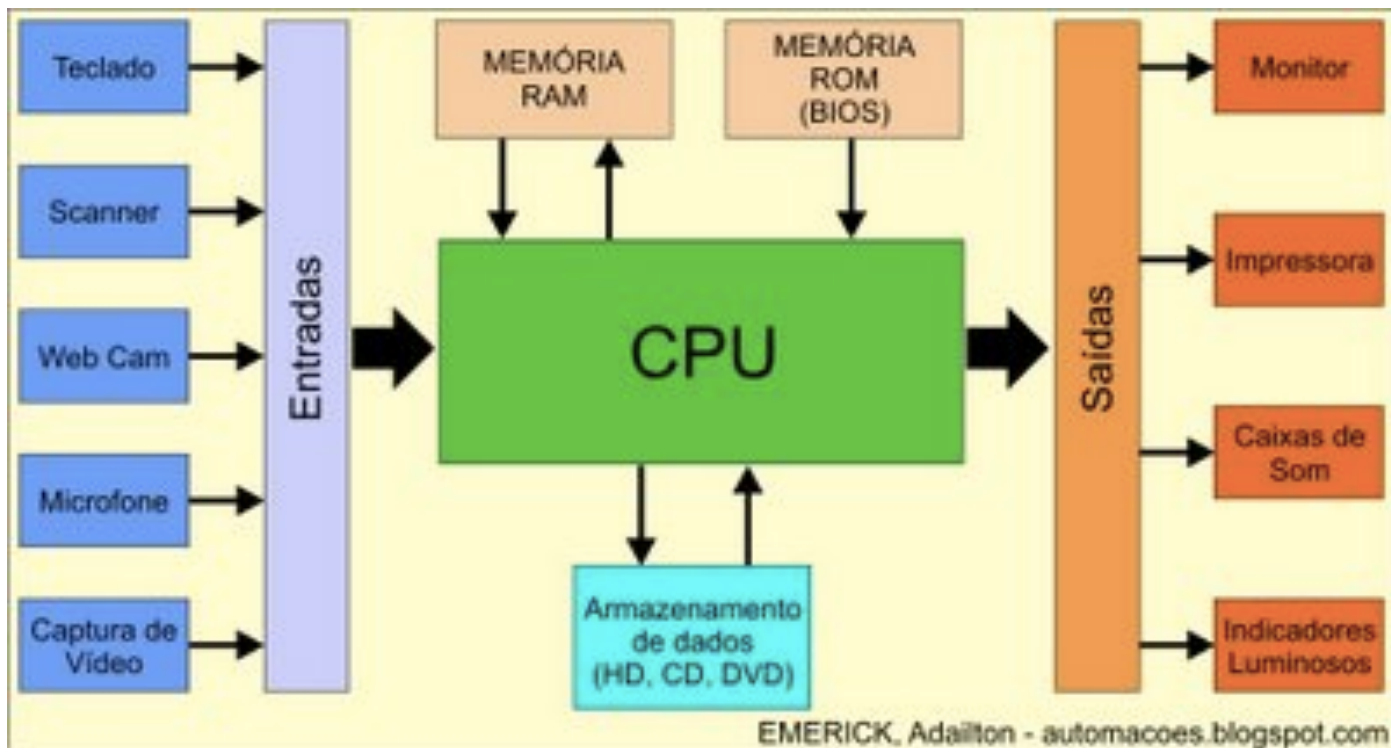


Lógica de Programação e Algoritmos

Profa. Eliane Oliveira Santiago

Arquitetura dos Computadores



Unidades de Medida de Dados

Unidade	Símbolo	Valor Equivalente
Bit	0 ou 1	
Byte	B	8 bits
Kilobyte	KB	1024 B
Megabyte	MB	1024 KB
Gigabyte	GB	1024 MB
Terabyte	TB	1024 GB
Petabyte	PB	1024 TB
Exabyte	EB	1024 PB
Zettabyte	ZB	1024 EB
Yottabyte	YB	1024 ZB



Tipos de Dados

Os tipos de dados são as características comuns dos dados a serem manipulados.

Podemos considerar quatro classificações para os tipos de dados:

- Inteiro
- Real
- Caracter
- Lógico

Operador de atribuição (\leftarrow) e comandos de atribuição

Atribui um determinado valor para uma variável, tendo o cuidado de verificar se o valor que está sendo atribuído à variável tem o tipo de dado compatível, ou seja, se uma variável **x** foi declarada como inteiro, só é permitido atribuir valores inteiros à variável **x**.

