

Operadores e Estruturas de Decisão

Leonardo Gresta Paulino Murta
leomurta@ic.uff.br

Aula de hoje...

- Operadores
 - Aritméticos (usados em contas)
 - Relacionais (usados em comparações numéricas)
 - Lógicos (usados em comparações lógicas)
 - De atribuição (armazenamento de valores em variáveis)
- Estruturas de decisão
 - *If...then*
 - *If...then...else*
 - *Case*

Operadores aritméticos

Operador	Exemplo	Prioridade
(expr)	$(1 + 2) * 3 \rightarrow 9$	1
*	$5 * 3 \rightarrow 15$	2
/	$5 / 3 \rightarrow 1.666666666$	2
div	$5 \text{ div } 3 \rightarrow 1$	2
mod	$5 \text{ mod } 3 \rightarrow 2$	2
+	$5 + 3 \rightarrow 8$	3
-	$5 - 3 \rightarrow 2$	3

Operadores aritméticos

- Operadores com a mesma prioridade (precedência)
 - Analisados da esquerda para a direita
- Divisão de inteiros (div)
 - Numerador e denominador inteiros
 - Resultado é somente a parte inteira da divisão
- Divisão em modo misto (/)
 - Resultado fracionário

Exemplo

- Considerando

```
var
  x : integer;
  y : real;
begin
  x := 512;
  y := 9.2 - (x div 10 - 14 / 5) + 14 * 0.1;
  ...
```

Resolução de y

```
y := 9.2 - (512 div 10 - 14 / 5) + 14 * 0.1
y := 9.2 - (51 - 14 / 5) + 14 * 0.1
y := 9.2 - (51 - 2.8) + 14 * 0.1
y := 9.2 - 48.2 + 14 * 0.1
y := 9.2 - 48.2 + 1.4
y := -39.0 + 1.4
y := -37.6
```

Funções matemáticas

Método	Descrição	Exemplo
<code>abs(expr)</code>	Valor absoluto	<code>abs(-5.3) → 5.3</code>
<code>round(expr)</code>	Arredonda um número	<code>round(5.5) → 6</code>
<code>trunc(expr)</code>	Arredonda para baixo	<code>trunc(5.5) → 5</code>
<code>int(expr)</code>	Obtém a parte inteira	<code>int(5.55) → 5.0</code>
<code>frac(expr)</code>	Obtém a parte fracionária	<code>frac(5.55) → 0.55</code>
<code>sqrt(expr)</code>	Raiz quadrada	<code>sqrt(4) → 2.0</code>
<code>sqr(expr)</code>	Quadrado	<code>sqr(4) → 16</code>
<code>power(expr1, expr2)</code>	Potência	<code>power(2, 3) → 8</code>
<code>ln(expr)</code>	Logaritmo natural (base E)	<code>ln(exp(3)) → 3.0</code>
<code>exp(expr)</code>	Exponencial (e^{expr})	<code>exp(0) → 1.0</code>

Funções matemáticas

Função	Descrição	Exemplo
<code>sin(expr)</code>	Seno	<code>sin(0) → 0.0</code>
<code>cos(expr)</code>	Cosseno	<code>cos(0) → 1.0</code>
<code>arctan(expr)</code>	Arco tangente	<code>arctan(1) → 0.7853981633974483</code>
<code>pi</code>	Valor de π	<code>pi → 3.1415926535897932</code>

- Funções trigonométricas trabalham com radiano
- Existem algumas outras funções menos usadas

Exemplo

- Considerando

```
var x, y : real;
begin
  x := trunc(3.5 + 7 / 2 - 5);
  y := trunc(3.5) + 7 / 2 - 5;
  ...
```

- Resolução de x

```
x := trunc(3.5 + 3.5 - 5)
x := trunc(7.0 - 5)
x := trunc(2.0)
x := 2
```

- Resolução de y

```
y := trunc(3.5) + 3.5 - 5
y := 3 + 3.5 - 5
y := 6.5 - 5
y := 1.5
```


Números aleatórios

- Algumas aplicações necessitam que o computador sorteie um número
 - Função **random()** gera número pseudo aleatório entre $[0, 1)$
 - Antes de usar **random()**, é necessário usar **randomize()** para iniciar uma nova sequência de números aleatórios
- A partir desse número, é possível gerar números em outros intervalos

$$\text{inicio} + (\text{fim} - \text{inicio}) * \text{random}()$$

Exemplo

- Primeiro gere a sequência de números aleatórios:

```
randomize();
```

- Número entre 0 e 1

```
writeln(random());
```

- Número entre 5 e 6

```
writeln(5 + random());
```

- Número entre 0 e 10

```
writeln(random() * 10);
```

- Número entre 50 e 70

```
writeln(50 + random() * 20);
```

Operadores relacionais

Operador	Exemplo	Prioridade
$\text{expr1} < \text{expr2}$	$5 < 3 \rightarrow \text{false}$	4
$\text{expr1} \leq \text{expr2}$	$5 \leq 3 \rightarrow \text{false}$	4
$\text{expr1} > \text{expr2}$	$5 > 3 \rightarrow \text{true}$	4
$\text{expr1} \geq \text{expr2}$	$5 \geq 3 \rightarrow \text{true}$	4
$\text{expr1} = \text{expr2}$	$5 = 3 \rightarrow \text{false}$	4
$\text{expr1} \neq \text{expr2}$	$5 \neq 3 \rightarrow \text{true}$	4
expr in conj	$5 \text{ in } [1,2,7..9] \rightarrow \text{false}$	4

- Prioridade sempre inferior aos operadores aritméticos
- Sempre têm **números como operandos**
- Sempre têm **resultado booleano**

Operadores lógicos

Operador	Exemplo	Prioridade
not expr	not true \rightarrow false	1
expr1 and expr2	true and false \rightarrow false	2
expr1 xor expr2	true xor true \rightarrow false	3
expr1 or expr2	true or true \rightarrow true	3

- Prioridade sempre superior aos operadores relacionais, então **sempre use parênteses ao redor dos operadores relacionais**
- Quando têm **booleanos como operandos** geram **resultado booleano**
- Quando têm números como operandos geram resultados numéricos (*bitwise* – não veremos isso no curso)

Tabela verdade

a	b	not a	a and b	a xor b	a or b
true	true	false	true	false	true
true	false	false	false	true	true
false	true	true	false	true	true
false	false	true	false	false	false

Operador de atribuição

Operador	Exemplo
var := expr	x := 10 + 5

- Primeiro resolve a expressão do lado direito
- Depois atribui o valor gerado à variável do lado esquerdo
- **Atenção:** Não confunda o **operador de atribuição** “:=” com o **operador relacional** “=”

Exemplo

- Considerando

```

var
  x, y, z : real;
  w : boolean;
Begin
  x := 10;
  y := -2;
  z := 5;
  w := (x * y < z / x) or (x / y > z * x) and (z * y > x);

```

Resolução de w

```

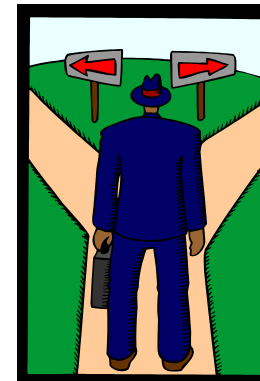
(10 * -2 < 5 / 10) or (10 / -2 > 5 * 10) and (5 * -2 > 10)
(-20 < 0.5) or (-5 > 50) and (-10 > 10)
true or false and false
true or false
true

```

Decisão

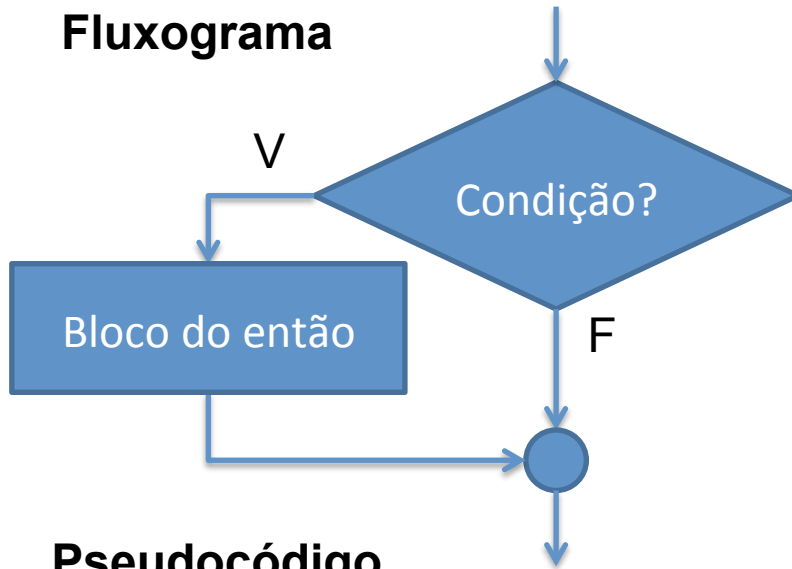
Mecanismos de decisão:

- *If...then*
 - Executa algo somente quando uma condição é verdadeira
- *If...then...else*
 - Bifurca a execução do código em função de uma condição
- *Case*
 - Executa múltiplos trechos de código em função do valor de uma expressão



Decisão do tipo *if...then*

Fluxograma



Pseudocódigo

```

...
Se CONDIÇÃO então
  INSTRUÇÃO 1
  INSTRUÇÃO 2
  ...
  INSTRUÇÃO N
...
  
```

Pascal

```

...
if CONDIÇÃO then
  INSTRUÇÃO;
...
  
```

Ou

```

...
if CONDIÇÃO then
begin
  INSTRUÇÃO 1;
  INSTRUÇÃO 2;
  ...
  INSTRUÇÃO N;
end;
...
  
```

Decisão do tipo *if...then*

- Executa o bloco de instruções somente se a condição for verdadeira
- A condição é uma expressão booleana que pode fazer uso de quaisquer operadores
- Pode omitir *begin* e *end* caso execute somente uma instrução

Exemplo de *if...then*

- Programa para informar o valor absoluto de um número:

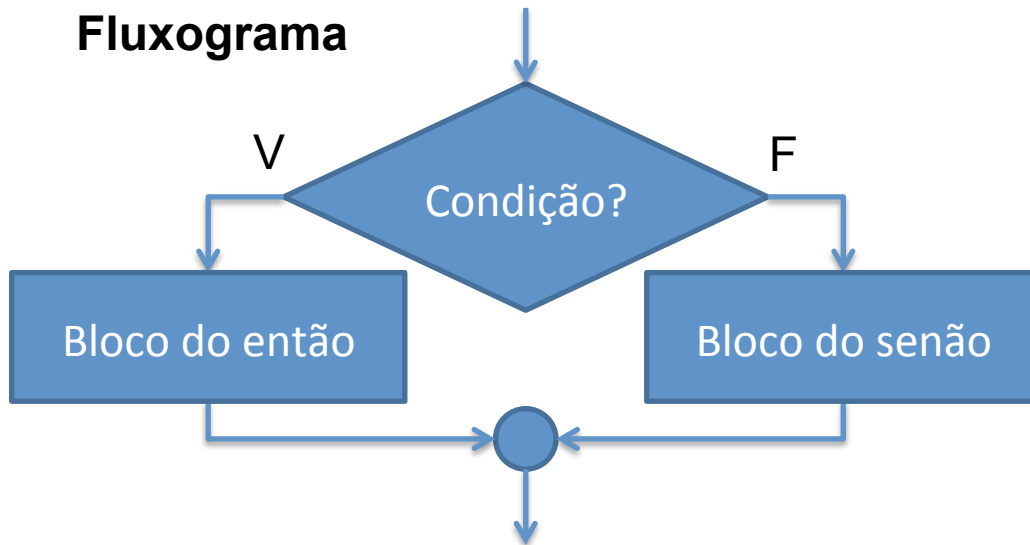
```

program absoluto;
var numero : real;
begin
  write('Entre com um número: ');
  readln(numero);
  if numero < 0 then
    numero := -numero;
  writeln(numero);
end.

```

Decisão do tipo *if..then..else*

Fluxograma



Pseudocódigo

```

...
Se CONDIÇÃO então
  INSTRUÇÃO 1
  INSTRUÇÃO 2
  ...
  INSTRUÇÃO N
Senão
  INSTRUÇÃO 1
  INSTRUÇÃO 2
  ...
  INSTRUÇÃO N
...
  
```

Decisão do tipo *if...then...else*

Pascal

```

...
if CONDIÇÃO then
    INSTRUÇÃO
else
    INSTRUÇÃO;
...

```

Ou

```

...
if CONDIÇÃO then
begin
    INSTRUÇÃO 1;
    INSTRUÇÃO 2;
    ...
    INSTRUÇÃO N;
end
else
begin
    INSTRUÇÃO 1;
    INSTRUÇÃO 2;
    ...
    INSTRUÇÃO N;
end;
...

```

Decisão do tipo *if...then...else*

- Executa um ou o outro bloco de instruções em função da condição ser verdadeira ou falsa
- Valem as mesmas regras para *if...then*
- Qualquer combinação de instrução individual ou em bloco é aceita no *then* e no *else*
- Não pode ter ; no final da última instrução antes do *else*
- Podem ser aninhados com outras estruturas do tipo *if...then...else*

Exemplo de *if...then...else*

- Programa para informar se um número é par ou ímpar:

```

program paridade;
var numero : integer;
begin
  write('Entre com um número: ');
  readln(numero);
  if numero mod 2 = 0 then
    writeln('O número é par.')
  else
    writeln('O número é ímpar.');
```

end.

Exemplo de *if* aninhado

- Programa para informar o número de dias de um mês:

```

program dia_mes;
var mes, ano : integer;
begin
  write('Entre com um mês (1 a 12): ');
  readln(mes);
  if (mes=1) or (mes=3) or (mes=5) or (mes=7) or (mes=8) or (mes=10) or (mes=12) then
    writeln('Esse mês tem 31 dias.')
  else if (mes=4) or (mes=6) or (mes=9) or (mes=11) then
    writeln('Esse mês tem 30 dias.')
  else
    begin
      write('Entre com o ano (4 dígitos): ');
      readln(ano);
      if (ano mod 400 = 0) or (ano mod 4 = 0) and (ano mod 100 <> 0) then
        writeln('Esse mês tem 29 dias.')
      else
        writeln('Esse mês tem 28 dias.');
    end;
end.

```


Exemplo de *if* aninhado

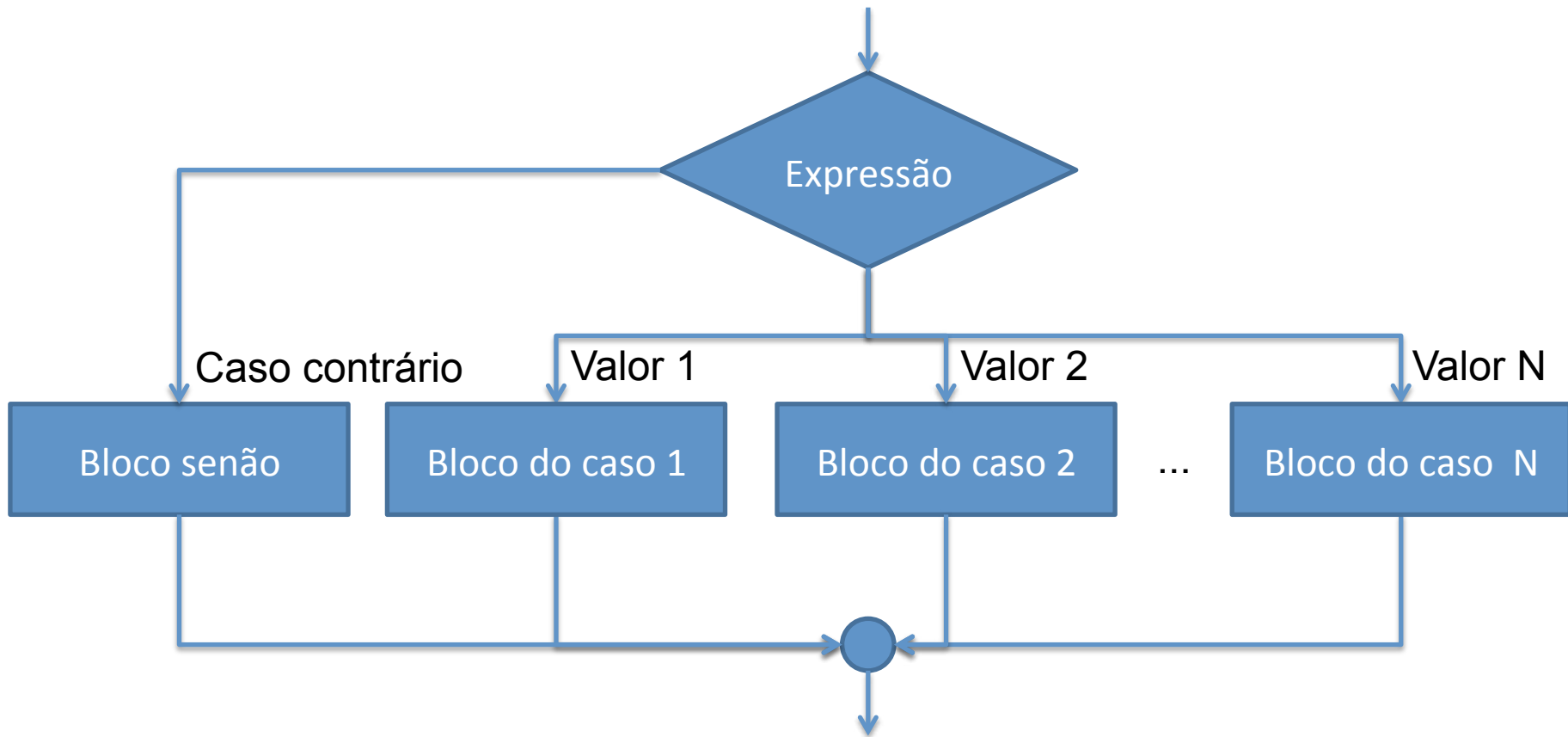
- Usando conjunto:

```

program dia_mes;
var mes, ano : integer;
begin
  write('Entre com um mês (1 a 12): ');
  readln(mes);
  if mes in [1, 3, 5, 7, 8, 10, 12] then
    writeln('Esse mês tem 31 dias.')
  else if mes in [4, 6, 9, 11] then
    writeln('Esse mês tem 30 dias.')
  else
    begin
      write('Entre com o ano (4 dígitos): ');
      readln(ano);
      if (ano mod 400 = 0) or (ano mod 4 = 0) and (ano mod 100 <> 0) then
        writeln('Esse mês tem 29 dias.')
      else
        writeln('Esse mês tem 28 dias.');
    end;
end.

