

BC-0005 Bases Computacionais da Ciência

Aula 06 – Lógica de Programação: Estruturas condicionais

Programar...

 Aprender a programar não é uma tarefa trivial (pois envolve manipular uma representação abstrata)

- Nessa aula veremos um conceito mais elaborado:
 - → As estruturas condicionais.

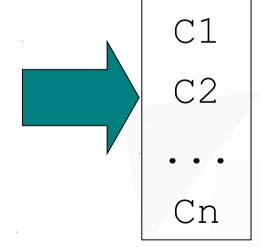
Estruturas de controle

- Estruturas de controle permitem o controle do fluxo de execução dos comandos.
- Temos três estruturas básicas de controle:
 - Sequencial
 - Condicional ou Desvio (if)
 - Repetição

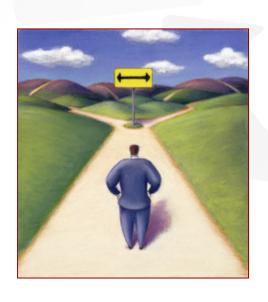
Estrutura sequencial

- É padrão em toda a forma de algoritmo.
- Uma estrutura sequencial é um conjunto de comandos que serão executados em uma sequência linear, de cima para baixo.

Os comandos serão executados na mesma ordem em que foram escritos



- É também conhecida como de decisão ou seleção
- Um desvio condicional é usado para escolher entre cursos alternativos de ação em um progr



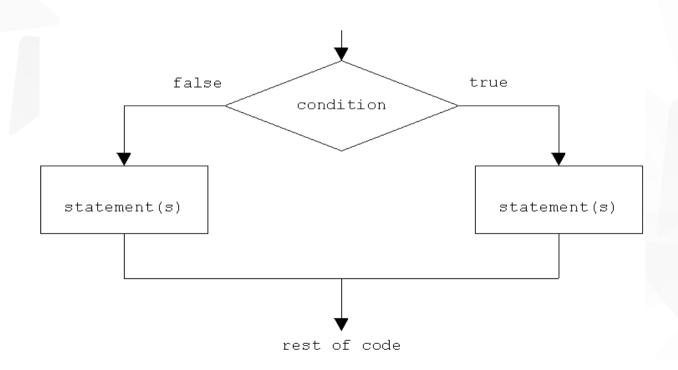


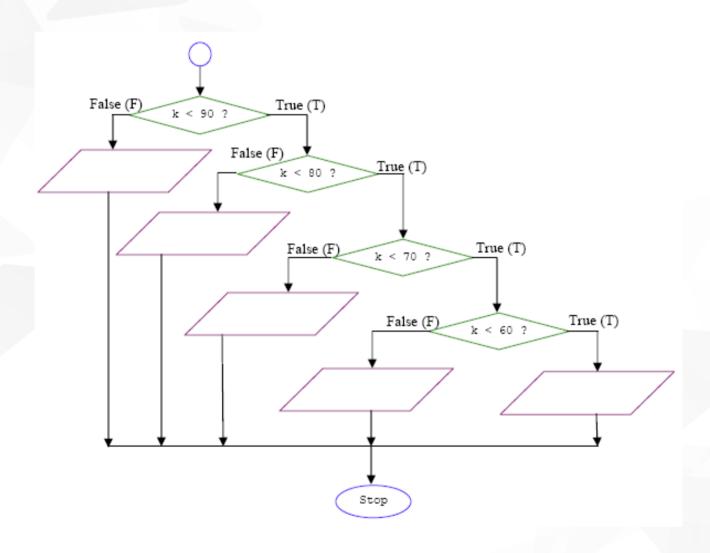
Se subdivide em:

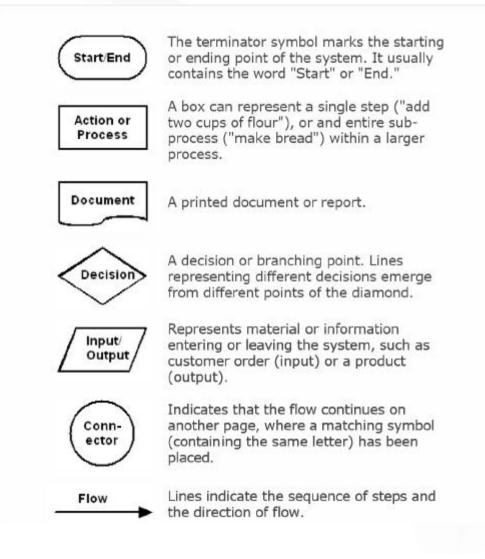
- Estrutura/desvio condicional simples (se-então)
- Estrutura/desvio condicional composta(o) (se-então-senão)

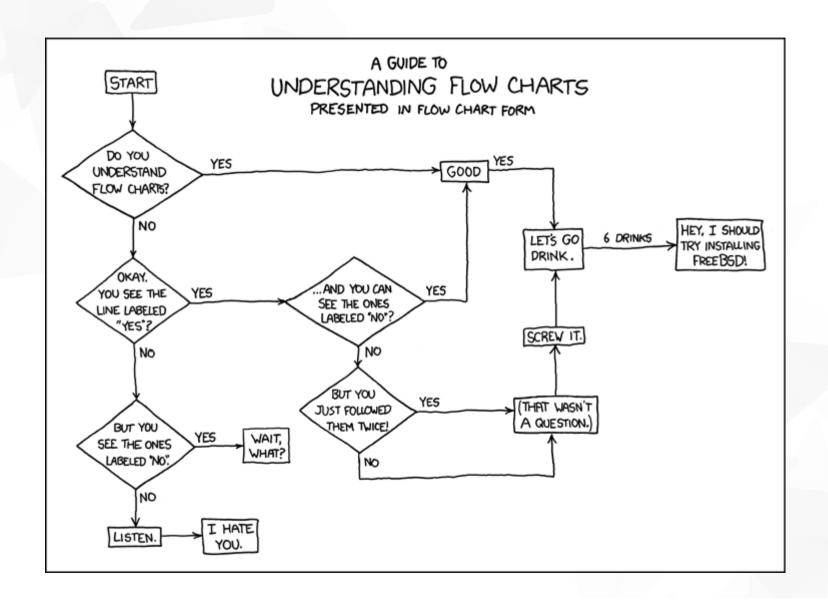
Se subdivide em:

- Estrutura/desvio condicional simples (se-então)
- Estrutura/desvio condicional composta(o) (se-então-senão)









Operadores de relação

```
-->a = 10;
-->b = 20;
-->a == b
ans =
 F
-->a*2 == b
ans =
```

Operadores de relação

```
-->a = 10;
-->b = 20;
```

comparison operator	meaning
==	is equal to
~=	is not equal to
<	is less than
<=	is less than or equal to
>	is greater than
>=	is greater than or equal to

Operadores lógicos

```
-->a = 10;

-->b = 20;

-->a<=b & a==10

ans =

T

-->a<0 | b<0

ans =
```

a	Ъ	a & b	a l b
true	true	true	true
true	false	false	true
falæ	true	false	true
falæ	false	false	falæ

"Truth table"

Operadores lógicos

```
-->a = 10;

-->b = 20;

-->a<=b & a==10

ans =

T

-->a<0 | b<0

ans =
```

>~(a<0 b<0)
ans =
Т

a	ъ	a & b	a b
true	true	true	true
true	false	false	true
falæ	true	false	true
falæ	false	false	falæ

"Truth table"

Operadores lógicos

```
-->%t & %f
ans =
F
-->%t | %f
ans =
T
```

a	b	a & b	a _ b
true	true	true	true
true	false	false	true
falæ	true	false	true
falæ	false	false	falæ

"Truth table"

(1) Valor absoluto de um número

```
-->valorAbsoluto(10)
ans =
   10.
-->valorAbsoluto(-10)
ans =
   10.
```

(1) Valor absoluto de um número

```
function r = valorAbsoluto(numero)

function r = valorAbsolut
```

Módulo

Teste as seguintes instruções. O que realiza a função?

