação no decorrer do tempo, ou seja, seu valor é constante desde o início até o fim da execução do algoritmo.

Constantes de caracter são envolvidas por aspas simples ('). Por exemplo, 'a' e '\$' são constantes tipo caracter. Constantes de string são envolvidas por aspas duplas ("). Por exemplo, "Pare" e "Não fume".

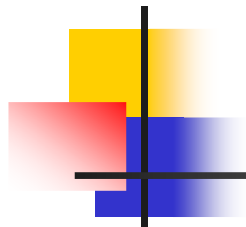
Outros exemplos de constantes:

5, "Parabéns", 23358, -0.256

Outro exemplo de constante:

$$\frac{N1+N2+N3}{3}$$

← CONSTANTE

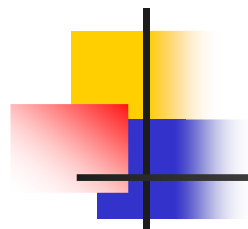


Operadores

Operadores

Os operadores são meios pelo qual incrementamos, decrementamos, comparamos e avaliamos dados dentro do computador. Temos três tipos de operadores:

- Operadores Aritméticos
- Operadores Relacionais
- Operadores Lógicos



Operadores Aritméticos

Operadores Aritméticos

Os operadores aritméticos são os utilizados para obter resultados numéricos. Além da adição, subtração, multiplicação e divisão, podem utilizar também o operador para exponenciação. Os símbolos para os operadores aritméticos são:

OPERAÇÃO	SÍMBOLO
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Exponenciação	**

Hierarquia das Operações Aritméticas

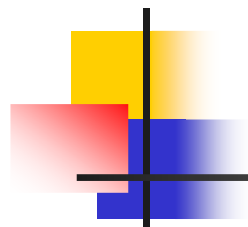
- 1º () Parênteses
- 2º Exponenciação
- 3º Multiplicação, divisão (o que aparecer primeiro)
- 4º + ou - (o que aparecer primeiro)

Exemplo

TOTAL = PRECO * QUANTIDADE

$1 + 7 * 2 ** 2 - 1 = 28$

$3 * (1 - 2) + 4 * 2 = 5$



Operadores Relacionais

Operadores Relacionais

Os operadores relacionais são utilizados para comparar **String** de caracteres e números. Os valores a serem comparados podem ser caracteres ou variáveis.

Estes operadores sempre retornam valores lógicos (verdadeiro ou falso/ True ou False)

Para estabelecer prioridades no que diz respeito a qual operação executar primeiro, utilize os parênteses.

Os operadores relacionais são:

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



Operadores Lógicos

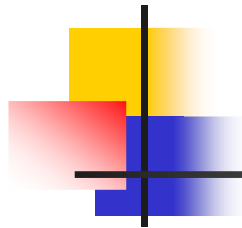
Os operadores lógicos servem para combinar resultados de expressões, retornando se o resultado final é verdadeiro ou falso.

Os operadores lógicos são:

E	AND
OU	OR
NAO	NOT

E / AND	Uma expressão AND (E) é verdadeira se todas as condições forem verdadeiras
OR/OU	Uma expressão OR (OU) é verdadeira se pelo menos uma condição for verdadeira
NOT	Um expressão NOT (NÃO) inverte o valor da expressão ou condição, se verdadeira inverte para falsa e vice-versa.

A tabela abaixo mostra todos os valores possíveis criados pelos três operadores lógicos (AND, OR e NOT)



Operadores Lógicos – cont.

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

Exemplos:

Suponha que temos três variáveis A = 5, B = 8 e C = 1

Os resultados das expressões seriam:

Expressões			Resultado
A = B	AND	B > C	Falso
A <> B	OR	B < C	Verdadeiro
A > B	NOT		Verdadeiro
A < B	AND	B > C	Verdadeiro
A >= B	OR	B = C	Falso
A <= B	NOT		Falso



Operadores - Exemplos.