Sintaxe:

<nome da variável> \leftarrow <expressão> ;

onde **expressão** pode ser uma expressão lógica ou aritmética

Exemplo de uso do operador de atribuição (\leftarrow)

Por exemplo, vamos considerar que as variáveis **x**, **y** e **z** foram declaradas como do tipo **numérico_inteiro**:

```
x  $\leftarrow$  25;
```

```
y  $\leftarrow$  x + 15 - 3;
```

```
z  $\leftarrow$  y - x + rad(x) - pot(y , 2);
```

Equação de 2º. grau

Equação: $2x^2 + 3x + 5$

Variáveis do problema: $ax^2 + bx + c$

$$\Delta = b^2 - 4.a.c$$

$$x = \frac{-b \pm \sqrt{\Delta}}{2.a}$$

Expressão aritmética para calcular o delta

Algoritmo Equação 2º. grau

Equação: $3/2x^2 + 3x + 5$

início_algoritmo

Variáveis do problema: $ax^2 + bx + c$

var a, b, c, delta, x1, x2 : **real**

leia (a, b, c);

$\text{delta} \leftarrow b*b - 4 * a * c;$

escreva (delta);

fim_algoritmo.

$$\Delta = b^2 - 4.a.c$$

Algoritmo para calcular o delta

início_algoritmo

var

a, b, c , delta: **inteiro**

x1, x2 : **real**

//entrada

leia (a, b, c);

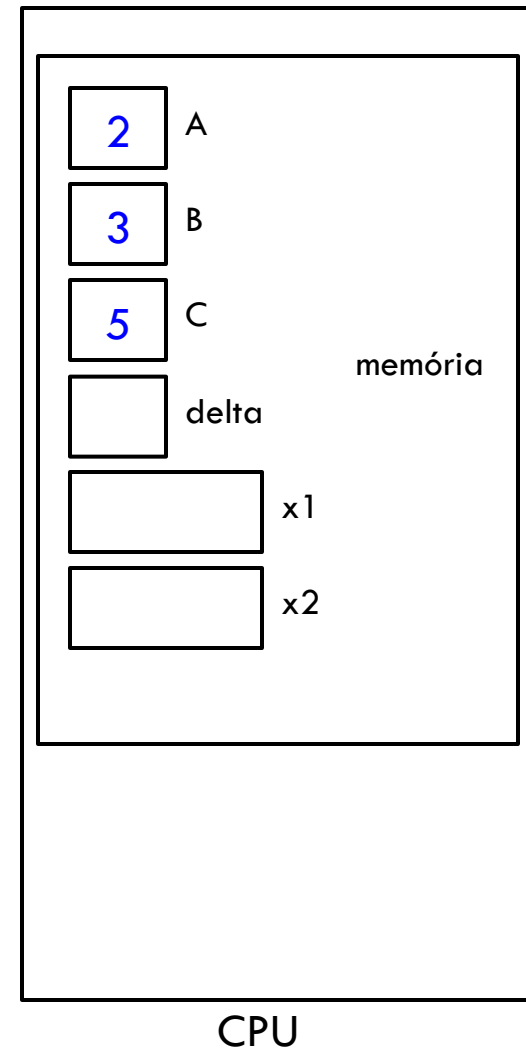
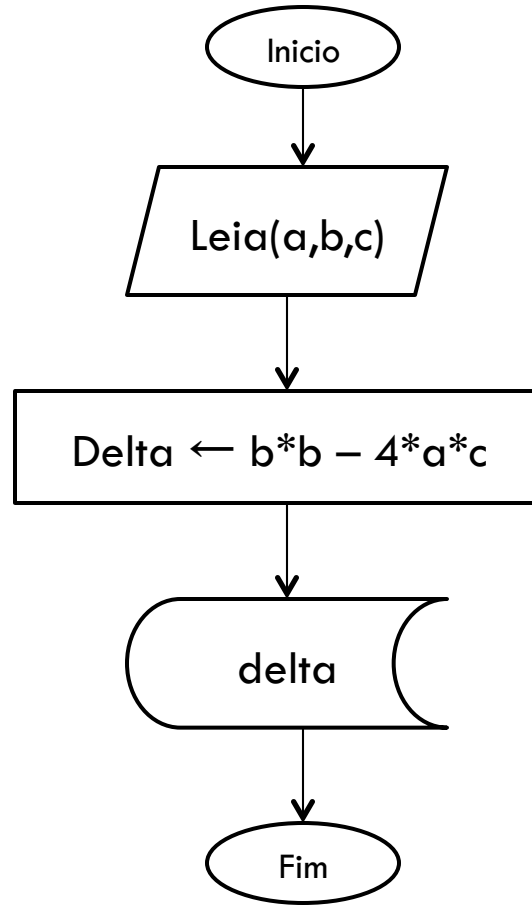
//processamento

delta \leftarrow b*b - 4 * a * c;