- -->modulo(0, 2)
- -->modulo(1, 2)
- -->modulo(11, 3)
- -->modulo(17, 3)
- -->modulo(15, 5)
- -->modulo(15, 8)
- -->modulo(15, 18)

(2) Número par

(2) Número par

```
function r = numeroPar(numero)
function r = numeroPar(num
```

(3) Maior elemento

Crie uma função para, dado como parâmetros dois número, determinar o maior número.

Exemplos:

```
-->maiorElemento(13, 15)
ans =
15.

-->maiorElemento(15, 13)
ans =
15.
```

(3) Maior elemento

```
function r = = maiorElemento(a,b)

function r = = maiorElemento(a,b)

function r = = emaiorElemento(a,b)

function r = = emaiorElemento(a,b)

function r = = emaiorElemento(a,b)

function r = emaiorElemento
```

(4) Cara ou coroa

```
function sorte == caraOuCoroa()
function sorte == caraOuCoroa()
function sorte == rand();
f
```

```
-->caraOuCoroa()
ans =
coroa

-->caraOuCoroa()
ans =
cara
```

(5) Quantidade de dias no mês

Crie uma função para determinar a quantidade de dias para um determinado mês do ano.

```
-->quantidadeDeDias(1)
ans =
31.
-->quantidadeDeDias(2)
ans =
28.
```

Nota: Assuma ano não bissexto.

```
tem 31 dias
Janeiro
Fevereiro tem 28 (ou 29 dias nos anos bissextos)
         tem 31 dias
Março
Abril
         tem 30 dias
Maio
         tem 31 dias
Junho
         tem 30 dias
Julho
         tem 31 dias
Agosto
         tem 31 dias
Setembro tem 30 dias
Outubro tem 31 dias
Novembro tem 30 dias
Dezembro tem 31 dias
```

(5) Quantidade de dias no mês

Ano bissexto:

- De 4 em 4 anos é ano bissexto.
- De 100 em 100 anos não é ano bissexto.
- De 400 em 400 anos é ano bissexto.
- Prevalecem as últimas regras sobre as primeiras.

São bissextos todos os anos múltiplos de 400.

p.ex: 1600, 2000, 2400, 2800...

São bissextos todos os múltiplos de 4 e não múltiplos de 100.

p.ex: 1996, 2004, 2008, 2012, 2016...

Não são bissextos todos os demais anos.

Se (ano módulo 4 é 0 e (ano módulo 400 é 0 ou ano módulo 100 diferente 0))

Então

Verdadeiro // Ano Bissexto

Senão

Falso // Ano não é Bissexto

```
function anoBissexto(ano)

function anoBissexto(anoBissexto(ano)

function anoBissexto(anoBissexto(ano)

function anoBissexto(anoBissexto(anoBissexto(anoBissexto(ano)

function anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoBissexto(anoB
```

```
-->anoBissexto(2012)
Ano Bisexto
```

-->anoBissexto(2014) Ano não é Bisexto

-->anoBissexto(2016)
Ano Bisexto

```
function-bissexto-=-anoBissexto2-(ano)

if (modulo(ano, 4)==0.&-(modulo(ano, 400)==0-|-modulo(ano, 100)~=0)-)-then

bissexto-=-%T;

else

----bissexto-=-%F;

endfunction
```

```
--> if (anoBissexto2(2000)) then disp("Ano Bissexto") else disp("Ano não é bissexto") end Ano Bisexto
```

```
--> if (anoBissexto2(2000)) then disp("Ano Bissexto") else disp("Ano não é bissexto") end Ano não é Bisexto
```

```
--> quantidadeDeDias2(2,2016) --> quantidadeDeDias(2,2017) 28
```

(7) Índice de massa corporal

Crie uma função para que, dado como parâmetros o peso (em kgr), a altura (em metros), se determine o índice de massa corporal e sua classificação.