Exemplo 1:

2	<	5	e	15/3	=	5
		V	e	5	=	5
		V	e	V		
			V			

=====

Exemplo 2:

2	<	5	ou	15/3	=	5
		V	ou	5	=	5
		V	ou	V		
			V			

=====

Exemplo 3:

F	ou	20 resto (18/3) <> 7 resto 2
F	ou	20 resto 6 <> 7 resto 2
F	ou	2 <> 1
F	ou	V
		V



Operadores - Exemplos.

Exemplo 4:

Não V	ou	$9/3 < 15 - 0$
F	ou	$3 < 15$
F	ou	V
	V	

=====

Exercício:

Não $(5 <> 10 / 2 \text{ ou } V \text{ e } 2 - 5 > 5 - 2 \text{ ou } V)$

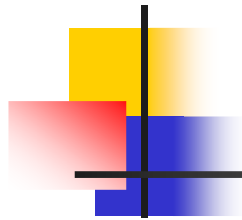
$2 < 5 \text{ e } 15 / 3 = 5$

$2 < 5 \text{ ou } 15 / 3 = 5$

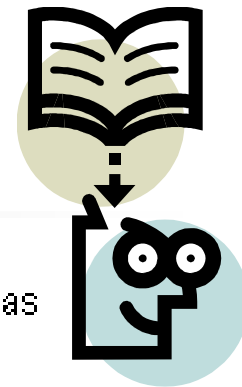
$F \text{ ou } 20 / (18 / 3) <> (21 / 3) / 2$

Não V ou $\text{pot}(3,2) / 3 < 15 - 35 \bmod 7$

Obs.: pot = potenciação e mod = resto da divisão.



Exercícios



- 1) Tendo as variáveis SALARIO, IR e SALLIQ, e considerando os valores abaixo. Informe se as expressões são verdadeiras ou falsas.

SALARIO	IR	SALLIQ	EXPRESSAO	V ou F
100,00	0,00	100	(SALLIQ >= 100,00)	
200,00	10,00	190,00	(SALLIQ < 190,00)	
300,00	15,00	285,00	SALLIQ = SALARIO - IR	

- 2) Sabendo que A=3, B=7 e C=4, informe se as expressões abaixo são verdadeiras ou falsas.

- a) $(A + C) > B$ ()
- b) $B \geq (A + 2)$ ()
- c) $C = (B - A)$ ()
- d) $(B + A) \leq C$ ()
- e) $(C + A) > B$ ()

- 3) Sabendo que A=5, B=4 e C=3 e D=6, informe se as expressões abaixo são verdadeiras ou falsas.

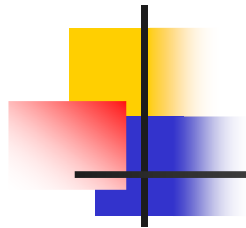
- a) $(A > C)$ **AND** $(C \leq D)$ ()
- b) $(A + B) > 10$ **OR** $(A + B) = (C + D)$ ()
- c) $(A \geq C)$ **AND** $(D \geq C)$ ()



Constantes, Variáveis e Tipos de Dados

Constantes, Variáveis e Tipos de Dados

Variáveis e constantes são os elementos básicos que um programa manipula. Uma variável é um espaço reservado na memória do computador para armazenar um tipo de dado determinado. Variáveis devem receber nomes para poderem ser referenciadas e modificadas quando necessário. Um programa deve conter declarações que especificam de que tipo são as variáveis que ele utilizará e as vezes um valor inicial. Tipos podem ser por exemplo: inteiros, reais, caracteres, etc. As expressões combinam variáveis e constantes para calcular novos valores.

