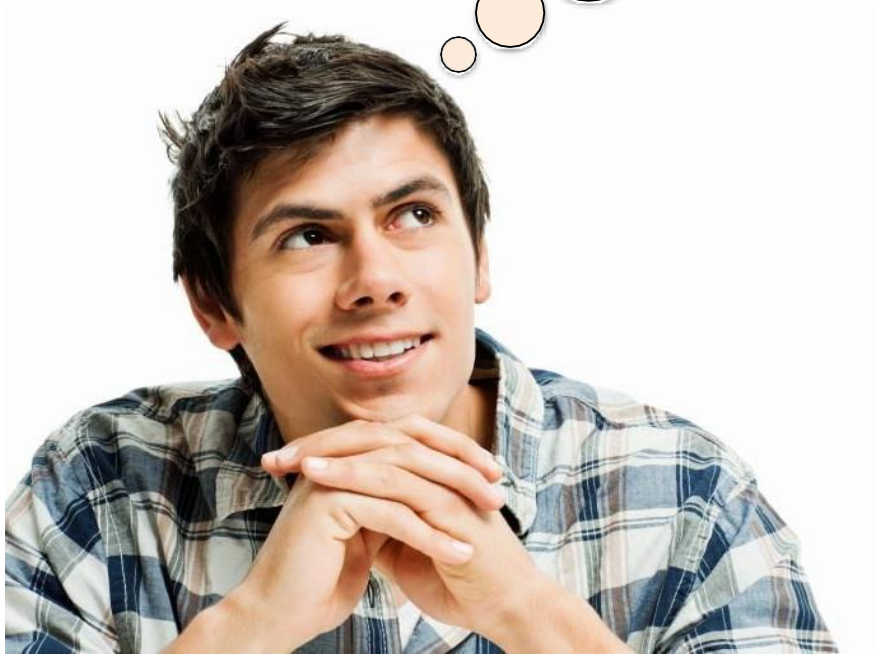
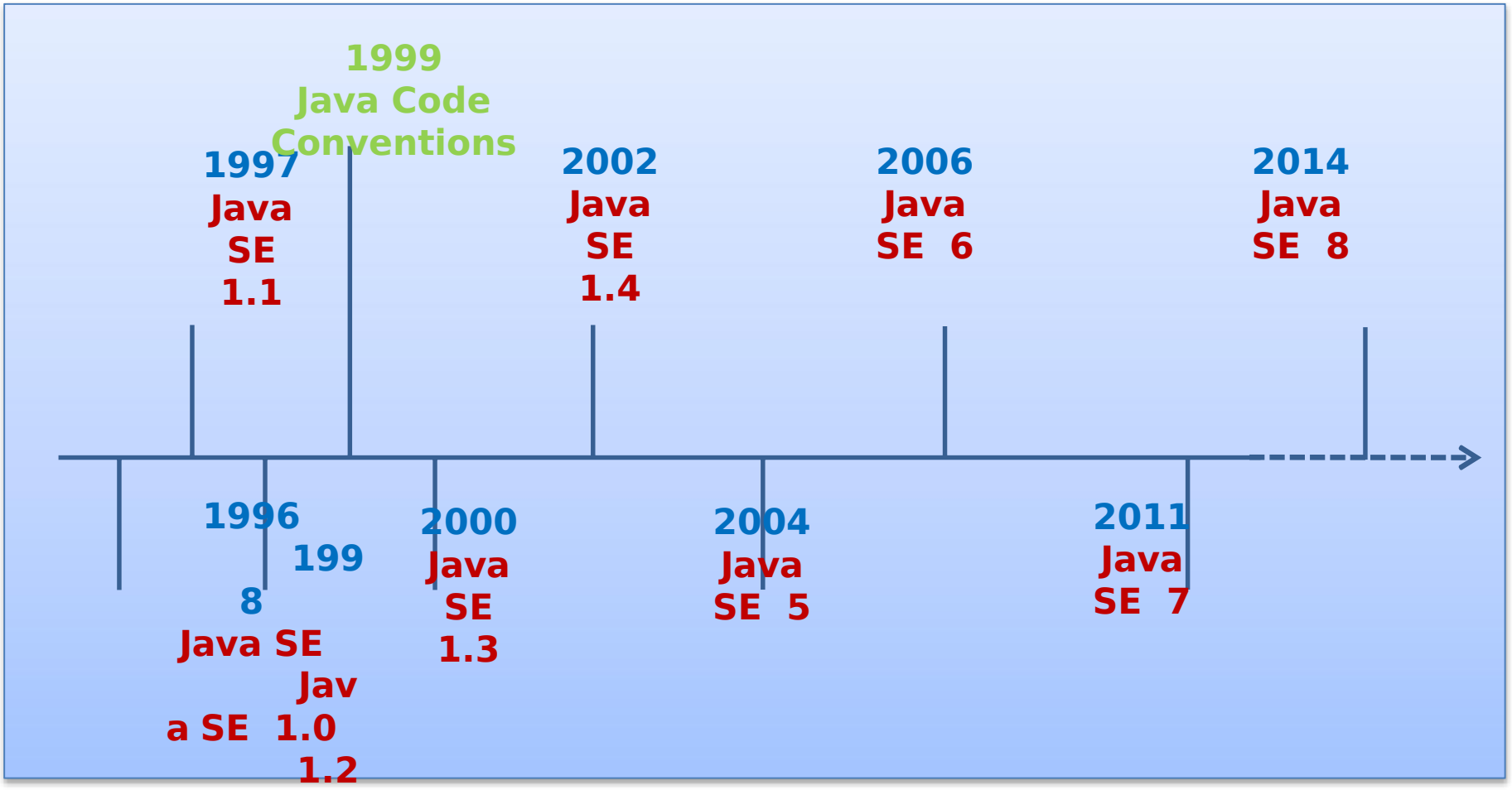
Java Orientado a Objeto Introdução à linguagem Java

O que é Java?

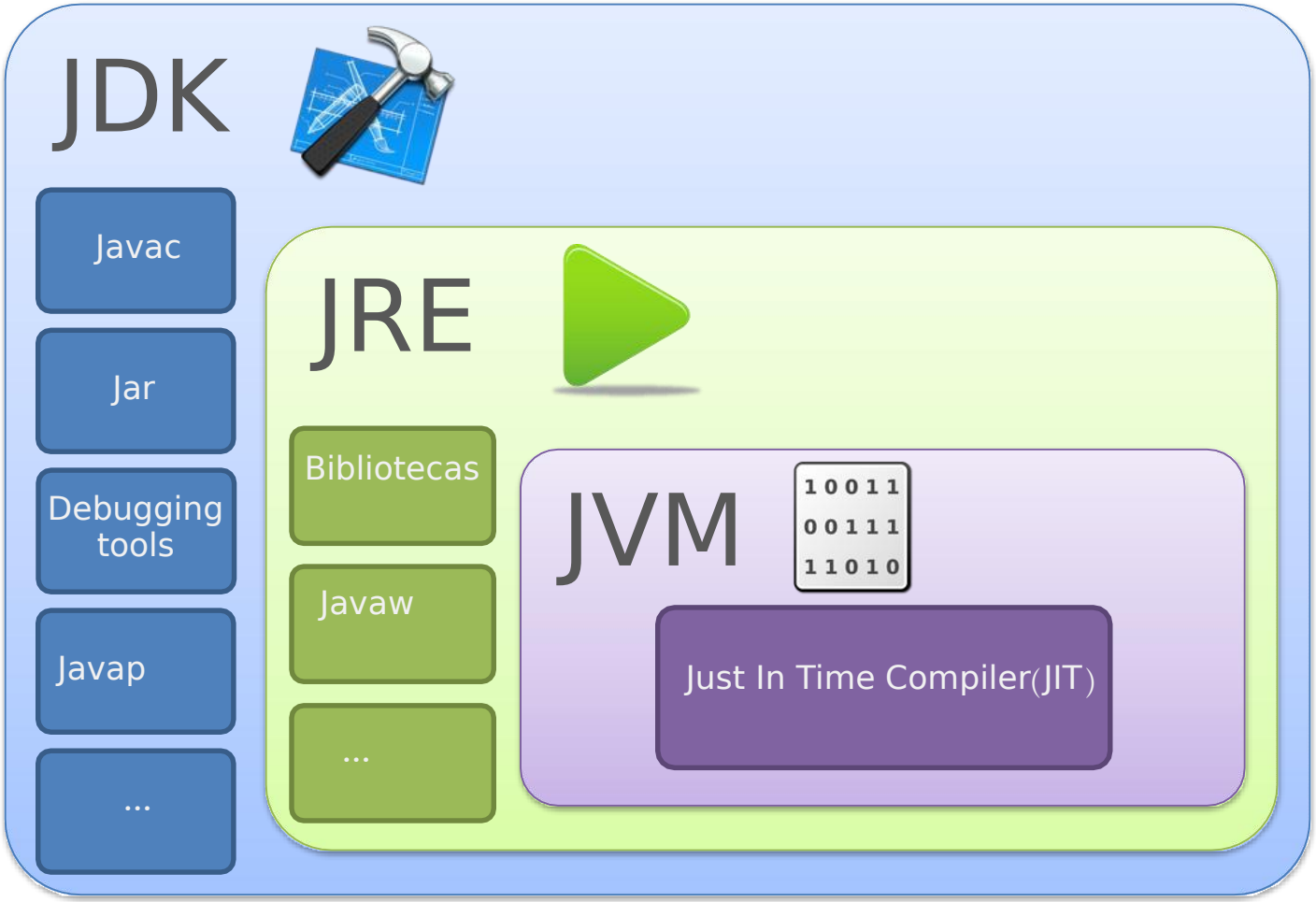


Um pouco da História

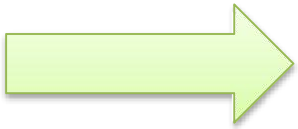


Onde encontro Java ?

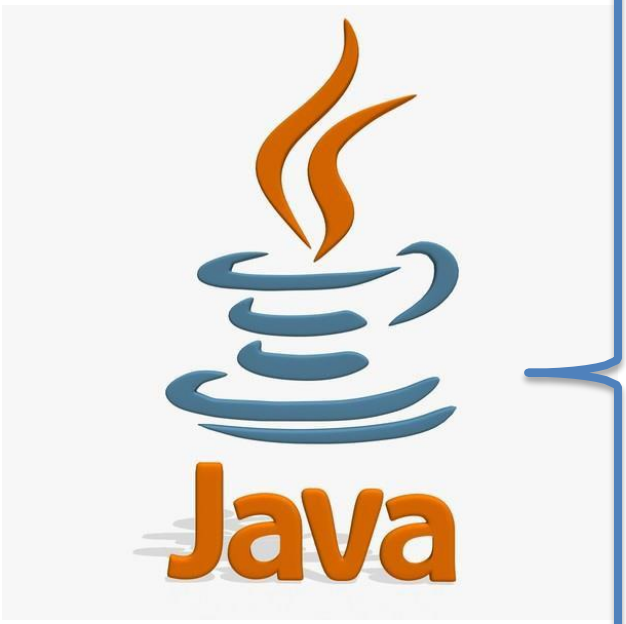
Plataforma Java



Garbage Collection (Coletor de lixo)

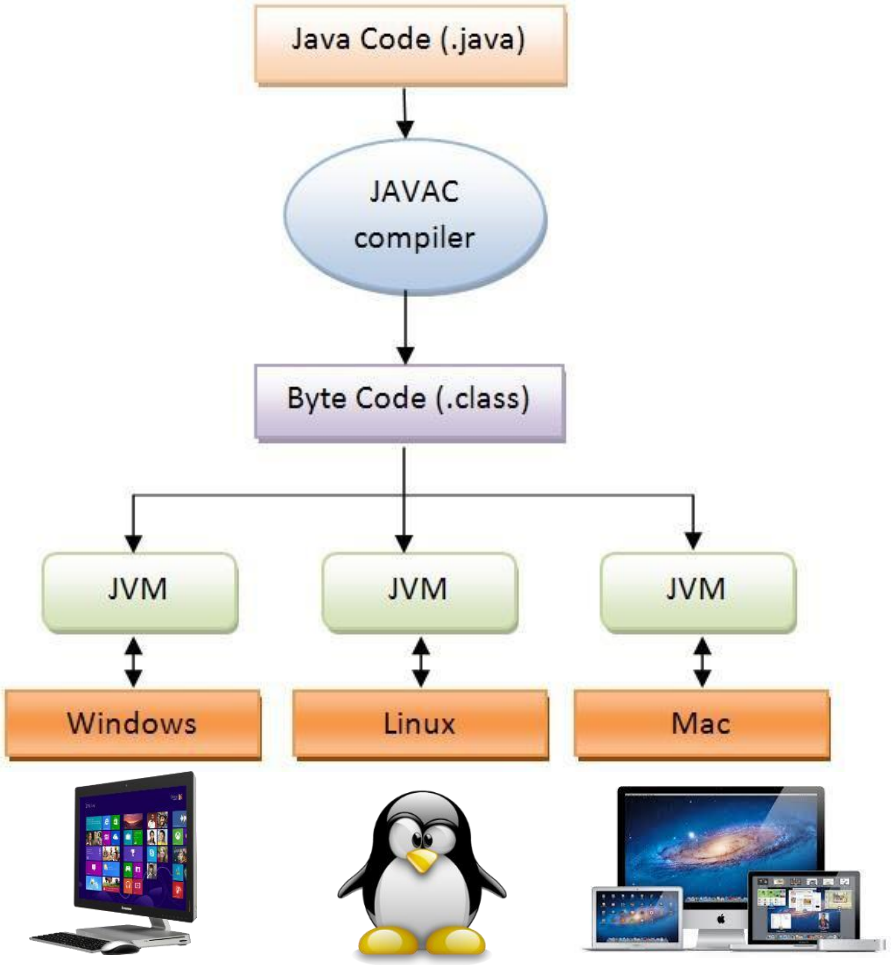


Divisão da Plataforma



- Java FX
- Java SE
- Java ME
- Web Services

