## TRILHA DE KOTLIN





## **EXPRESSÕES ARITMÉTICAS**

As expressões aritméticas são aquelas em que os operadores são aritméticos e os operandos são valores do tipo numérico (inteiro ou real). Esses valores numéricos podem ser acessados por meio de identificadores constantes ou por meio de variáveis. As operações aritméticas fundamentais são: adição, subtração, multiplicação, divisão, potenciação, divisão inteira e o resto (módulo).







Apresenta os operadores para cada uma dessas operações aritméticas

Operação	Operador	Exemplo
Adição	+	2+3, 9+x
Subtração	-	X - 1, $f(x)-2$
Multiplicação	*	2*1, x*y, 2*g(x)
Divisão	/	1/x, a/z, 2*f(x)
Potenciação	۸	10^x, 2^3
Resto	mod	10 mod 2, 120 mod 10
Divisão Inteira	div	10 div 2, 120 div 10

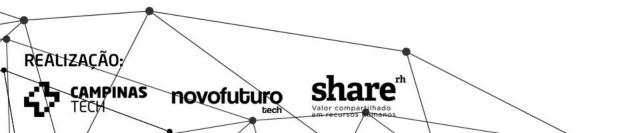






# PRECEDÊNCIA GERAL DOS OPERADORES ARITMÉTICOS

Quando uma expressão aritmética precisa ser avaliada num algoritmo, o analisador processa a expressão dando prioridade para certos operadores. As sub-expressões que contém estes operadores serão avaliadas primeiro e seu valor substituído pela sub-expressão inteira. A seguir a próxima sub-expressão na ordem é avaliada e assim







por diante até que toda a expressão corresponda a um só valor. A tabela abaixo mostra a ordem de prioridade na avaliação dos operadores numa expressão aritmética, chamada de precedência de operadores.

Ordem	Operação	Símbolo
1°	Parênteses	()
2°	Potenciação	**
3°	Multiplicação, Divisão, Resto e Divisão Inteira	*, /, mod, div
4°	Adição, Subtração	+, -







#### **EXPRESSÕES LÓGICAS**

As expressões lógicas são aquelas cujo valor só pode ser verdadeiro ou falso. São compostas por operadores relacionais, operadores lógicos, e por identificadores variáveis ou constantes do tipo lógico. As expressões lógicas também podem ser compostas por resultados de expressões aritméticas.

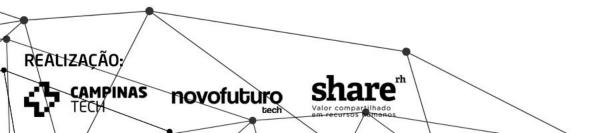






#### **OPERADORES RELACIONAIS**

Os operadores relacionais são aqueles que comparam dois valores do mesmo tipo. O retorno da expressão relacional indica se o resultado da comparação foi verdadeiro ou falso. Por exemplo, a expressão 2 < 3 é uma expressão lógica válida cujo valor é verdadeiro. Em contrapartida, a expressão 2 = 8 é uma expressão lógica também válida, mas cujo valor é falso.







Operador	Símbolo
Igual a	=
Maior que	>
Menor que	<
Maior ou Igual a	>=
Menor ou Igual a	<=
Diferente de	<b>&lt;&gt;</b>

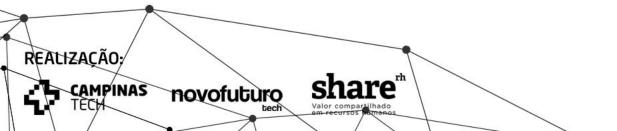






## **OPERADORES LÓGICOS**

Os operadores lógicos são usados para representar situações lógicas que não podem ser representadas por operadores aritméticos. Também são chamados conectivos lógicos por unirem duas expressões simples numa composta. Podem ser operadores binários, que operam em duas sentenças ou expressões, ou unário que opera numa sentença só.







#### **OPERADORES LÓGICOS**

#### Operador logico – E

Duas expressões são unidas por este operador, a expressão resultante só é verdadeira se ambas expressões constituintes também são. Por exemplo "chove **e** venta" só é verdadeiro se as duas coisas forem verdadeiras,







#### **OPERADORES LÓGICOS**

#### **Operador logico – OU**

Neste caso, se qualquer uma das expressões constituintes for verdadeira, a expressão completa também será. Por exemplo, "vou à praia **ou** vou ao campo" é um sentença verdadeira caso qualquer uma das duas ações acontecer, ou ambas. É verdadeira, se eu for a praia e não ao campo, se eu for ao campo e não a praia e se eu for a ambos.







condição1	operação	condição2	resultado
NÃO	E	NÃO	NÃO
SIM	E	NÃO	NÃO
NÃO	E	SIM	NÃO
SIM	E	SIM	SIM
NÃO	ou	NÃO	NÃO
SIM	ou	NÃO	SIM
NÃO	ou	SIM	SIM
SIM	ου	SIM	SIM

PATROCÍNIO:









Para imprimirmos algum tipo de informação na tela do computador utilizamos o comando escreva seguido da informação que será escrita. Dessa forma, se quisermos imprimir uma mensagem como por exemplo "Ola mundo!", isto seria feito com a instrução

escreva ("Olá Mundo")







Para imprimir o valor de uma variável, basta colocar o seu identificador diretamente. O fragmento de código

valor  $\leftarrow$  123

escreva (valor)