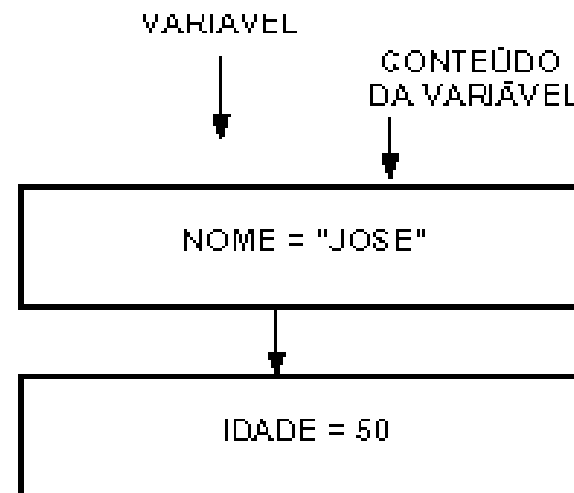


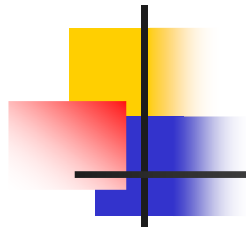
Variáveis

Variáveis

Variável é a representação simbólica dos elementos de um certo conjunto. Cada variável corresponde a uma posição de memória, cujo conteúdo pode se alterado ao longo do tempo durante a execução de um programa. Embora uma variável possa assumir diferentes valores, ela só pode armazenar um valor a cada instante

Exemplos de variáveis





Variáveis – cont.

Tipos primitivos

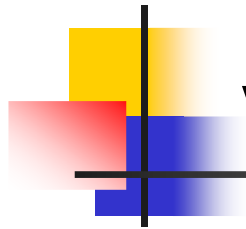
A informação é a material-prima que faz com que seja necessária a existência dos computadores, pois eles são capazes de manipular e armazenar um grande volume de dados com alta performance. Tendo como base as informações manipuladas pelo computador, podemos dividi-las em quatro tipos primitivos, que serão os tipos básicos que usaremos na construção de algoritmos.

Inteiro: toda e qualquer informação numérica que pertença ao conjunto dos números inteiros relativos (negativa, nula ou positiva).

Real: toda e qualquer informação numérica que pertença ao conjunto dos números reais (negativa, nula ou positiva).

Caracter: toda e qualquer informação composta por um conjunto de caracteres alfanuméricos: numéricos(0....9), alfabéticos (AZ, az) e especiais (por exemplo, #, ?, !, @, etc.).

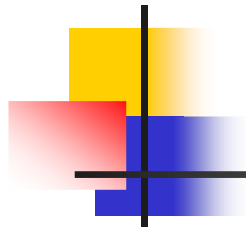
Lógico: toda e qualquer informação que pode assumir apenas duas situações (biestável: verdadeiro ou falso).



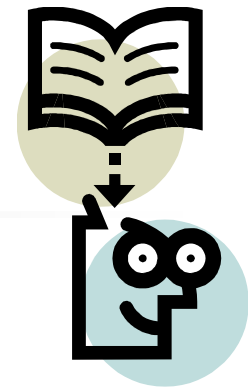
Variáveis – continuação

Regras para declaração de variáveis e constantes:

1. Devem começar por caractere alfabético;
2. Podem ser seguidos por mais caracteres alfabéticos ou numéricos;
3. Não devem ser usados caracteres especiais.

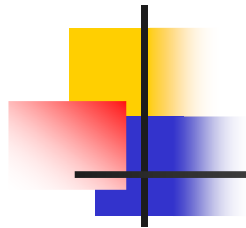


Exercícios

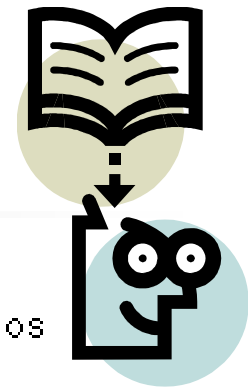


1) O que é uma constante? Dê dois exemplos.

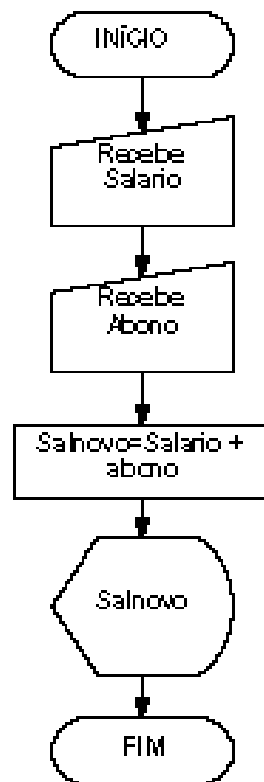
2) O que é uma variável? Dê dois exemplos.