//saída

escreva (delta);

fim_algoritmo.



Instância do problema: $2x^2 + 3x + 5$

Variáveis

É um espaço alocado na memória em tempo de execução dos programas para armazenar e representar um valor ou expressão. Enquanto as variáveis só "existem" em tempo de execução, elas são associadas a "nomes", chamados identificadores, durante o tempo de desenvolvimento.

Exemplo:

<nome da variável> : <tipo> ;

Por exemplo:

declare x : inteiro;
 salario: real;

x ← 90;

escreva(sen(x))

x ← 45;

escreva(sen(x))

⇔ x é a variável

45

x

Constante

Dizemos que determinado dado é uma constante quando este dado não sofre nenhuma alteração, ou seja, ele é fixo. Por exemplo, podemos definir uma constante $\pi \leftarrow 3.14$, dentro do algoritmo e, nesse algoritmo, esse valor nunca sofrerá alteração.

A declaração de um dado constante em pseudocódigo terá a seguinte regra sintática:

constante <nome da constante> \leftarrow <valor da constante> <tipo de dado da constante> ;

Por exemplo:

constante $\pi \leftarrow 3.14159265359$ **real;**

const $\pi \leftarrow 3.14159265359$ **real;**

declare const $\pi \leftarrow 3.14$ **real;**

r : real;

leia(r);

escreva("area da circunferencia =", $2*\pi*r*r$);

leia(r);

~~**leia(π);**~~ //errado porque π é constante, logo não muda

3.14

π

r

Identificadores

Os identificadores são os nomes dados às informações de um programa, por exemplo, nome da variável, nome da constante, nome do programa etc.

- ❑ todos os caracteres devem ser letras, números ou o sublinhado (o único caracter especial permitido);
- ❑ o primeiro caracter deve ser uma letra;
- ❑ os demais caracteres podem ser letras, números ou sublinhado;
- ❑ não são permitidos caracteres especiais (com exceção do sublinhado);
- ❑ palavras reservadas (de uma linguagem de programação ou do pseudocódigo) não devem ser usadas como identificadores.

Operadores Aritméticos

Operador	Função	Exemplo
+	adição	$5 + 3$
-	subtração	$2 - a$
*	multiplicação	$b * c$
/	divisão	$d / 10$

Operadores Aritméticos Auxiliares

Operador	Função	Exemplo	Resultado
pot	potenciação	pot(2,3)	$2^3 = 8$
rad	radiciação	rad(9)	$\sqrt{9} = 3$
mod	resto da divisão	7 mod 3	resto da divisão = 1
div	quociente da divisão	7 div 3	quoc. da divisão = 2

pot(5,2) $\Leftrightarrow 5^2$

pot(base,expoente) $\Leftrightarrow b^e$

5 mod 2 $\Leftrightarrow 1$

9 div 4 $\Leftrightarrow 2$

Operadores Unários

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 10;

x++; // ⇔ $x = x + 1$

++x; // ⇔ $x = x + 1$

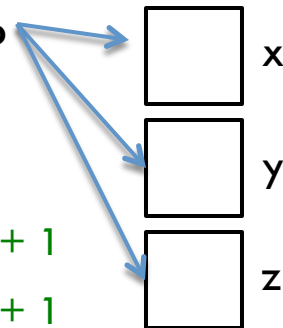
escreva(x); //

y--; // ⇔ $y = y - 1$

--y; // ⇔ $y = y - 1$

escreva(y) //

fim_algoritmo.



Operadores Unários

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ $x = x + 1$

++x; // ⇔ $x = x + 1$

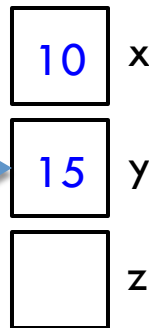
escreva(x); //

y--; // ⇔ $y = y - 1$

--y; // ⇔ $y = y - 1$

escreva(y) //

fim_algoritmo.



