



Equivalência Portugol – Java

Linguagem Java



Objetivos

- Compreender como desenvolver algoritmos básicos em JAVA
- Aprender como escrever programas na Linguagem JAVA baseando-se na “Linguagem Portugol” – aprender as sintaxes equivalentes entre as linguagens

Analogia

Portugol Studio X Java

Principais Tipos de Dados

Tipos Básicos de Dados

Operadores Aritméticos

Operadores Lógicos

Operadores Relacionais

Operadores Unários

Regras de Precedência

Regras para nomeação em Java

TIPOS DE DADOS

PRINCIPAIS TIPOS DE DADOS EXISTENTES

Descrição	Em PORTUGOL Studio	Em Java
Caractere	caracter	char
Literal	cadeia	String (classe)
Inteiro	inteiro	int
Inteiro longo	Não se aplica	long
Real (ponto flutuante)	real	float
Real “longo”	Não se aplica	double
Booleano	Não se aplica	boolean

Tipos básicos de dados

- **Inteiro (*int*)**: números inteiros sem parte fracionária, podendo ser negativo, nulo ou positivo
- **Real (*float*, *double*)**: números com parte fracionária, podendo ser negativo, nulo ou positivo
 - **float** – 7 casas de precisão após a vírgula
 - **double** – 15 casas de precisão após a vírgula
- **Literal (*char*, *String*)**: conjunto de caracteres alfanuméricos números (0..9), letras (A..Z, a..z) e símbolos (#, ?, !, @.....)
 - Para um caracter usar aspas simples
 - Para um conjunto de caracteres usar aspas dupla



Tipos básicos de dados

- **Lógico (*boolean*)**: poderá assumir valores Verdadeiro ou falso
 - *true* – verdadeiro
 - *false* – falso



Constante

- **Definição**
 - Valor fixo que **NÃO** se modifica durante a execução de um programa
- **Exemplos**
 - Número
 - Valor Lógico (Verdadeiro ou falso)
 - Seqüência de Caracteres
- **Classificação**
 - Numérica
 - Literal
 - Lógica

Constante Numérica

- Nos algoritmos utiliza a notação decimal
- As constantes numéricas podem
 - Possuir ou não uma parte fracionária
 - Ter uma parte exponencial
 - Fator 10 elevado a um número inteiro
- Exemplos
 - 25
 - 3.14
 - 7.8 E10 (idêntico a $7.8 * 10^{10}$)



Constante Numérica

- Pode ser positiva ou negativa
 - Depende do sinal que precede a constante
 - Caso não exista sinal → positiva
 - Expoente também pode possuir um sinal (indica o deslocamento da virgula)
 - Caso não exista sinal → positiva
- Exemplos
 - -3.4
 - 26E-10 (idêntico a $26 \cdot 10^{-10}$)
 - **3,4 (Erro não representa um número em java)**

Os números fracionários sempre serão representados através da notação de ponto



Constante Lógica

- Só pode ser
 - Verdadeiro TRUE 1
 - Falso FALSE 0
- Utilizado em preposições Lógicas

Constante Literal



- Qualquer seqüência de caracteres
 - Letras, dígitos, símbolos Especiais
- Em java Todas as constantes literais que aparecem no algoritmo devem estar entre **aspas**
- Exemplos:
 - “José da Silva”
 - “12345”
 - “26/02/75”
 - “Mensagem”
 - “@#!ABC4”
 - “X1W!Z2”

Constantes Literais

- **Não confunda!!!!**
 - 12345
 - “12345”
 - false
 - “false”

Operadores

- São meios pelos quais
 - incrementamos,
 - decrementamos,
 - comparamos e
 - avaliamos dados dentro do computador
- Temos três tipos de operadores:
 - Operadores **Aritméticos**
 - Operadores **Relacionais**
 - Operadores **Lógicos**

Operadores Aritméticos

Descrição	Em Portugol Studio	Em Java
Multiplicação	*	*
Divisão real	/	/
Divisão inteira	Não se aplica	/
Módulo	%	%
Adição	+	+
Subtração	-	-
Incremento	++	++
Decremento	--	--



Exemplo

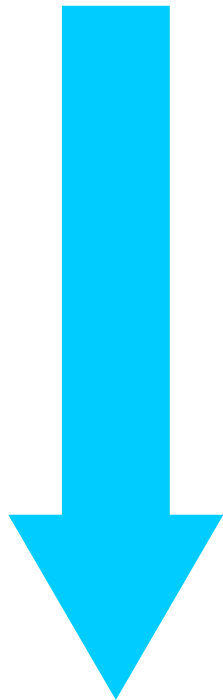
- Qual o resultado da avaliação da expressão

$$10 - 4 * 2 + 1$$

- 13, 18, 3 ?
- Depende da prioridade da avaliação dos operadores

Ordem de Prioridade

Menor



+ -

/ *

()