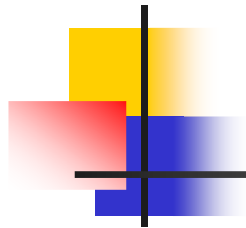
Exercícios – cont.



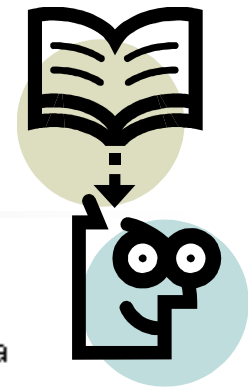
- 3) Faça um teste de mesa no diagrama de bloco abaixo e preencha a tabela ao lado com os dados do teste:



Salário	Abono	Salvo
600,00	60,00	
350,00		

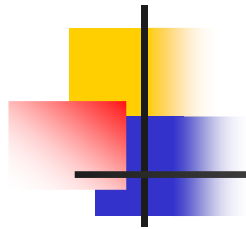


Exercícios – cont.



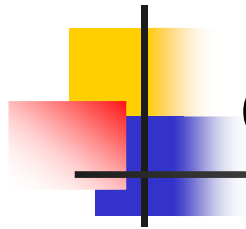
- 4) Sabendo-se que José tem direito a 15% de reajuste de salário, complete o diagrama abaixo:





Estrutura de Decisão

Como vimos no capítulo anterior em “Operações Lógicas”, verificamos que na maioria das vezes necessitamos tomar decisões no andamento do algoritmo. Essas decisões interferem diretamente no andamento do programa. Trabalharemos com dois tipos de estrutura. A estrutura de Decisão e a estrutura de Repetição



Comandos de Decisão

Os comandos de decisão ou desvio fazem parte das técnicas de programação que conduzem a estruturas de programas que não são totalmente seqüenciais. Com as instruções de SALTO ou DESVIO pode-se fazer com que o programa proceda de uma ou outra maneira, de acordo com as decisões lógicas tomadas em função dos dados ou resultados anteriores. As principais estruturas de decisão são: **"Se Então"**, **"Se então Senão"** e **"Caso Selecione"**

SE ENTÃO / IF...THEN

SE ENTÃO / IF ... THEN

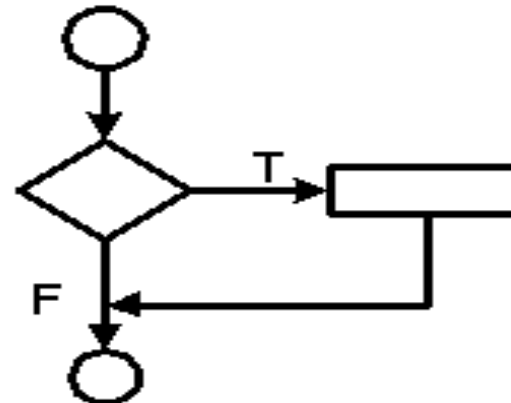
A estrutura de decisão "SE/IF" normalmente vem acompanhada de um comando, ou seja, se determinada condição for satisfeita pelo comando SE/IF então execute determinado comando.

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 ALUNO APROVADO

Em diagrama de blocos ficaria assim:

**Estrutura if
(seleção única)**



Em Visual Basic

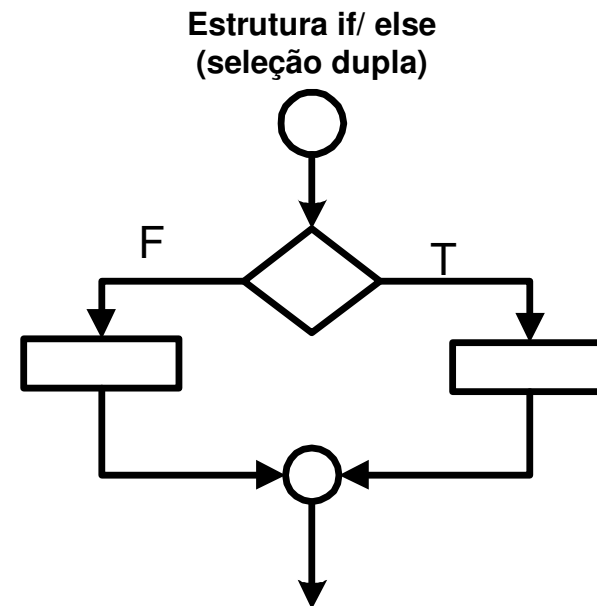
```
IF MEDIA >= 5 Then  
    Text1 = "APROVADO"  
ENDIF
```

SE ENTÃO SENÃO / IF ... THEN ... ELSE

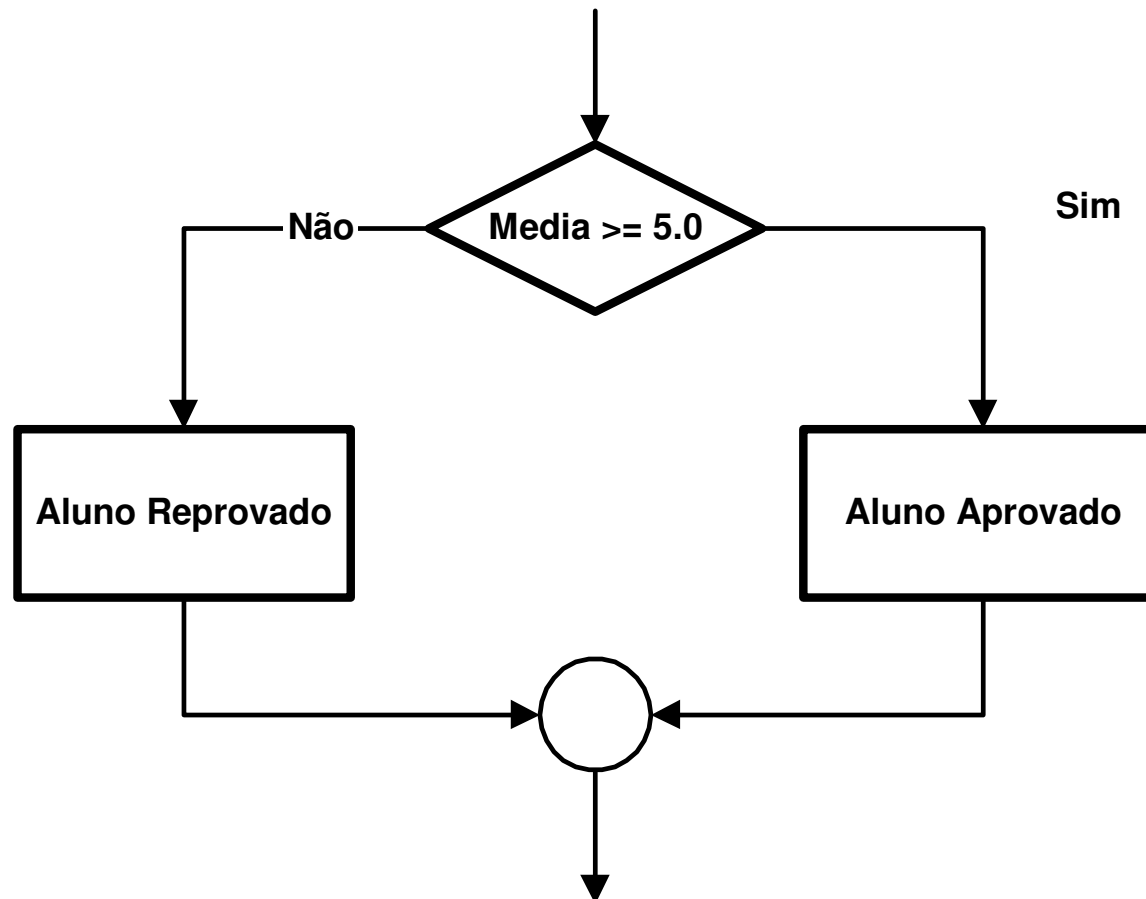
A estrutura de decisão "SE/ENTÃO/SENÃO", funciona exatamente como a estrutura "SE", com apenas uma diferença, em "SE" somente podemos executar comandos caso a condição seja verdadeira, diferente de "SE/SENÃO" pois sempre um comando será executado independente da condição, ou seja, caso a condição seja "verdadeira" o comando da condição será executado, caso contrário o comando da condição "falsa" será executado

