
Programação Orientada a Objetos

— Funcionamento e Dados em Java —

Prof. Leandro Rodrigues Pinto
<leandrorodp@gmail.com>

Funcionamento da Linguagem Java - Convenções

Uma linha:
`// Comentário de apenas uma linha`

Múltiplas linhas:
`/*
 Comentário de múltiplas
 linhas em Java
*/`

Documentação (**javadoc**)

```
/**  
 *  
 * @author Prof. Leandro  
 * @since 22/02/2022  
 * @version 1.0.0  
 */
```

Funcionamento da Linguagem Java - Convenções

- **Packages:** escrito em letras minúsculas.
- Um pacote é uma coleção de classes relacionadas que provê acesso protegido e gerenciamento de espaço de nomes
- **Exemplo:**

```
package aula;
```

Funcionamento da Linguagem Java - Convenções

- **Classes e Interfaces:** devem ser substantivos, iniciado sempre em letras maiúsculas e com a primeira letra de cada palavra interna em maiúscula.
- **Exemplo:**

```
class NomeDaClasse;  
  
interface NomeDaInterface;
```

```
class Aula3 {  
  
}
```

Funcionamento da Linguagem Java - Convenções

- **Métodos:** devem ser verbos, com a letra minúscula em primeiro lugar, com a primeira letra de cada palavra interna em maiúscula.
- **Exemplo:**

```
nomeDoMetodo();  
  
correr();  
  
buscarNome();
```

```
void correr() {  
  
}
```

Funcionamento da Linguagem Java - Convenções

- **Atributos ou Variáveis:** Os nomes de variáveis devem ser curtos, mas significativos. A escolha de um nome variável deve ser mnemônico, isto é, concebidos para indicar ao observador casual a intenção da sua utilização.
- **Exemplo:**

```
nomeDaVariavel;  
  
x;  
  
idadePessoa;
```

Funcionamento da Linguagem Java - Convenções

- **Constantes:** Os nomes de variáveis declaradas constantes de classes e de constantes ANSI deve ser todo em letras maiúsculas com palavras separadas por sublinhados ("_").
- **Exemplo:**

```
VALOR_PI = 3.14;
```

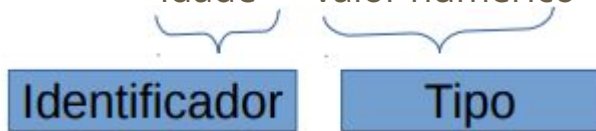
```
MAX_VALOR = 999;
```

```
PEGAR_VALOR_CPU = 1;
```

Dados em Java

Declarando e usando variáveis

- Diferentes tipos de dados, exigem um espaço diferente na memória.
- Em Java, toda variável tem um tipo que não pode ser mudado uma vez que declarado.
- Possui um identificador, que é escolhido pelo programador e não pode ser igual a uma palavra reservada da linguagem.
- Exemplo:
 - Armazenar os dados de uma pessoa: nome, sexo, idade
 - Nome → sequência de caracteres → (Literal)
 - Sexo → uma única letra (M, F) → (Literal)
 - Idade → valor numérico → (Inteiro)



Dados em Java - Palavras Reservadas

<p>→ Arquivos Java class interface enum****</p>	<p>→ Declarações de Controle → Condicionais if else switch case default → Loop while do for → Transferir break continue return</p>		<p>→ Modificadores static final abstract native transient volatile synchronized strictfp**</p>	<p>→ Manipulação de Exceção try catch finally throw throws assert***</p>
<p>→ Tipos de Dados byte short int long float double char boolean void</p>			<p>→ Representação de Objeto this super instanceof</p>	<p>→ Não Usadas const* goto*</p>
<p>→ Locação de Memória new</p>	<p>→ Modificadores de Acessibilidade private protected public</p>	<p>→ Pacote package import</p>	<p>→ Relação de Herança extends implements</p>	
<p>→ Documentação JDK 8.</p> <p>* Não Usado ** Adicionado na versão 1.2 *** Adicionado na versão 1.4 **** Adicionado na versão 5.0</p>				

Dados em Java - Palavras Reservadas

true, false e null: As palavras true , false e null não são consideradas palavras chave porque são na realidade valores literais, contudo, sendo que não podem ser usados como nome de classe, variável ou método, são consideradas palavras reservadas.

