

JAVA - BÁSICO

Prof. André Aparecido da Silva

Disponível em: <http://www.oxnar.com.br/aulas/algoritmos>



ATIVIDADES DA AULA PASSADA

```
1 public class Ola
2 {
3     public Ola ()
4     {
5         System.out.println("OLÁ TURMA 2º INF");
6     }
7     public static void main (String args [])
8     {
9         new Ola();
10    }
11 }
12
```

General Output

```
-----Configuration: <Default>-----
OLÁ TURMA 2º INF

Process completed.
|
```

```
1 public class Ola
2 {
3     public Ola ()
4     {
5         System.out.println("OLÁ TURMA 2° INF");
6     }
7     public static void main (String args [])
8     {
9         new Ola();
10    }
11 }
12
```

NOME E TIPO DE ACESSO A CLASSE

General Output

-----Configuration: <Default>-----

OLÁ TURMA 2° INF

Process completed.

```
1 public class Ola
2 {
3     public Ola ()
4     {
5         System.out.println("OLÁ TURMA 2º INF");
6     }
7     public static void main (String args [])
8     {
9         new Ola();
10    }
11 }
12
```

INICIO E FIM DA CLASSE

General Output

```
-----Configuration: <Default>-----
OLÁ TURMA 2º INF

Process completed.
|
```

```
1 public class Ola
2 {
3     public Ola ()
4     {
5         System.out.println("OLÁ TURMA 2º INF");
6     }
7     public static void main (String args [])
8     {
9         new Ola();
10    }
11 }
12
```

**INICIO E FIM DO OBJETO QUE
EXECUTARÁ A CLASSE.**

General Output

```
-----Configuration: <Default>-----
OLÁ TURMA 2º INF

Process completed.
|
```

```
2 {  
3 public Ola ()  
4 {  
5     System.out.println("OLÁ TURMA 2º INF");  
6 }  
7 public static void main (String args [])  
8 {  
9     new Ola();  
10 }  
11 }  
12
```

**INICIO E FIM D CONSTRUTOTO
MÉTOCO**

General Output

```
-----Configuration: <Default>-----  
OLÁ TURMA 2º INF  
Process completed.  
|
```

**RESULTADO DA EXECUÇÃO DA
CLASSE.**

```
1 public class Ola
2 {
3     public Ola ()
4     {
5         System.out.println("OLÁ TURMA 2° INF");
6     }
7     public static void main (String args [])
8     {
9         new Ola();
10    }
11 }
12
```

General Output

```
-----Configuration: <Default>-----
OLÁ TURMA 2° INF

Process completed.
|
```

```
1 public class Ola
2 {
3     public Ola ()
4     {
5         System.out.println("OLÁ TURMA 2º INF");
6     }
7     public static void main (String args [])
8     {
9         new Ola();
10    }
11 }
12
```

MÉTODO CONSTRUTOR

General Output

```
-----Configuration: <Default>-----
OLÁ TURMA 2º INF

Process completed.
|
```

```
1 public class Ola
2 {
3     public Ola ()
4     {
5         System.out.println("OLÁ TURMA 2º INF");
6     }
7     public static void main (String args [])
8     {
9         new Ola();
10    }
11 }
12
```

General Output

-----Configura

OLÁ TURMA 2º INF

Process completed.