Em algoritmo ficaria assim:

```
SE MÉDIA >= 5.0 ENTÃO  
    ALUNO APROVADO;  
SENÃO  
    ALUNO REPROVADO;  
FIMSE
```

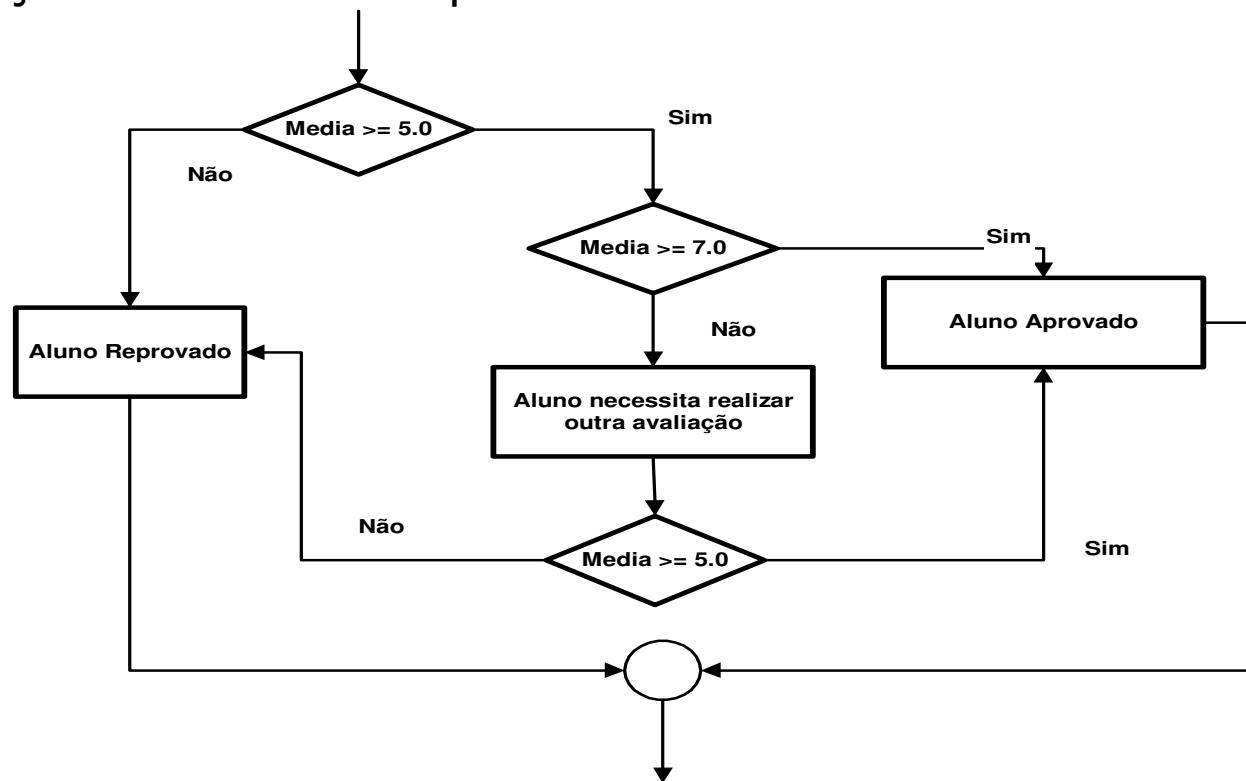


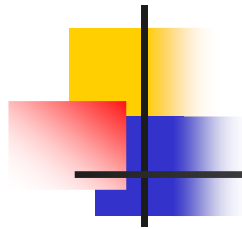
SE ENTÃO SENÃO / IF ... THEN ... ELSE em Diagrama de Blocos



SE ENTÃO SENÃO / IF ... THEN ... ELSE - cont.