Maior

Exemplos

$$6 * 4 + 5 - 3$$

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

Operadores Lógicos

- Estabelecem uma relação de comparação entre valores ou expressões
- Resultam sempre em um valor lógico
 - Verdadeiro (**true**) ou falso (**false**)

Operadores Lógicos



Descrição	Em Portugol Studio	Em Java
E	e	&&
OU	ou	
NÃO	não	!

Operadores Lógicos



- Os operadores lógicos mais utilizados são:
 - E → **&&**
 - OU → **||**
 - NÃO → **!**
- Tais operadores retornam valores lógicos como **(V)erdadeiro** ou **(F)also**


‘Eu vou almoçar se tiver macarronada **ou** frango!’


‘Eu vou almoçar se tiver macarronada **e** frango!’

Operadores Lógicos

- Tabela-verdade para os operadores **&&** e **||**

A	B	A && B	A B
V	V	V	V
V	F	F	V
F	V	F	V
F	F	F	F

 **&&** – somente resulta em **VERDADEIRO** quando todas as sentenças avaliadas são verdadeiras

 **||** – somente resulta em **FALSO** quando todas as sentenças avaliadas são falsas

Operadores Lógicos

- O operador **!** (não) faz a negação de uma sentença:

‘Pedro mora na Vila Rica’

- Ao aplicar o operador **!** (não), a sentença passa a ser:

‘Pedro **NÃO** mora na Vila Rica’

A	! A
V	F
F	V

Precedência dos operadores: **!**, **&&** ou **||**

Operadores Relacionais

Descrição	Em Portugol Studio	Em Java
Maior	>	>
Maior ou igual	>=	>=
Menor	<	<
Menor ou igual	<=	<=
Igualdade	==	==
Desigualdade	!=	!=

Operadores Unários



■ Operador de Incremento em Java:

- **Função:** Incrementar o operando em **1 unidade**
- Trabalha de dois modos:
- Pré-fixado → **++num**
 - A variável num é incrementada **antes** de seu valor ser usado
- Pós-fixado → **num++**
 - A variável num é incrementada **depois** de seu valor ser usado

Pré-fixado

```
num = 5;  
x = ++num;
```

x=6 num=6

Pós-fixado

```
num = 5;  
x = num++;
```

x=5 num=6

Operadores Unários



Operador de Decremento em Java:

- **Função:** Decrementar o operando em **1 unidade**
- Trabalha de dois modos:
- Pré-fixado → `--num`
 - A variável `num` é decrementada **antes** de seu valor ser usado
- Pós-fixado → `num--`
 - A variável `num` é decrementada **depois** de seu valor ser usado

Pré-fixado

```
num = 5;  
x = --num;
```

x=4 num=4

Pós-fixado

```
num = 5;  
x = num--;
```

x=5 num=4

Precedência de Operadores JAVA

Operador	Observação
()	Parênteses para agrupar expressões
x++, y--	Incremento e decremento pós-fixado
--x, --y, !	Incremento e decremento pós-fixado, negação
*, /, %	Operadores aritméticos de multiplicação, divisão e módulo • observação: “%” não respeita a propriedade associativa com multiplicação/divisão $2 * 3 \% 5 \neq 2 * (3 \% 5)$
+, -	Operadores aritméticos de adição e subtração
<, >, <=, >=	Operadores relacionais
==, !=	Operadores de igualdade e diferença
&&	Operador lógico e
	Operador lógico ou
=	Operador de atribuição

Regras para nomear variáveis em JAVA



- Pode conter um ou mais caracteres;
- Deve começar sempre por uma letra;
- Pode ser seguidos por letras e números;
 - casa21
- Não pode ter espaços entre as letras;
 - casa 21 (errado!!) – usar_underscore_para_separar
- Não pode conter caracteres especiais;
 - (?, ç, @, #, !, etc....)
- Não pode ser uma palavra reservada;
 - float, int, for , ...
- Ser sucinto e utilizar nomes coerentes

Regras para nomear variáveis



- **Fique atento!!!**

- Os nomes de variáveis abaixo são iguais???

NomeCliente

nomecliente

nomeCliente

- **Não**, pois variáveis em Java são *case-sensitive*
 - nomes com letras maiúsculas são diferenciados de nomes com letras minúsculas!!