Dados em Java - Declaração de Variáveis

Simples	<pre>int idade = 3; float nota = 7.5f; char c = 'A'; boolean aprovado = false;</pre>
Typecast	<pre>int idade = (int) 3; float nota = (float) 7.5; char c = (char) 'A'; boolean aprovado = (boolean) false;</pre>
Wrapper Class	<pre>Integer idade = new Integer(3); Float nota = new Float(7.5); Character c = new Character('A'); Boolean aprovado = new Boolean(false);</pre>

Dados em Java - Tipos primitivos

Família	Tipo Primitivo	Classe Invólucro	Tamanho em bits	Exemplo
Lógico	boolean	Boolean	1 bit	true
Literais	char	Character	16 bits	'H' ou '\u0041'
	----	String	16 bits cada	"JAVA"
Inteiros	byte	Byte	8 bits	127
	short	Short	16 bits	32 767
	int	Integer	32 bits	$2^{31} - 1$
	long	Long	64 bits	$2^{63} - 1$
Reais	float	Float	32 bits	$3.4E^{+38}$
	double	Double	64 bits	$1.8E^{+308}$

Dados em Java - Operadores: Aritméticos

Função	Sinal	Exemplo
Adição	+	$x + y$
Subtração	-	$x - y$
Multiplicação	*	$x * y$
Divisão	/	x / y
Resto da divisão inteira	%	$x \% y$
Sinal negativo	-	-x
Sinal positivo	+	+y
Incremento unitário	++	++x ou x++
Decremento unitário	--	--y ou y--

Dados em Java - Operadores: Relacionais

Função	Caractere(s) utilizado(s)	Exemplo
Igual	==	x == y
Diferente	!=	x != y
Maior que	>	x > y
Maior ou igual a	>=	x >= y
Menor que	<	x < y
Menor ou igual a	<=	x <= y

Dados em Java - Operadores: Atribuição

Função	Caractere(s) utilizado(s)	Exemplo
Simples	=	x = 1
Incremental	+=	x += 2
Decremental	-=	x -= 1
Multiplicação	*=	x *= 6
Divisão	/=	x /= 3
Resto da Divisão	%=	x %= 3

Dados em Java - Operadores: Operador Bit a Bit

Função	Operador	Exemplo
E	&	x & 1
OU		x 2
OU EXCLUSIVO	^	x ^ 1
COMPLEMENTO	~	~x
Deslocamento a Esquerda	<<	x << 2
Deslocamento a Direita	>>	x >> 1
Deslocamento a Direita com Zero	>>>	x >>> 2

Dados em Java - Operadores: Lógicos

Função	Caractere(s) utilizado(s)	Exemplo
E lógico (AND)	&&	x && y
OU lógico (OR)		x y
Negação (NOT)	!	!x

Dados em Java - Operadores: Concatenação

Função	Caractere(s) utilizado(s)	Exemplo
Juntar duas Strings	+	"Olá" + "Mundo."

Dados em Java - Sequência de Escape

Sequência	Caractere especial
\b	Espaço
\f	Form feed (Nova página)
\n	Nova linha
\r	Retorno – início da linha atual
\t	Tabulação horizontal
\"	Mostra aspas dupla
\'	Mostra aspas simples
\\	Mostra barra invertida

Dados em Java - Conversão de tipos

Supondo a variável x	Converter em	y recebe o valor convertido
int x = 10	float	float y = (float) x
int x = 10	double	double y = (double) x
float x = 10.5	int	int y = (int) x
String x = "10"	int	int y = Integer.parseInt(x)
String x = "20.54"	float	float y = Float.parseFloat(x)
String x = "20.54"	double	double y = Double.parseDouble(x)
String x = "JAVA"	Vetor de bytes	byte b[] = x.getBytes()
int x = 10	String	String y = String.valueOf(x)
float x = 10.35	String	String y = String.valueOf(x)
double x = 254.34	String	String y = String.valueOf(x)
byte x[] - (x é um vetor de bytes)	String	String y = new String(x)

Dados em Java - Casting

Na linguagem Java, é possível se atribuir o valor de um tipo de variável a outro tipo de variável, porém para tal é necessário que esta operação seja apontada ao compilador. A este apontamento damos o nome de casting.

É possível fazer conversões de tipos de ponto flutuante para inteiros, e inclusive entre o tipo caractere, porém essas conversões podem ocasionar a perda de valores, quando se molda um tipo de maior tamanho, como um double dentro de um int.