No exemplo anterior está sendo executada uma condição que, se for verdadeira, executa o comando "APROVADO", caso contrário executa o segundo comando "REPROVADO". Podemos também dentro de uma mesma condição testar outras condições. Como no exemplo abaixo:

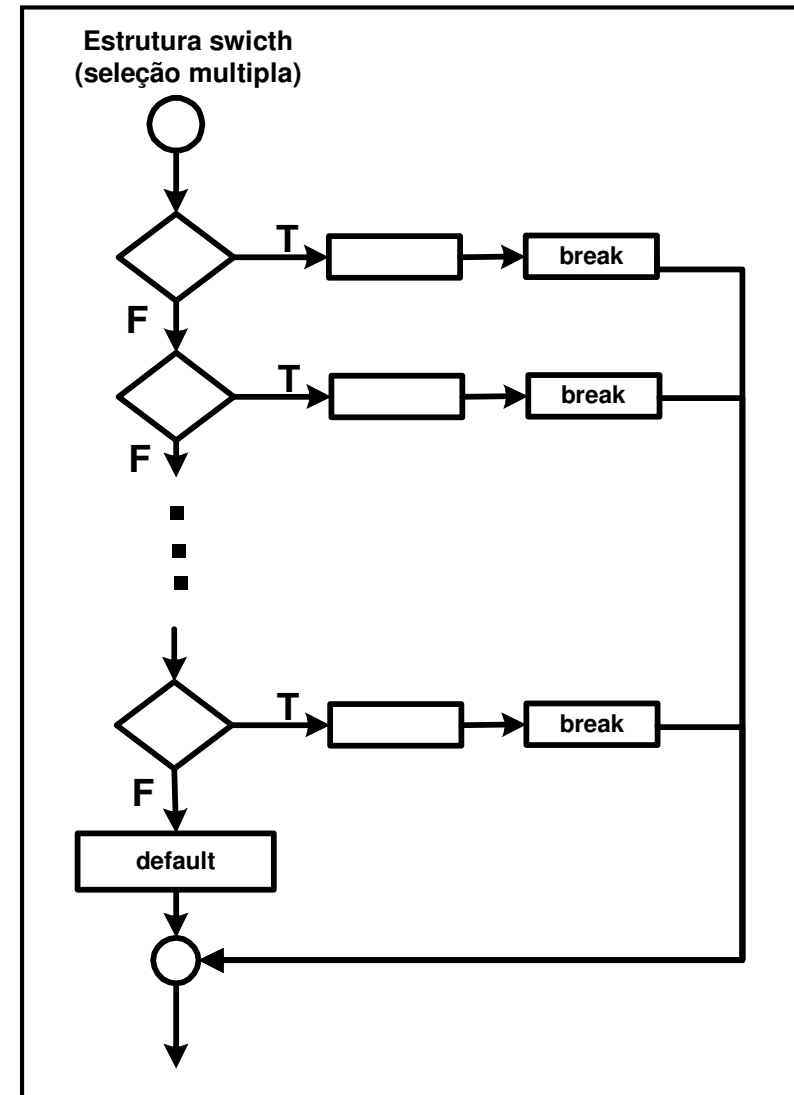




Caso Selezione / Select ... Case

A estrutura de decisão CASO/SELEZIONE é utilizada para testar, na condição, uma única expressão, que produz um resultado, ou, então, o valor de uma variável, em que está armazenado um determinado conteúdo. Compara-se, então, o resultado obtido no teste com os valores fornecidos em cada cláusula “Caso”.

No exemplo do diagrama de blocos abaixo, é recebido uma variável “Op” e testado seu conteúdo, caso uma das condições seja satisfeita, é atribuído para a variável Titulo a String “Opção X”, caso contrário é atribuído a string “Opção Errada”.



Caso Selecione / Select ... Case

Exemplo:

