



FLEXTRONICS X



# Oficina de Java

Kalebe Silva

WEB ACADEMY  
Capacitação em Desenvolvimento Web Full-Stack

# Sumário

- **Variáveis e tipos primitivos**
- **Entrada e saída de dados**
- **Estruturas de decisão**
- **Estruturas de repetição**
- **Classes e objetos**
- **Interfaces**

# Sumário

- **Variáveis e tipos primitivos**
- Entrada e saída de dados
- Estruturas de decisão
- Estruturas de repetição
- Classes e objetos
- Interfaces

# Variáveis e tipos primitivos

O que é uma variável?

- No livro “Java: Como Programar” de Deitel, uma variável é descrita como um contêiner que armazena um valor durante a execução do programa.
- Na realidade:



# Variáveis e tipos primitivos

A tipagem da linguagem influencia totalmente na forma em que a utilizamos. Java é uma linguagem estática e fortemente tipada.

```
public class Main {  
    public static void main(String[] args){  
        Integer numeroInteiro = 10;  
        System.out.println(numeroInteiro.getClass());  
        numeroInteiro = "Bom dia";  
        System.out.println();  
    }  
}
```

Console × Shell +

Show Only Latest Clear History

Run Ask AI

./src/main/java/Main.java:9: error: incompatible types:  
String cannot be converted to Integer  
 numeroInteiro = "Bom dia";  
 ^  
1 error

# Variáveis e tipos primitivos

O mesmo exemplo, porém em JavaScript, uma linguagem dinâmica e fracamente tipada.

```
> numeroInteiro = 10 //variável do tipo inteiro
< 10
> typeof(numeroInteiro)
< 'number'
> numeroInteiro = "Boa tarde!"
< 'Boa tarde!'
> typeof(numeroInteiro)
< 'string'
> numeroInteiro + 10
< 'Boa tarde!10'
```

# Variáveis e tipos primitivos

Tipos primitivos: Números inteiros

Tipo	Tamanho (bits)	Faixa	Valor Padrão
byte	8	-128 a 127	0
short	16	-32.768 a 32.767	0
int	32	$-2^{31}$ a $2^{31} - 1$	0
long	64	$-2^{63}$ a $2^{63} - 1$	0L

# Variáveis e tipos primitivos

Tipos primitivos: Números de ponto flutuante

Tipo	Tamanho (bits)	Faixa
float	32	IEEE 754 $\pm 1,40129846432481707e-45$ a $3,40282346638528860e+38$
double	64	IEEE 754 $\pm 4,94065645841246544e-324$ a $1,79769313486231570e+308$



# Variáveis e tipos primitivos

Além destes, a linguagem possui os tipos primitivos, char e boolean.

- O tipo char permite armazenar um caractere Unicode, utilizando para isso 16 bits.
  - O Unicode é um padrão da indústria para representar dados relacionados a texto, incluindo letras, símbolos e caracteres especiais. Valor padrão para o tipo char: `'\u0000'`
- Podemos armazenar um conjunto de caracteres usando um tipo especial de referência denominado String (que é, na verdade, uma classe), o qual será visto posteriormente.
- O tipo boolean armazena o valor lógico true or false, possuindo o tamanho de 1 bit.

# Variáveis e tipos primitivos

## Operação matemáticas

Operação	Descrição	Exemplo
+	Adição	<code>int a = 5 + 3;</code>
-	Subtração	<code>int b = 5 - 3;</code>
*	Multiplicação	<code>int c = 5 * 3;</code>
/	Divisão	<code>int d = 5 / 3;</code>
%	Módulo	<code>int e = 5 % 3;</code>
(int)	Casting	<code>int f = (int) 5.5;</code>

# Variáveis e tipos primitivos

Repositório:

<https://abre.ai/webacademy01>

# Sumário

- Variáveis e tipos primitivos
- **Entrada e saída de dados**
- Estruturas de decisão
- Estruturas de repetição
- Classes e objetos
- Interfaces

# Entrada e saída de dados

Declarando uma classe:

- Todo programa Java consiste em pelo menos uma classe que você (a/o programador/a) define.



# Entrada e saída de dados

Comentando programas:

- Inserimos comentários para documentar programas e aprimorar sua legibilidade. O compilador Java ignora os comentários, portanto, eles não fazem o computador realizar qualquer ação quando o programa é executado.

Existem três tipos de comentários, sendo um voltado para documentação.

```
// Comentário de uma linha

/*
  Comentário grande
*/

/**
 * Comentário de documentação
 */
```

# Entrada e saída de dados

Cuidado com os comentários:

- À medida que você escreve novos programas ou modifica aqueles existentes, mantenha seus comentários atualizados com o código.
- Esquecer um dos delimitadores de um comentário tradicional ou Javadoc causa um erro de sintaxe.



# Entrada e saída de dados

O que é um método:

- Na prática, um método é como se fosse uma ação, ele faz alguma coisa. Em outras linguagens de programação, um método é similar a uma função.





# Entrada e saída de dados

Declarando um método:

- O método **main** é o ponto de partida de cada aplicativo Java.

```
public static void main(String[] args)
```

- Os **parênteses** indicam um bloco de construção do método
- O **void** significa que o método não vai retornar nada após sua execução

# Entrada e saída de dados

Gerando saída com System.out.println:

- O System.out é conhecido como objeto de saída padrão.
- Ele posiciona o cursor do mouse no final do último caractere escrito.
- Pode ser utilizado para debugar o código

```
public static void main(String[] args) {  
    System.out.println("Hello world!");  
    System.out.println();  
}
```

# Entrada e saída de dados

Exibindo múltiplas linhas de texto com uma única instrução:

- Em uma única instrução é possível quebrar a mensagem em várias linhas, utilizando um caractere de **nova linha** (/n).

```
public static void main(String[] args) {  
    System.out.println("Hello\nworld!\nquebrando\na\nlinha\nvarias\nvezes");  
    System.out.println();  
}
```

- Saída no console:

```
Hello  
world!  
quebrando  
a  
linha  
varias  
vezes
```

# Entrada e saída de dados

Exibindo texto com printf:

- O printf ou print formatado, permite que algumas formatações especiais sejam realizadas na saída de dados.

```
int a = 10;  
String b = "bom dia";  
  
System.out.printf("%s%n%s%n", a, b);
```

- Saída:

```
10  
bom dia
```

# Entrada e saída de dados

Declarando um import:

- Java é bom porque é velho.
- Milhares de bibliotecas, prontas para serem utilizadas, você não precisa “reinventar a roda”.

```
import java.util.Scanner;
```

# Atividade

- 1 - Escreve um código que satisfaça as sentenças a seguir, de forma incremental.
  - {Crie uma pasta com seu nome e sobrenome para guardar os códigos criados.}
  - a) Crie um programa que imprima as quatro operações matemáticas básicas (soma, subtração, multiplicação e divisão).
  - b) Crie um input (entrada de dados) para cada variável, desta forma possibilitando que o usuário digite os números que serão guardados nas variáveis, após isso, altere a saída criada anteriormente para uma saída formatada.
  - c) crie comentários simples, que informem o que cada parte do código faz.
  - d) transforme os comentários simples, em comentários javadoc.
  - e) gere a documentação do código com o comando: *javadoc -d docs nomeDoArquivo.java*.