Dados em Java - Casting

<i>DE \ PARA</i>	<i>byte</i>	<i>short</i>	<i>char</i>	<i>int</i>	<i>long</i>	<i>float</i>	<i>double</i>
<i>byte</i>		Implícito	char	Implícito	Implícito	Implícito	Implícito
<i>short</i>	byte		char	Implícito	Implícito	Implícito	Implícito
<i>char</i>	byte	short		Implícito	Implícito	Implícito	Implícito
<i>int</i>	byte	short	char		Implícito	Implícito	Implícito
<i>long</i>	byte	short	char	int		Implícito	Implícito
<i>float</i>	byte	short	char	int	long		Implícito
<i>double</i>	byte	short	char	int	long	float	

DE \ PARA	byte	short	char	int	long	float	double
byte		Implícito	char	Implícito	Implícito	Implícito	Implícito
short	byte		char	Implícito	Implícito	Implícito	Implícito
char	byte	short		Implícito	Implícito	Implícito	Implícito
int	byte	short	char		Implícito	Implícito	Implícito
long	byte	short	char	int		Implícito	Implícito
float	byte	short	char	int	long		Implícito
double	byte	short	char	int	long	float	

Família	Tipo Primitivo	Classe Invólucro	Tamanho em bits	Exemplo
Lógico	boolean	Boolean	1 bit	true
Literais	char	Character	16 bits	'H' ou '\u0041'
	----	String	16 bits cada	"JAVA"
Inteiros	byte	Byte	8 bits	127
	short	Short	16 bits	32 767
	int	Integer	32 bits	$2^{31} - 1$
	long	Long	64 bits	$2^{63} - 1$
Reais	float	Float	32 bits	$3.4E^{+38}$
	double	Double	64 bits	$1.8E^{+308}$

Atividade prática

As **classes** são usadas para **modelar os objetos**, ou seja, uma classe serve como um modelo para armazenar informações e realizar determinadas tarefas.

Um sistema completo é composto, geralmente, por muitas **classes**, que são copiadas na memória do computador durante a execução do programa.

Essa cópia feita na memória do computador no momento em que o programa está sendo executado chama-se **objeto**

Atividade prática

A criação de uma classe deve seguir um padrão, que é composto pelo modificador de acesso, pela palavra reservada `class`, pelo nome da classe e, tudo que estiver entre a abertura do colchete e o final, faz parte da classe, por exemplo:

```
public class Pessoa {
```

```
}
```

Seu trabalho consiste em criar uma classe em Java, usando a IDE Netbeans ou Eclipse, que calcule a área de um triângulo.

Lembre-se que a área do triângulo é calculada pela fórmula: $A = (bxh)/2$

Exercício 1

Em uma certa empresa, há tabelas com o quanto foi gasto em cada mês. Para fechar o balanço do primeiro trimestre, precisamos somar o gasto total.

Sabendo que, em **Janeiro**, foram gastos **15.000** reais, em **Fevereiro**, **23.000** reais e em **Março**, **17.000** reais, faça um programa que calcule e imprima o gasto total no trimestre. Siga esses passos:

- Crie uma classe chamada BalancoTrimestral com um bloco main;
- Dentro do main, declare uma variável inteira chamada gastosJaneiro e inicialize-a com 15.000;
- Crie também as variáveis gastosFevereiro e gastosMarco, inicializando-as com 23.000 e 17.000, respectivamente, utilize uma linha para cada declaração;
- Crie uma variável chamada gastosTrimestre e inicialize-a com a soma das outras 3 variáveis: `int gastosTrimestre = gastosJaneiro + gastosFevereiro + gastosMarco;`
- Imprima a variável gastosTrimestre.

Exercício 2

Adicione código (sem alterar as linhas que já existem) na classe anterior para imprimir a média mensal de gasto, criando uma variável *mediaMensal* junto com uma mensagem.

Para isso, concatenar a String com o valor, usando "Valor da média mensal = " + *mediaMensal*.

Exercício 3

Apresente na tela um exemplo para cada Operador Aritmético da linguagem Java, exibindo uma expressão matemática seguida do resultado do cálculo da expressão.

Exercício 4

Usando os operadores lógicos, apresente na tela uma tabela verdade para as operações AND, OR e NOT.

Exercício 5

Crie uma classe onde sejam realizadas as operações de Casting.

Bibliografia

Cadenhead, Rogers; LEMAY, Laura, Aprenda em 21 dias Java 2. 4. ed. São Paulo: Campus, 2005.

DEITEL, Harvey H .; DEITEL, Paul J. Java: como Programar. 8. Ed. São Paulo: Pearson / Prentice-Hall, 2010.

Perguntas?



Obrigado!