Como a sequência **valor** não tem aspas, durante a execução o algoritmo considera ttt como sendo o identificador de uma variável e o substitui pelo seu conteúdo, neste caso 123. Se por outro lado, colocássemos

valor  $\leftarrow$  123

escreva ("valor")







**escreva** (8 < 9)

irá escrever o valor da expressão relacional 8 < 9, neste caso Verdadeiro. Se quiséssemos escrever literalmente 8< 9 sem interpretação, deveriamos delimita-lá com aspas.

escreva ("8 < 9") na tela escreveria: 8 < 9

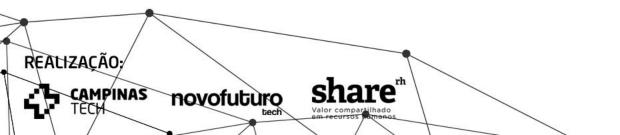
Diversos dados a serem escritos podem ser informados numa mesma instrução separando-os por vírgula ex: **escreva** (8 < 9, 7 > 10)







Da mesma maneira que necessitamos enviar informações de dentro do algoritmo para a saida padrão (em geral a tela), também necessitamos receber informações de fora do algoritmo, a partir da entrada padrão (em geral o teclado). Considere por exemplo um sistema de supermercado, sempre que compramos em uma rede de supermecado (fidelização / garantir descontos), o sistema irá necessitar de algumas informações como, por exemplo: o nosso código de cliente (ou o nome) e o CPF. Essas informações são fornecidas pelo sistema a partir de comandos de entrada de dados.



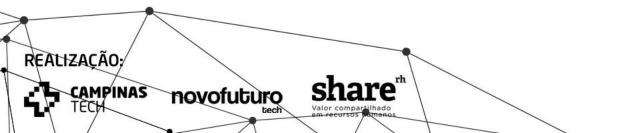




Para realizarmos a entrada de dados utilizaremos o comando leia.

Ao utilizar o comando **leia** o programador deve saber de antemão qual a variável que irá armazenar o valor que será fornecido pelo usuário

Após declaradas as variáveis que receberão os valores fornecidos pelo usuário podemos utilizar o comando **leia** para receber esses valores.







Algoritmo: cliente\_rede\_supermercados

var codigo\_cliente: inteiro

cliente\_cpf: caractere

leia(codigo\_cliente)

leia(cliente\_cpf)

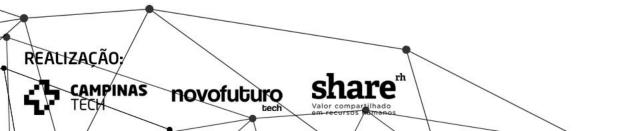






É possível avisarmos o usuário sobre qual tipo de informação o algoritmo está precisando, para isso, utilizamos o comando de saída de dados escreva imediatamente antes do leia que informa ao usuário o que deve entrar.

Por exemplo, no Algoritmo informaria ao usuário o que digitar antes de executar o comando leia e esperar pela entrada do usuário.







Algoritmo: cliente\_rede\_supermercados

var codigo\_cliente: inteiro

cliente\_cpf: caractere

escreval(digite código do cliente e tecle <ENTER>)

leia(codigo\_cliente)

escreval(digite o CPF e tecle <ENTER>)

leia(cliente\_cpf)

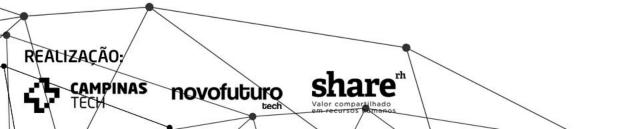






## APLICAÇÃO PRÁTICA COM VISUALG

**BORA POR A MÃO NA MASSA!** 







## **EXPRESSÃO ARITMÉTICA**

VERIFICAR A SAIDA DAS EXPRESSÕES ARITMÉTICA:

$$(2 * 8 + 20 - 2) - ((30 - 25) + 20)$$

$$(3 * (3^2 * 3)) / (50 - (10 DIV 2 * 8))$$

$$((2^4 / 3) + (3 * 8) / (5 MOD 3))$$







## **EXERCÍCIOS**

- CRIAR UM PROGRAMA PARA O CALCULO A ÁREA DE UM CIRCULO
- ENTRAR COM OS DADOS DO RAIO
- FORMULA DO CIRCULO: A = PI \* (RAIO ^2)
- REALIZAR A SAIDA: "O VALOR DA ÁREA É: ", VALOR DA ÁREA DO CIRCULO







## **EXERCÍCIOS**

- CRIAR UM PROGRAMA PARA MOSTRAR O SEU NOME E SOBRE NOME
- SOLICITAR A ENTRADA COM O NOME, SOBRE NOME
- SAIDA: "BEM VINDO " NOME " ", SOBRE NOME

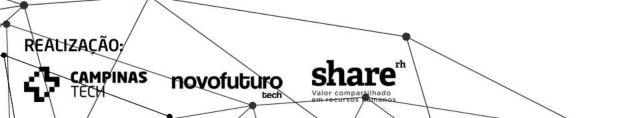






## **EXERCÍCIOS**

- CRIAR UM PROGRAMA PARA VERIFICAR MAIOR NUMERO
- ENTRAR COM O NUMERO1, NUMERO2
- LER OS VALORES
- SE O NUMERO1 FOR MAIOR QUE NUMERO2 DEVE MOSTRAR VERDADEIRO, SENAO MOSTRAR FALSO
- USAR OPERADORES LÓGICOS







REALIZAÇÃO:

CAMPINAS novofuturo

share the valor compartified of the valor co

PATROCÍNIO:

