

Programação Orientada a Objetos Elementos Básicos da Linguagem Java

1

Apresentação

A utilização da linguagem Java requer conhecimentos sobre as palavras chave e regras de sintaxe para a definição de classes, construtores, métodos e interação entre objetos.

Neste material serão apresentados os elementos fundamentais da linguagem Java.

2

Elementos da linguagem

3

Palavras chave (keywords)

abstract	default	if	private	this
boolean	do	implements	protected	throw
break	double	import	public	throws
byte	else	instanceof	return	transient
case	extends	int	short	try
catch	final	interface	static	void
char	finally	long	strictfp	volatile
class	float	native	super	while
const	for	new	switch	
continue	goto	package	synchronized	

4

Chaves, ponto e vírgula, vírgula, espaços em branco

- Chaves: um bloco é um conjunto de instruções delimitadas por **{ }**
- Ponto e vírgula: uma declaração consiste de uma ou mais linhas de código, que terminada por um **;**

5

Tipos de dados primitivos

- A linguagem Java possui oito tipos básicos de dados, denominados tipos primitivos, que podem agrupados em quatro categorias:
 - Tipos Inteiros
 - Tipos Ponto Flutuante
 - Tipo Caractere
 - Tipo Lógico

6

Tipos de Dados Inteiros

- Existem quatro diferentes tipos de dados inteiros
 - byte (8 bits) (-128 a +127)
 - short (inteiro curto – 16 bits) (-32768 a +32767)
 - int (inteiro – 32 bits) (-2.147.483.648 a +2.147.483.648)
 - long (inteiro longo – 64 bits) (-9.233.372.036.854.755.808 a +9.233.372.036.854.755.807)

7

Tipo de Dados em Ponto Flutuante

- No Java existem duas representações para números em ponto flutuante que se diferenciam pela precisão oferecida:
 - tipo **float** permite representar valores reais com precisão simples (representação interna de 32 bits)
 - tipo **double** oferece dupla precisão (representação interna de 64 bits).
- Deve ser utilizado o ponto como separador de casas decimais.

8

Tipo de Dados Caractere

- O tipo *char* permite a representação de caracteres individuais.
- A delimitação é feita pelo caracter ' (aspas simples)
- Alguns caracteres são considerados especiais pois não possuem uma representação visual, sendo a maioria caracteres de controle e outros caracteres cujo uso é reservado pela linguagem.

9

Tipo de Dados Caractere

Representação	Significado
\n	Pula linha (<i>newline</i> ou <i>linefeed</i>)
\r	Retorno de carro (<i>carriage return</i>)
\b	Retrocesso (<i>backspace</i>)
\t	Tabulação (<i>horizontal tabulation</i>)
\f	Nova página (<i>formfeed</i>)
\'	Apóstrofe
\"	Aspas
\\	Barra invertida
\u223d	Caractere UNICODE 233d
\g37	Octal
\fca	Hexadecimal

10

Tipo de Dados Lógico

- O Java dispõe do tipo lógico *boolean* capaz de assumir os valores:
- *false* (falso) ou
- *true* (verdadeiro).

11

Identificadores

- Identificadores são rótulos/indicadores/*labels* que os programadores utilizam para atribuir dados ou endereços.
- Regras para a criação de identificadores:
 - O 1º carácter deve ser uma letra
 - Não pode conter símbolos %, #, ...
 - Não pode conter espaços
 - São *Case Sensitive*
 - Não podem ser palavras reservadas

12

Operadores

- A linguagem Java oferece um conjunto bastante amplo de operadores destinados a realização de operações aritméticas, lógicas, relacionais e de atribuição.

13

Operador de Atribuição

- Atribuição é a operação que permite definir o valor de uma variável através de uma constante ou através do resultado de uma expressão envolvendo operações diversas.
- Os operadores de atribuição utilizam o símbolo de igual (=).

LEMBRETE: não utilizar “=” em comparações... é incorreto!

14

Operadores Aritméticos

Operador	Significado	Exemplo
+	Adição	$a + b$
-	Subtração	$a - b$
*	Multiplicação	$a * b$
/	Divisão	a / b
%	Resto da divisão inteira	$a \% b$
-	Sinal negativo (- unário)	$-a$
+	Sinal positivo (+ unário)	$+a$
++	Incremento unitário	$++a$ ou $a++$
--	Decremento unitário	$--a$ ou $a--$

15

Operadores Aritméticos (unários)

Operador	Exemplo	Resultado
++ e --	int x=4; int y=0; y=++x;	x é 4 y é 0 y é 5 e x é 5
	int x=4; int y=0; y=x++;	x é 4 y é 0 y é 4 e x é 5
=	x=45;	x é 45
=	x=45; x=2;	x é 45 x é 90
/=	x=70; x/=5;	x é 70 x é 14
%=	x=90; x%=8;	x é 90 x é 2

16

Comparação e Operadores Lógicos

- Além dos operadores aritméticos o Java possui operadores relacionais
- Operadores que permitem comparar valores literais, variáveis ou o resultado de expressões retornando um resultado falso ou verdadeiro.

17

Comparação e Operadores Lógicos

- A linguagem Java possui 6 operadores de comparação.
- Os valores resultantes podem ser atribuídos a variáveis booleanas.

Operador	Significado	Exemplo
<code>==</code>	Igual	<code>a == b</code>
<code>!=</code>	Diferente	<code>a != b</code>
<code>></code>	Maior que	<code>a > b</code>
<code>>=</code>	Maior ou igual a	<code>a >= b</code>
<code><</code>	Menor que	<code>a < b</code>
<code><=</code>	Menor ou igual a	<code>a <= b</code>

18

Comparação e Operadores Lógicos

- O Java também possui operadores lógicos
- Operadores que permitem conectar logicamente o resultado de diferentes expressões aritméticas ou relacionais
- A expressão resultante pode ser composta de várias partes e portanto avaliar expressões complexas.

19

Comparação e Operadores Lógicos

Operador	Operação booleana	Exemplos
&& (AND)	op1 && op2 = resultado true && true = true false && true = false true && false = false false && false = false op1 e op2 são operandos	int x = 10, y = 20, a = 10, b = 30; if((x < y) && (x > b))
 (OR)	op1 op2 = resultado true true = true false true = true true false = true false false = false op1 e op2 são operandos	int x = 10, y = 20, a = 10, b = 30; if((x < y) (x > b))
! (NOT)	!op1	!(x > y)

20

Operador Condicional

- Sintaxe:

<condição> ? <valor verdade> : <caso contrário>

<condição>? Valor booleano ou valor

<valor verdade> Valor a ser atribuído caso o resultado da expressão seja *true*

<caso contrário> Valor a ser atribuído caso o resultado da expressão seja *false*

21

Operador Condicional

- Linhas gerais para o uso do operador condicional

- sintaxe exemplo: `a = x ? b : c;`
- o tipo de dado de “x” deve ser *boolean*, ou “x” deve ser uma expressão que retorna um booleano.
- o tipo de dado de “b” e “c” devem ser compatíveis com “a”.
- uma instrução *if* pode realizar a mesma tarefa que o operador condicional.
- o valor de “a” será o valor de “b” se a condição for *true* e o valor de “c” caso contrário.

22

Conceitos de casting

23

Casting e conversão

- Casting atribui um valor de um tipo (como int ou double) a uma variável de outro tipo (como long ou short).
- Se os dois tipos forem compatíveis, Java realiza a conversão de forma automática.
- Por exemplo: um valor int sempre poderá ser atribuído a uma variável long.

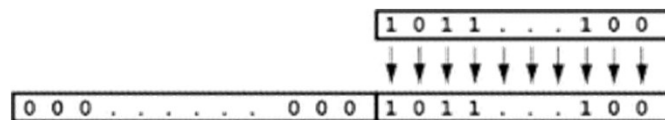
24

Casting e conversão

- Quando a informação tende a ser perdida em uma atribuição, o compilador necessita que o usuário realize um *typecast*.

Exemplo de casting explícito:

```
long a = 99L;  
int b = (int) a;
```



25