Operadores Unários

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ $x = x + 1$

++x; // ⇔ $x = x + 1$

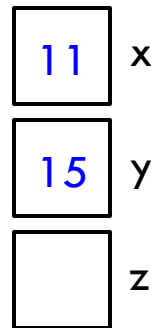
escreva(x); //

y--; // ⇔ $y = y - 1$

--y; // ⇔ $y = y - 1$

escreva(y) //

fim_algoritmo.



Operadores Unários

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ x = x + 1

++x; // ⇔ x = x + 1

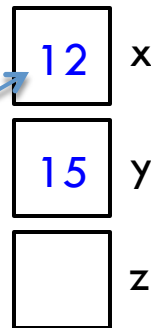
escreva(x); //

y--; // ⇔ y = y - 1

--y; // ⇔ y = y - 1

escreva(y) //

fim_algoritmo.



Operadores Unários

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ $x = x + 1$

++x; // ⇔ $x = x + 1$

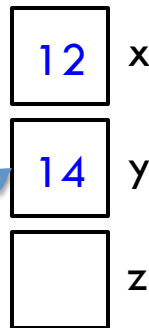
escreva(x); //

y--; // ⇔ $y = y - 1$

--y; // ⇔ $y = y - 1$

escreva(y) //

fim_algoritmo.



Operadores Unários

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ $x = x + 1$

++x; // ⇔ $x = x + 1$

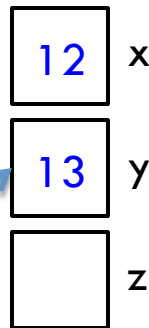
escreva(x); //

--y; // ⇔ $y = y - 1$

y--; // ⇔ $y = y - 1$

escreva(y) //

fim_algoritmo.



Operadores Unários

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ $x = x + 1$

++x; // ⇔ $x = x + 1$

escreva(x); //

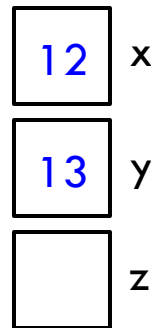
--y; // ⇔ $y = y - 1$

y--; // ⇔ $y = y - 1$

z ← x++;

escreva(y) //

fim_algoritmo.



Operadores Unários

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ $x = x + 1$

++x; // ⇔ $x = x + 1$

escreva(x); //

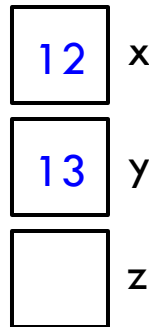
--y; // ⇔ $y = y - 1$

y--; // ⇔ $y = y - 1$

z ← x++;

escreva(y) //

fim_algoritmo.



Operações:

- Atribuição
- incremento

1o. Atribui depois incrementa

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ $x = x + 1$

++x; // ⇔ $x = x + 1$

escreva(x); //

--y; // ⇔ $y = y - 1$

y--; // ⇔ $y = y - 1$

z ← x++;

escreva(y) //

fim_algoritmo.

12 x
13 y
12 z

Operações:

- Atribuição
- incremento

$z \leftarrow x++ \Leftrightarrow$ 1o atribui, depois incrementa.

1o. Incrementa depois atribui

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ x = x + 1

++x; // ⇔ x = x + 1

escreva(x); //

--y; // ⇔ y = y - 1

y--; // ⇔ y = y - 1

z ← x++;

escreva(y) //

fim_algoritmo.

13 x
13 y
12 z

Operações:

- Atribuição
- incremento

z ← x++ ⇔ 1o atribui, depois incrementa.

Operadores Unários

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ x = x + 1

++x; // ⇔ x = x + 1

escreva(x); //

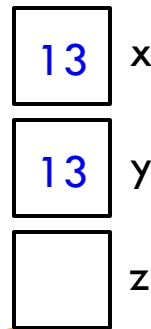
--y; // ⇔ y = y - 1

y--; // ⇔ y = y - 1

z ← x++;

escreva(y) //

fim_algoritmo.



Operações:

- Atribuição
- incremento

z ← x++ ⇔ 1º atribui, depois incrementa.