MÉTODO EXECUTOR

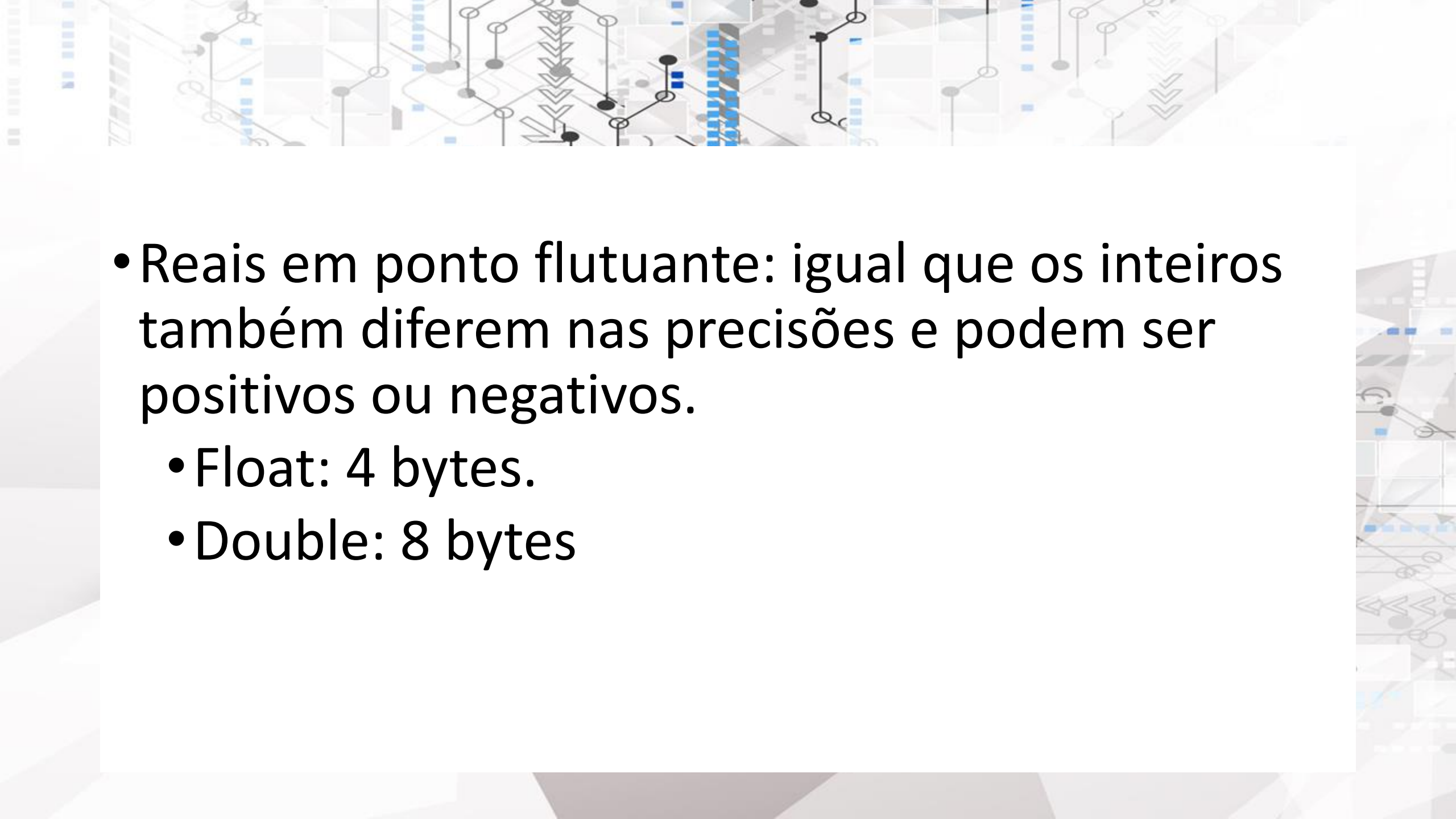
TIPOS PRIMITIVOS JAVA

- Boolean: Não é um valor numérico, só admite os valores true ou false.
- Char: Usa o código UNICODE e ocupa cada caractere 16 bits.
- Inteiros: Diferem nas precisões e podem ser positivos ou negativos.
 - Byte: 1 byte.
 - Short: 2 bytes.
 - Int: 4 bytes.
 - Long: 8 bytes.
-

The background is a complex, abstract composition of overlapping geometric shapes, primarily triangles and polygons, in shades of gray and white. Overlaid on these shapes are various technical and network-like elements: thin black lines connecting circular nodes, some of which are highlighted in blue; dashed blue lines; and several circular icons containing arrows, suggesting movement or flow. The overall aesthetic is modern and digital, resembling a network map or a technical diagram.

TIPOS PRIMITIVOS JAVA

- Boolean: Não é um valor numérico, só admite os valores true ou false.
- Char: Usa o código UNICODE e ocupa cada caractere 16 bits.
- Inteiros: Diferem nas precisões e podem ser positivos ou negativos.
 - Byte: 1 byte.
 - Short: 2 bytes.
 - Int: 4 bytes.
 - Long: 8 bytes.

- 
- Reais em ponto flutuante: igual que os inteiros também diferem nas precisões e podem ser positivos ou negativos.
 - Float: 4 bytes.
 - Double: 8 bytes

- String é uma classe contida no pacote `java.lang` e que descende da classe `java.lang.Object`.