IMC = Peso/Altura²

IMC	CLASSIFICAÇÃO
< 18,5	Peso Baixo
18,5 - 24,9	Peso Normal
25,0 - 29,9	Sobrepeso
30,0 - 34,9	Obesidade (Grau I)
35,0 - 39,9	Obesidade Severa (Grau II)
≥ 40,0	Obesidade Mórbida (Grau III)

-->IMC(55, 1.65) ans = 20.20202

(7) Índice de massa corporal

```
1 |function [imc, classificacao] = IMC(peso, altura)
2 | · · · imc = peso/altura^2;
3 \cdot \cdot \cdot \cdot \text{if imc} < 18.5 \cdot \text{then}
4 | ---- classificacao = "peso baixo";
5 | e elseif imc<=24.9 then
6 | e e e e classificacao = "peso normal";
7 | · · · elseif imc<=29.9 then
9 | elseif imc<=34.9 then
10 ----- classificacao = "obesidade (grau I)";
11 ---- elseif imc<=39.9 then
12 ---- classificacao = "obesidade severa (grau II)";
13 - else
14 ----- classificacao = "obesidade severa mórbida (grau III)";
15 --- end
16 endfunction
```

```
--> [imc, classificacao] = IMC(55, 1.65)
classificacao = peso normal
imc = 20.20202
```

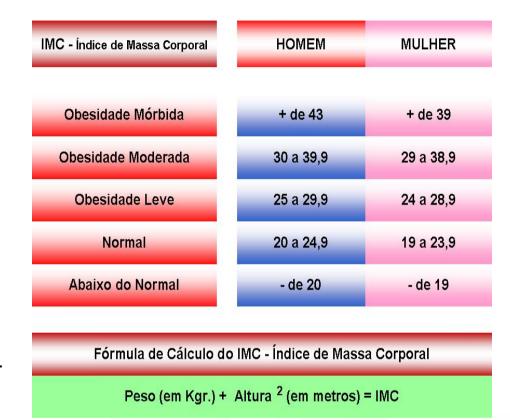
Semana 06: Atividade 1)

- Faça um script que leia as notas de duas provas, p1 e p2, e converta a nota para conceito:
 - Se a média for menor que 5, o conceito é F
 - Se a média estiver entre 5 e 6,5, o conceito é D
 - Se a média estiver entre 6,5 e 7,5, o conceito é C
 - Se a média estiver entre 7,5 e 9, o conceito é B
 - Se a média esiver entre 7,5 e 10(inclusive) o conceito é A

Semana 06: Atividade 2)

Considere a seguinte tabela e equação para determinar o IMC de uma pessoa.

Crie uma função para que, dada como entrada o peso (em kgr), a altura (em metros), e o gênero ('homem' ou 'mulher') seja determinado tanto o IMC quanto a sua classificação.



Exemplo de uso: [imc, classificação] = imc2(55, 1.65, 'mulher')

Semana 06: Atividade 3.a)

Crie uma função para que, dada como entrada 3 coeficientes de uma equação do segundo grau ($ax^2+bx+c=0$), e calcule as raízes (x_1 e x_2) da equação. As raízes da equação dependem do valor de delta (b^2 – 4ac) Se delta >0

$$X_1 = \frac{-b + \sqrt{delta}}{2a}, X_1 = \frac{-b - \sqrt{delta}}{2a}$$

Se delta = 0

$$X_1 = \frac{-b}{2a}, X_2 = X_1$$

Se delta <0

$$X_{1} = \frac{-b}{2a} - \frac{\sqrt{-delta}}{2a}i, X_{2} = \frac{-b}{2a} + \frac{\sqrt{-delta}}{2a}i$$

Em que i pode ser representado como %i no scilab

function [x1,x2] = raizes(a,b,c)

endfuncion

Semana 06: Atividade 3.b)

- Faça um script que leia os valores de a, b, e c a partir da entradas do usuário, chame a função desenolvida no slide anterior, e plot o gráfico da função no intervalo -10,10
- Se as raizes forem reais, adicione ao gráfico o(s) ponto(s) referente(s) a essa(s) raíze(s)
- Para chamar a função, você pode fazer[x1,x2] = raizes(a,b,c)
- Para testar se o número é real ou não, você pode usar a função isreal(x), que testa se o x é um valor real, devolvendo verdadeiro ou falso