```

Decisão do tipo *switch...case*



Decisão do tipo *switch...case*

Pascal

```

...
case EXPRESSÃO of
  VALORES 1:
    INSTRUÇÃO OU BLOCO 1;
  VALORES 2:
    INSTRUÇÃO OU BLOCO 2;
  ...
  VALORES N:
    INSTRUÇÃO OU BLOCO N;
[else:
  INSTRUÇÃO OU BLOCO; ]
end;
...

```

Decisão do tipo *case*

- Aceita expressões dos tipos integer e char
- É equivalente a *if* aninhado
 - Escolher o que tem melhor legibilidade
- O uso de *else* é opcional

Exemplo de *case*

```

program dia_mes;
var mes, ano : integer;
begin
  write('Entre com um mês (1 a 12): ');
  readln(mes);
  case mes of
    1, 3, 5, 7, 8, 10, 12:
      writeln('Esse mês tem 31 dias.');
```





Exemplo de *case*

```

2:
begin
  write('Entre com o ano (4 dígitos): ');
  readln(ano);
  if (ano mod 400=0) or (ano mod 4=0) and (ano mod 100<>0) then
    writeln('Esse mês tem 29 dias.')
  else
    writeln('Esse mês tem 28 dias.');
```

end;

```

else
  writeln('Valor informado não é um mês.');
```

end;

```

end.
```

Exercícios

- Faça um programa que calcule o IMC de uma pessoa ($\text{IMC} = \text{massa em kg} / \text{altura em metros}^2$) e informe a sua classificação segundo a tabela a seguir, obtida na Wikipédia

IMC	Classificação
< 18,5	Abaixo do Peso
18,6 – 24,9	Saudável
25,0 – 29,9	Peso em excesso
30,0 – 34,9	Obesidade Grau I
35,0 – 39,9	Obesidade Grau II (severa)
≥ 40,0	Obesidade Grau III (mórbida)

Exercícios

- Faça um programa que leia três coordenadas num espaço 2D e indique se formam um triângulo, juntamente com o seu tipo (equilátero, isósceles e escaleno)
 - Equilátero: todos os lados iguais
 - Isósceles: dois lados iguais
 - Escaleno: todos os lados diferentes

Exercícios

- Faça um programa que leia um número inteiro de 5 dígitos e indique se ele é palíndromo
 - Um número palíndromo é aquele que se lido da esquerda para a direita ou da direita para a esquerda possui o mesmo valor (ex.: 15451)

Exercícios

- Faça um programa que leia um número inteiro entre 0 e 9999 e escreva o seu valor por extenso

Operadores e Estruturas de Decisão

Leonardo Gresta Paulino Murta
leomurta@ic.uff.br