



Prof. Dr. Fernando Almeida
proffernando.almeida@fiap.com.br





DDD (Domain Driven Design)

Java SE e Java EE





Aula 4

Variáveis Primitivas



O QUE VAMOS APRENDER HOJE?

Variáveis Primitivas

1

Variáveis e Constantes

2

Declaração de variáveis

3

Tipos de dados

4

Atribuição de dados

5

Comentários em Java

6

Casting



Variáveis



· Variáveis +

“ Uma variável é um **espaço de memória** utilizado para **armazenar** (**guardar informações**) em memória



Variáveis

• + •

- Em Java, é preciso identificar o **tipo** de dado e o **nome** (identificador).
- Exemplo:

```
1 | int numero;  
2 | String nome;
```

- Como a Linguagem Java é **fortemente tirada**, a declaração do tipo é **obrigatória**

■ □ •

• ●

●

● + • ●

• • •



Tipos de primitivos de dados em Java



Tipos de dados⁺

- Tipos primitivos:

Tipo	Descrição
<code>boolean</code>	Pode assumir os valores <code>true</code> ou <code>false</code>
<code>char</code>	Representa um caractere Unicode de 16 bits para armazenar dados alfanuméricos
<code>byte</code>	Inteiro de 8 bits. Pode assumir valores entre -2^7 e 2^7-1 (de -128 a 127)
<code>short</code>	Inteiro de 16 bits. Pode assumir valores entre -2^{15} e $2^{15}-1$ (de -32.768 a 32.767)
<code>int</code>	Inteiro de 32 bits. Pode assumir valores entre -2^{31} e $2^{31}-1$ (de -2.147.483.648 a 2.147.483.647)
<code>long</code>	Inteiro de 64 bits. Pode assumir valores entre -2^{63} e $2^{63}-1$
<code>float</code>	Número de ponto flutuante de 32 bits. Pode assumir valores entre $\pm 1.40129846432481707e-45$ e $\pm 3.40282346638528860e+38$
<code>double</code>	Número de ponto flutuante de 64 bits. Pode assumir valores entre $\pm 4.94065645841246544e-324$ e $\pm 1.79769313486231570e+308$

- String:

Tipo	Descrição
<code>String</code>	Cadeia de caracteres que usam 2 bytes por caractere. Strings podem ser vazias (zero caractere) e conter qualquer tipo de caractere.

Tipos de dado



O tipo de dado pode ser qualquer um dos tipos **primitivos** (`int`, `float`, `double`...), assim como objeto (classe/interface), nativo do Java (`String`, `ArrayList`...) ou não (`Produto`, `Carro`...)

Declaração de variáveis

tipo do atributo

int

pesoVeiculo;

identificador
(nome do atributo)

Exemplos:

```
float precoProduto;  
int idadeAluno;  
char conceito;  
String nome_Aluno;  
boolean maiorIdade;
```

· Declaração de variáveis dentro do `main()`

```
3  public class Main {  
4      public static void main(String[] args) {  
5          int numero = 20;  
6          String nome = "Silvana";  
7  
8          System.out.println(numero); // vai imprimir "20"  
9          System.out.println(nome); // vai imprimir "Silvana"  
10     }  
11 }
```

Exemplo de identificadores (nome das variáveis)

- O identificador (nome da variável) é formado por caracteres `Unicode`.
- Podem ser formados por letras, cifrão (\$), *underline*(_) e números (**não pode ser iniciado com número**)

```
1 public class Variaveis {
2
3     public static void main(String args[]){
4         String nome;    // válido
5         int $idade;     // válido
6         double 1preco;   // inválido
7         double preco1;  // válido
8         int ___$;       // válido
9         String :nome;   // inválido
10    }
11
12 }
```

- Palavras compostas - Camel Case - nomeAluno, numeroUm



“Java é *Case-Sensitive*!

numeroUm é diferente de numeroum





Contantes



Constantes

“ Uma constante é uma **informação** que não devem ser alterada durante a execução do programa

π

3.141592653589793

números
letras
palavras

• Variáveis finais⁺

- - No Java utilizamos a palavra-chave `final` antes do tipo da variável
 - Exemplo:

```
3 public class Main {  
4     public static void main(String[] args) {  
5         final String msgPadrao = "Olá!";  
6     }  
7 }
```

- A atribuição não precisa ocorrer no momento da sua criação!

• Variáveis finais⁺

- ⁺ • A atribuição não precisa ocorrer no momento da sua criação!

```
3 public class Main {  
4     public static void main(String[] args) {  
5         final String msgPadrao;  
6         msgPadrao = "Olá!";  
7     }  
8 }
```



• Variáveis finais⁺

- ⁺ • Outros exemplos...



```
3  class Main {  
4      public static void main(String[] args) {  
5          final String nomePagina = "home";  
6          final double e = 1.234e2;  
7  
8          System.out.println(nomePagina); // vai imprimir "home"  
9          System.out.println(e); // vai imprimir "1.234e2"  
10     }  
11 }
```

• • • • • • • • + • + • Constantes com static final


- Constantes serão melhor entendidas junto com o conceito de Orientação a Objetos
- Palavras-chave static final

```
3 public class Main {  
4     public static final float PI = 3.1416F;  
5  
6     public static void main(String[] args) {  
7         System.out.println(PI);  
8     }  
9 }
```

- Diferente de uma variável final, uma **constante** precisa receber o seu valor no momento em que ela for declarada



“Por convenção, usamos letras
maiúsculas para declarar
constantes



Atribuição

pesoVeiculo = 1500

identificador
(nome do atributo)

valor do atributo

Exemplos:

```
float precoProduto = 3.5;  
int idadeAluno = 18;  
char conceito = 'A';  
String nome_Aluno = "João";  
boolean maiorIdade = false;
```

Exemplo de atribuição

- + .
- + • Uma vez declarada, a variável deve ser inicializado, e após isso, ser modificada e utilizada. Elas podem ser declaradas e inicializadas em uma mesma linha de código.

```
1 String nome;           // declara uma variável do tipo String
2 nome = "Frederico Maia"; // inicializa com um valor
3 int idade = 21;         // declara e inicializa na mesma linha
4 System.out.println(nome+" "+idade); // imprime o valor das variáveis na tela
```

Operadores Aritméticos

• + •
+ •

| +
media = somaNotas / 2;
Expressão

operador

Exemplos:

Operação	Operador	Expressão Algébrica	Expressão Java
Adição	+	$X + 1$	$X + 1$
Subtração	-	$Y - 2$	$Y - 2$
Multiplicação	*	$K \cdot X$	$K * X$
Divisão	/	$C / 2$	$C / 2$
Resto	%	$X \text{ mod } Y$	$X \% Y$

+
•
•
•
• + •
•
• • •

Precedência de Operadores Aritméticos

• + •

+ •

|

+

Operador	Operação	Expressão Algébrica
* / %	Multiplicação, Divisão e Resto	É o primeiro a ser avaliado. A ordem de avaliação é da esquerda para a direita.
- +	Subtração e soma	É avaliado posteriormente. A ordem de avaliação também é da esquerda para a direita.

•

• +

■ □ •

• •

•

• + • •

• • •

Operadores Relacionais

operador

\perp

(media \geq 6)

Respostas
true ou **false**

Exemplos:

Operação	Operador Matemático	Operador Java	Exemplo	Significado
Igual	=	==	X == Y	X é igual a Y
Diferente	\neq	!=	X != Y	X é diferente de Y
Maior	>	>	X > Y	X é Maior que Y
Menor	<	<	X < Y	X é menor que Y
Maior ou Igual	\geq	>=	X >= Y	X é maior ou igual a Y
Menor ou Igual	\leq	<=	X <= Y	X é menor ou igual a Y