- De forma conceitual podemos dizer que é uma “sequência de caracteres usada como uma única unidade”.
- Os valores armazenados nesse tipo de variável são representados como **literais de string** – sequência de caracteres entre aspas duplas. Os valores podem ser caracteres normais (letras e números) ou caracteres especiais como +, -, * / e \$.

Para textos:

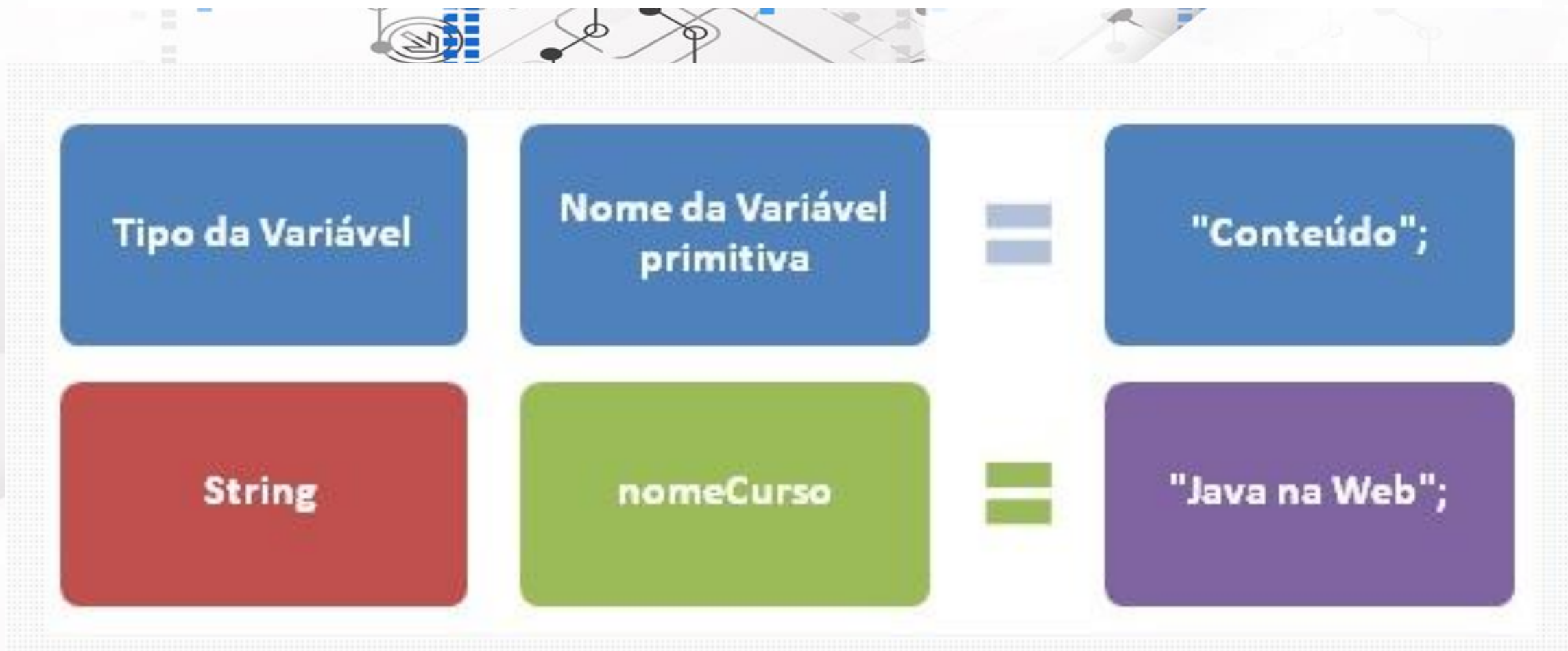
- Use o tipo String;

String Nome = "Julia Caroline Bertoldi";

String Endereco = "Av. Batel, 1500 Batel, Curitiba / Pr."

String Colegio = "Colegio Estadual Maria Aguiar Teixeira"

O processo de declaração de um tipo string é formado pelos seguintes itens:



Declaração de variável primitiva



NOMES DAS VARIÁVEIS

- Deve ser significativo;
- Usar letras minúsculas ou só a primeira maiúscula.
- Não usar caracteres especiais como por exemplo ç.