Operadores Unários

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ x = x + 1

++x; // ⇔ x = x + 1

escreva(x); //

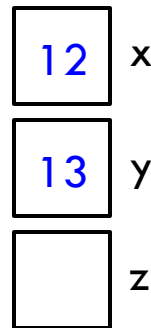
--y; // ⇔ y = y - 1

y--; // ⇔ y = y - 1

z ← ++x;

escreva(y) //

fim_algoritmo.



Operações:

- Atribuição
- incremento

Operadores Unários

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ $x = x + 1$

++x; // ⇔ $x = x + 1$

escreva(x); //

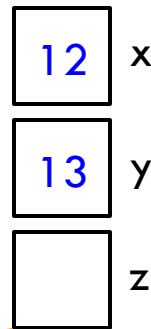
--y; // ⇔ $y = y - 1$

y--; // ⇔ $y = y - 1$

z ← ++x;

escreva(y) //

fim_algoritmo.



Operações:

- Atribuição
- incremento

1o. Incrementa depois atribui

Operador	Função	Exemplo	Equivalência
++	pré-incremento	++ x	$x = x + 1$ (incrementa e logo depois usa o valor de x)
++	pós-incremento	y ++	$x = x + 1$ (usa o valor de x e logo depois incrementa)
--	pré-decremento	-- z	$y = y - 1$ (decrementa e logo depois usa o valor de y)
--	pós-decremento	k --	$y = y - 1$ (usa o valor de y e logo depois decrementa)
+=	atribui a adição	a += 2	$a = a + 2$
-=	atribui a subtração	b -= 5	$b = b - 5$
*=	atribui a multiplicação	c *= 9	$c = c * 9$
/=	atribui a divisão	d /= 7	$d = d / 7$
%=	atribui o resto da div.	e %= 8	$e = e \% 8$

Algoritmo Operações

início_algoritmo

declare x, y, z: inteiro

x ← 10;

y ← 15;

x++; // ⇔ x = x + 1

++x; // ⇔ x = x + 1

escreva(x); //

--y; // ⇔ y = y - 1

y--; // ⇔ y = y - 1

z ← ++x;

escreva(y) //

fim_algoritmo.

13 x
13 y
13 z

Operações:

- Atribuição
- incremento

z ← ++x ⇔ 1o incrementa, depois atribui.

Algoritmo Operações Matemáticas

início_algoritmo

declare x, y: inteiro

r : real

//entrada

leia(x,y)

//processamento

r ← x/y

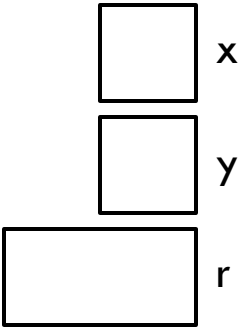
r ← r + x

//saida

escreva(r) //

fim_algoritmo.

+=	atribui a adição	a += 2	a = a + 2
-=	atribui a subtração	b -= 5	b = b - 5
*=	atribui a multiplicação	c * = 9	c = c * 9
/=	atribui a divisão	d / = 7	d = d / 7
%=	atribui o resto da div.	e % = 8	e = e % 8



r += x

salario_liquido ← salario_liquido + valor_hs_extras

salario_liquido += valor_hs_extras

Operadores Relacionais

Operador	Função	Exemplo	Resultado
>	maior que	$3 > 2$	Verdadeiro
<	menor que	$3 < 2$	Falso
> =	maior ou igual a	$5 - 2 > = 1 + 2$	Verdadeiro
< =	menor ou igual a	$5 - 2 < = 1 + 2$	Verdadeiro
=	igual a	$10 = 9$ ==	Falso
< >	diferente de	$10 < > 10$!=	Falso

Operadores Lógicos

Operador	Função	Operador
não	negação	!
e	conjunção	&&
ou	Disjunção	

A	B	não A	A e B	A ou B
V	V	F	V	V
V	F	F	F	V
F	V	V	F	V
F	F	V	F	F

Operadores Lógicos

Operador	Função	Operador
não	negação	!
e	conjunção	&&
ou	Disjunção	

P: Eliane **é** a Professora de Algoritmos.

!P: Eliane **não é** a Professora de Algoritmos.

P	!P
V	F
F	V

Operadores Lógicos

Operador	Função	Operador
não	negação	!
e	conjunção	&&
ou	Disjunção	

MS: Média ≥ 7.0

F: Frequência $\geq 75\%$

Aprovado: MS **E** F

MS	F	Aprovados MS && F
V	V	V
V	F	F
F	V	F
F	F	F

Operadores Lógicos

Operador	Função	Operador
não	negação	!
e	conjunção	&&
ou	Disjunção	

I: ser Inteligente

R: ser Rico

Maria deseja casar-se com uma pessoa
Inteligente **OU** Rica

	I	R	I R
G1	V	V	V
G2	V	F	V
G4	F	V	V
G5	F	F	F

Equivalência da Lógica de Programação com a Lógica Matemática

\wedge \Leftrightarrow `&&` \Leftrightarrow E

\vee \Leftrightarrow `||` \Leftrightarrow OU

\sim \Leftrightarrow `!` \Leftrightarrow Não

Operadores Lógicos

Operador	Função	Operador
não	negação	!
e	conjunção	&&
ou	Disjunção	

I && R
Maria casacom uma
pessoa que seja
Inteligente E Rica

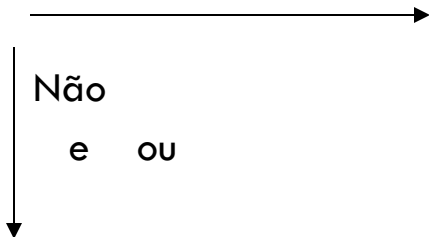
I: ser Inteligente

R: ser Rico

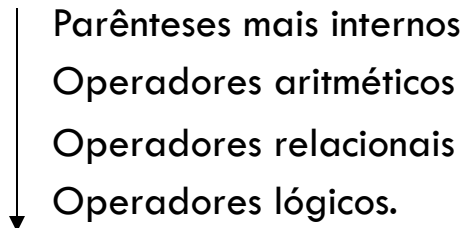
Maria deseja casar-se com uma pessoa
Inteligente OU Rica

	I	R	I R	I && R
G1	V	V	V	V
G2	V	F	V	F
G4	F	V	V	F
G5	F	F	F	F

Prioridades



Prioridades entre operadores



Exemplos com operadores

Expressões aritméticas

a) $5 * 2 - 3 + 14 / 2 + 9$

$$10 - 3 + 7 + 9$$

$$9$$

b) $5 - \text{pot}(2, 3) + 4 - 2 * \text{rad}(4)$

$$5 - 8 + 4 - 2 * 2$$

$$5 - 8 + 4 - 4$$

$$-3$$

c) $\text{pot}(3 - 1, 2) - 7 + \text{rad}(8 + 1) * 2$

$$\text{pot}(2, 2) - 7 + \text{rad}(9) * 2$$

$$4 - 7 + 3 * 2$$

$$4 - 7 + 6$$

$$3$$

Exemplos com operadores

Expressões lógicas com operadores relacionais e aritméticos

a) $5 * 2 = 4 + 10 / 2$
 $10 = \underline{4 + 5}$
 $10 = \underline{9}$
F

b) $5 \bmod 2 + 3 < \text{pot}(3, 2) * 10$
 $1 + 3 < 9 * 10$
 $4 < 90$
V

c) $5 \text{ div } 2 - 1 \geq 4 / 2 + 7$
 $2 - 1 \geq 2 + 7$
 $1 \geq 9$
F

Exemplos com operadores

Expressões lógicas com operadores lógicos, relacionais e aritméticos

a)

$$3 < 7 \text{ e } 5 * 2 = 2 + 1$$

$$3 < 7 \text{ e } 10 = 3$$

V e F

F

c)

não V ou $2 * 3 \bmod 4 / 2 - \text{pot}(2, 1) + 7 = 3$ e $4 * 7 > 12$

não V ou $6 \bmod 2 - 2 + 7 = 3$ e $28 > 12$

não V ou $0 - 2 + 7 = 3$ e V

não V ou $5 = 3$ e V

não V ou F e V

F ou F e V

F ou F

F

b)

$$\text{pot}(2, 3) \leq \text{rad}(9) \text{ ou } 2 * 3 - 6 / 3 = 4$$

$$8 \leq 3 \text{ ou } 6 - 2 = 4$$

F ou 4 = 4

F ou V

V

A	B	A B A v B A ou B	A && B A ^ B A e B	!A ~A Não A
V	V	V	V	F
V	F	V	F	F
F	V	V	F	V
F	F	F	F	V

Exercícios com operadores

1. Calcule o resultado das seguintes expressões aritméticas:

a) $5 * 9 \bmod 2 - 4 / 2$

b) $4 - 7 \operatorname{div} 2 + \operatorname{pot}(2, 4) / \operatorname{rad}(4)$