Permite saber a relação existente
entre seus dois operandos

Operadores Lógicos

resulta em **true**

$(2 > 1) \text{ || } (3 < 7)$

resulta em **false**

$(3 > 2) \text{ \&\& } (2 == 2)$

resulta em **true**

$(5 != 0) \text{ || } (1 < 2)$

resulta em **false**

!true

Exemplos:

Operação	Operador Matemático	Operador Java	Exemplo
OU	\vee	<code> </code>	<code>(notaEnem > 6) (notaRedacao == 10)</code>
E	\wedge	<code>&&</code>	<code>(mediaFinal >= 6) && (totalFaltas < 25%)</code>
Negação	\sim	<code>!</code>	<code>!pendenciaDocumento</code>

conectam duas ou mais expressões relacionais



Exemplo **Prático**



```
3 public class Main {
4     public static void main(String[] args) {
5         final float pi = 3.146f;
6         final double e;
7
8         e = 1.234e2;
9
10        String nome = "Estevao Dias";
11        String dataAniversario = "23/05/2000";
12        int lote = 35456;
13        boolean aprovado;
14
15        aprovado = true;
16
17        System.out.println(pi); // vai imprimir "3.146f"
18        System.out.println(e); // vai imprimir "1.234e2"
19        System.out.println(nome); // vai imprimir "Estevao Dias"
20        System.out.println(dataAniversario); // vai imprimir "23/05/2000"
21        System.out.println(lote); // vai imprimir "35456"
22        System.out.println(aprovado); // vai imprimir "true"
23    }
24 }
```



Comentários em Linguagem Java





Comentários no código

única linha, várias linhas, documentação



Comentários em Java

- São textos ignorados pelo compilador, mas que podem ser útil para nós humanos
- Utilizados para explicar funcionalidade, melhorar a compreensão do que foi implementado

```
1 | // texto ignorado pelo compilador
```

Comentários de apenas uma linha (//)

```
1 | /* texto que pode conter
2 |    várias linhas */
```

Comentários de várias linhas (/* */)

```
1 | /** comentários de documentação,
2 |    começam com barra e dois asteriscos */
```

Comentários de documentação (/** */)



Palavras reservadas



Palavras Reservadas

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



Conversões de Variáveis implícitas ou explícitas



Conversões de variáveis

Conversões

IMPLICITAS

Nenhuma sintaxe especial é necessária porque a conversão é de tipo seguro e nenhum dado será perdido. Exemplo:

- byte para int ou long para float → `byte b = 10; int i = b;`

Conversões

EXPLICITAS

As conversões explícitas exigem um operador cast. A conversão é necessária quando as informações podem ser perdidas na conversão ou quando a conversão pode não funcionar por outros motivos. Exemplo:

- Float para long ou int para byte → `int i = 10; byte b = (byte)i;`

Conversões

OUTRAS

O restante dos tipos de conversão disponíveis ou criados no programa. Exemplo:

- String para int → `String s = "123"; int i = Integer.parseInt(s);`

Conversões de dados

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	



Atividades



Atividades

- 1) Escreva um programa em Java que leia duas variáveis inteiras e troque o conteúdo entre elas.
- 2) Escreva um programa em Java que leia um número inteiro positivo e exiba o dobro desse número
- 3) Escreva um programa em Java para calcular e exibir a média ponderada de duas notas, considerando o seguinte critério:
 - 1) Nota 1 (possui peso 6)
 - 2) Nota 2 (possui peso 4)
- 4) Escreva um algoritmo para calcular e exibir o comprimento de uma circunferência, sendo dada o valor de seu raio ($C=2\pi R$).
- 5) Escreva um algoritmo capaz de converter uma temperatura dada em Fahrenheit e exibir o equivalente em Celsius ($C = (5/9) * (F-32)$).
- 6) Escreva um algoritmo para calcular a área de um triângulo, sendo dados a sua base e a sua altura ($area = (base * altura)/2$).

Obrigado e até a próxima aula!

FIAP

Copyright © 2022 | Professor Fernando Luiz de Almeida

Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente proibido sem consentimento formal, por escrito, do professor/autor.