PALAVRAS RESERVADAS JAVA

Modificadores de acesso

private: acesso apenas dentro da classe

protected: acesso por classes no mesmo pacote e subclasses

public: acesso de qualquer classe

PALAVRAS RESERVADAS JAVA

Modificadores de classes, variáveis ou métodos

abstract: classe que não pode ser instanciada ou método que precisa ser implementado por uma subclasse não abstrata

class: especifica uma classe

extends: indica a superclasse que a subclasse está estendendo

final: impossibilita que uma classe seja estendida, que um método seja sobrescrito ou que uma variável seja reinicializada

implements: indica as interfaces que uma classe irá implementar

interface: especifica uma interface

native: indica que um método está escrito em uma linguagem dependente de plataforma, como o C

new: instancia um novo objeto, chamando seu construtor

PALAVRAS RESERVADAS JAVA

Controle de fluxo dentro de um bloco de código

break: sai do bloco de código em que ele está

case: executa um bloco de código dependendo do teste do switch

continue: pula a execução do código que viria após essa linha e vai para a próxima passagem do loop

default: executa esse bloco de código caso nenhum dos testes de switch-case seja verdadeiro

do: executa um bloco de código uma vez, e então realiza um teste em conjunto com o while para determinar se o bloco deverá ser executado novamente

else: executa um bloco de código alternativo caso o teste if seja falso

for: usado para realizar um loop condicional de um bloco de código

if: usado para realizar um teste lógico de verdadeiro ou falso

instanceof: determina se um objeto é uma instância de determinada classe, superclasse ou interface

return: retorna de um método sem executar qualquer código que venha depois desta linha (também pode retornar uma variável)

switch: indica a variável a ser comparada nas expressões case

while: executa um bloco de código repetidamente enquanto a condição for verdadeira

PALAVRAS RESERVADAS JAVA

Tratamento de erros

assert: testa uma expressão condicional para verificar uma suposição do programador

catch: declara o bloco de código usado para tratar uma exceção

finally: bloco de código, após um try-catch, que é executado independentemente do fluxo de programa seguido ao lidar com uma exceção

throw: usado para passar uma exceção para o método que o chamou

throws: indica que um método pode passar uma exceção para o método que o chamou

try: bloco de código que tentará ser executado, mas que pode causar uma exceção

PALAVRAS RESERVADAS JAVA

Primitivos

boolean: um valor indicando verdadeiro ou falso

byte: um inteiro de 8 bits (signed)

char: um caracter unicode (16-bit unsigned)

double: um número de ponto flutuante de 64 bits (signed)

float: um número de ponto flutuante de 32 bits (signed)

int: um inteiro de 32 bits (signed)

long: um inteiro de 64 bits (signed)

short: um inteiro de 32 bits (signed)

PALAVRAS RESERVADAS JAVA

Controle de pacotes

import: importa pacotes ou classes para dentro do código

package: especifica a que pacote todas as classes de um arquivo pertencem

PALAVRAS RESERVADAS JAVA

Variáveis de referência

super: refere-se a superclasse imediata

this: refere-se a instância atual do objeto

PALAVRAS RESERVADAS JAVA

Retorno de um método

void: indica que o método não tem retorno

PALAVRAS RESERVADAS JAVA

Palavras reservadas não utilizadas

const: Não utilize para declarar constantes; use `public static final`

goto: não implementada na linguagem Java por ser considerada prejudicial



EXERCÍCIOS

EXERCÍCIO 01

- CRIE UMA CLASSE QUE RECEBE E IMPRIME SEUS DADOS EM VARIÁVEIS ESPECÍFICAS E POSTERIORMENTE OS IMPRIMA;
(Dados: Nome, Telefone, Endereço, Idade, Cargo e salário)
`System.out.println(“”+nome da variável);`

EXERCÍCIO 02

CRIE UMA CLASSE QUE:

CRIE UMA CLASSE QUE CONVERTA UM CERTO NUMERO CELSIUS PARA FAHRENHEIT.

$$F = C * (9/5) + 32$$

Sendo C a quantidade de graus Celsius.

$$F = C (1.8) + 32$$

$$C = (F - 32) / 1.8$$

$$K = C + 273.15$$

EXERCÍCIO 03

- CRIE UMA CLASSE QUE RETORNE A MÉDIA ENTRE 5 NÚMEROS