c) $\operatorname{rad}(\operatorname{pot}(3, 2)) + \operatorname{pot}(2, 3) \bmod \operatorname{rad}(4) - 8 / 2$

d) $\operatorname{rad}(16) \operatorname{div} 9 / 3 - 4 * 3 - \operatorname{pot}(5 + 3, \operatorname{rad}(9))$

e) $\operatorname{rad}(16 \bmod 4) + 5 * 2 / 10 - \operatorname{pot}(8 / 2, 2 \operatorname{div} 2)$

f) $\operatorname{rad}(16) / 2 + 2 * 5 \bmod 3 / 1 * \operatorname{pot}(3, 2 + 4)$

g) $\operatorname{rad}(\operatorname{rad}(16)) + \operatorname{pot}(\operatorname{pot}(2, 2), \operatorname{pot}(2, 3)) + 5 * 4 / 2$

h) $\operatorname{rad}(\operatorname{rad}(\operatorname{pot}(2, 4))) + \operatorname{pot}(\operatorname{rad}(4), \operatorname{pot}(\operatorname{rad}(4), 3)) + 9 - 5 * 4$

i) $5 * 3 \bmod 5 + 4 \operatorname{div} 8 / 2 + 5 \bmod (4 + 2)$

j) $\operatorname{rad}(100) \bmod \operatorname{pot}(4, 3) - \operatorname{pot}(2, 4) \operatorname{div} 8 / 2 + 5 - 3 * 2$

1. Calcule o resultado das seguintes expressões aritméticas:

b) $4 - 7 \div 2 + \text{pot}(2, 4) / \text{rad}(4)$

$4 - 3 + 2^4 / 2$

$4 - 3 + 16 / 2$

$4 - 3 + 8$

$4 - 11$

-7

Exercícios com operadores

2. Calcule o resultado das seguintes expressões lógicas com operadores relacionais e aritméticos:

a) $3 * 8 \bmod 2 = -20 / 4$

b) $5 - 10 \operatorname{div} 3 > \operatorname{pot}(3, 3) / \operatorname{rad}(25)$

c) $\operatorname{rad}(\operatorname{pot}(3, 2)) < \operatorname{pot}(5, 2) \bmod \operatorname{rad}(49) - 81 / 9$

d) $\operatorname{rad}(81) \operatorname{div} 8 / 3 - 4 \geq 5 - \operatorname{pot}(10 - 3, \operatorname{rad}(4))$

e) $\operatorname{rad}(15 \bmod 3) + 15 * 12 / 6 \leq \operatorname{pot}(10 / 5, 12 \operatorname{div} 10)$

f) $\operatorname{rad}(100) / 10 + 12 = 10 \bmod 25 / 5 * \operatorname{pot}(5, 2 + 1)$

g) $\operatorname{rad}(\operatorname{rad}(81)) + \operatorname{pot}(\operatorname{pot}(2, 3), \operatorname{pot}(2, 2)) \leq 15 * 10 / 5$

h) $\operatorname{rad}(\operatorname{rad}(\operatorname{pot}(3, 4))) + \operatorname{pot}(\operatorname{rad}(16), \operatorname{pot}(\operatorname{rad}(4), 2)) \geq 19 - 15 * 2$

i) $27 * 25 \bmod 13 + 51 \operatorname{div} 3 > 12 + 35 \bmod (17 - 12)$

j) $\operatorname{rad}(36) \bmod \operatorname{pot}(2, 3) - \operatorname{pot}(3, 4) \operatorname{div} 18 / 6 < 25 - 13 * 2$

2. Calcule o resultado das seguintes expressões lógicas com operadores relacionais e aritméticos:

a) $3 * 8 \bmod 2 = -20 / 4$

$$3 * 0 = -5$$

$$0 = -5$$

F

Exercícios com operadores

3. Calcule o resultado das seguintes expressões lógicas com operadores lógicos, relacionais e aritméticos

- a) $2 = 10 \bmod 2 \text{ e } -16 > 4$
- b) $5 - 10 \operatorname{div} 3 > \operatorname{pot}(3, 3) \text{ ou } \operatorname{rad}(25) = 5$
- c) **não** $F \text{ e } \operatorname{rad}(\operatorname{pot}(3, 2)) < \operatorname{pot}(5, 2) \text{ ou } 5 \bmod \operatorname{rad}(49) \geq 81 / 9$
- d) **não** $V \text{ ou } \operatorname{rad}(81) \operatorname{div} 8 / 3 - 4 \geq 5 - \operatorname{pot}(10 - 3, \operatorname{rad}(4))$
- e) $\operatorname{rad}(15 \bmod 3) \leq 15 \text{ ou } 12 / 6 \leq \operatorname{pot}(10 / 5, 12 \operatorname{div} 10)$
- f) $\operatorname{rad}(100) / 10 + 12 = 10 \bmod 25 \text{ ou } 5 * \operatorname{pot}(5, 2 + 1) \text{ e não } V$
- g) $\operatorname{rad}(\operatorname{rad}(81)) > 5 \text{ e } \operatorname{pot}(\operatorname{pot}(2, 3), \operatorname{pot}(2, 2)) \leq 15 \text{ e } 10 / 5 = 32$
- h) $\operatorname{rad}(\operatorname{rad}(\operatorname{pot}(3, 4))) > 3 \text{ ou } \operatorname{pot}(\operatorname{rad}(16), \operatorname{pot}(\operatorname{rad}(4), 2)) \geq 19 \text{ ou } -15 * 2 = 7$
- i) **não** $F \text{ ou } 27 * 25 \bmod 13 < 91 \text{ e } 51 \operatorname{div} 3 > 12 \text{ ou } 35 \bmod (17 - 12) \geq 13$
- j) $\operatorname{rad}(36) \bmod \operatorname{pot}(2, 3) \geq 10 \text{ ou não } V \text{ e } -\operatorname{pot}(3, 4) \operatorname{div} 18 / 6 < 25 - 13 * 2$

a) $2 = 10 \bmod 2 \text{ e } -16 > 4$

$2 = 0 \text{ e } F$

$F \text{ e } F$

F

Calcule o resultado das seguintes expressões lógicas com operadores lógicos, relacionais e aritméticos

não F e $\text{rad}(\text{pot}(3, 2)) < \text{pot}(5, 2)$ **ou** $5 \bmod \text{rad}(49) \geq 81 / 9$
V e $\text{rad}(9) < 25$ **ou** $5 \bmod 7 \geq 9$
V e 3 < 25 **ou** 5 ≥ 9
V e V **ou** F
V **ou** F
V

Média Semestral ≥ 7.0
Frequência $\geq 75\%$

Aprovados: $MS \geq 7$ e $F \geq 75\%$

	MS	F	Aprovado? $MS \geq 7$ e $F \geq 75\%$
Grupo1	V	V	V
Grupo2	V	F	F
Grupo3	F	V	F
Grupo4	F	F	F

Maria deseja casar com uma pessoa bonita OU rica

Crítérios da Maria:

- Ser bonita
- Ser rica

	B	R	Maria aceita casar com uma pessoa Bonita OU Rica
Candidatos1	V	V	V
Candidatos2	V	F	V
Candidatos3	F	V	V
Candidatos4	F	F	F

Eliane é Professora e LPA.

Eliane é Profa. de LPA	NÃO é verdade que Eliane é Profa. de LPA
V	F
F	V

Bibliografias

BÁSICA

- GOMES, Ana Fernanda A. Campos, Edilene Aparecida V. Fundamentos da Programação de Computadores – Algoritmos, Pascal e C/C++. Prentice Hall, 2007.
- CARBONI, Irenice de Fátima. Lógica de Programação. Thomson.
- XAVIER, Gley Fabiano Cardoso. Lógica de Programação - Cd-rom. Senac São Paulo – 2007.

COMPLEMENTAR

- FORBELLONE, André Luiz Villar. Eberspache, Henri Frederico. Lógica de Programação – A construção de Algoritmos e Estrutura de Dados. Makron Books, 2005.
- LEITE, Mário - Técnicas de Programação – Brasport - 2006.
- PAIVA, Severino – Introdução à Programação – Ed. Ciência Moderna – 2008.
- PAULA, Everaldo Antonio de. SILVA, Camila Ceccatto da. Lógica de Programação –Viena – 2007.
- CARVALHO, Fábio Romeu, ABE, Jair Minoru. Tomadas de decisão com ferramentas da lógica paraconsistente anotada: Método Paraconsistente de Decisão (MPD), Editora Edgard Blucher Ltda. - 2012.