Fases do Programa Java



Aplicações mais comuns em Java

Applets

Servlets

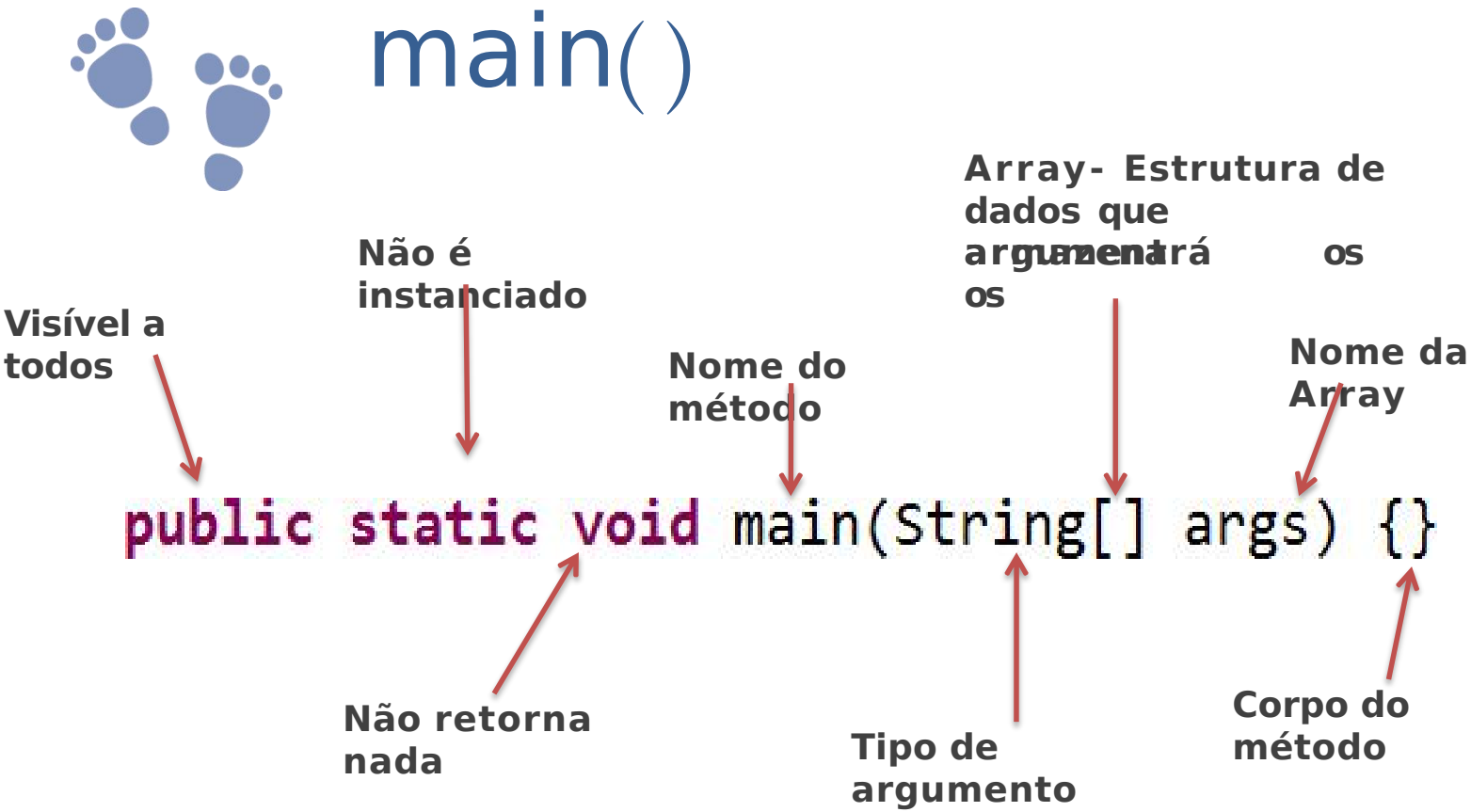
JSP

JSF

EJB

JPA

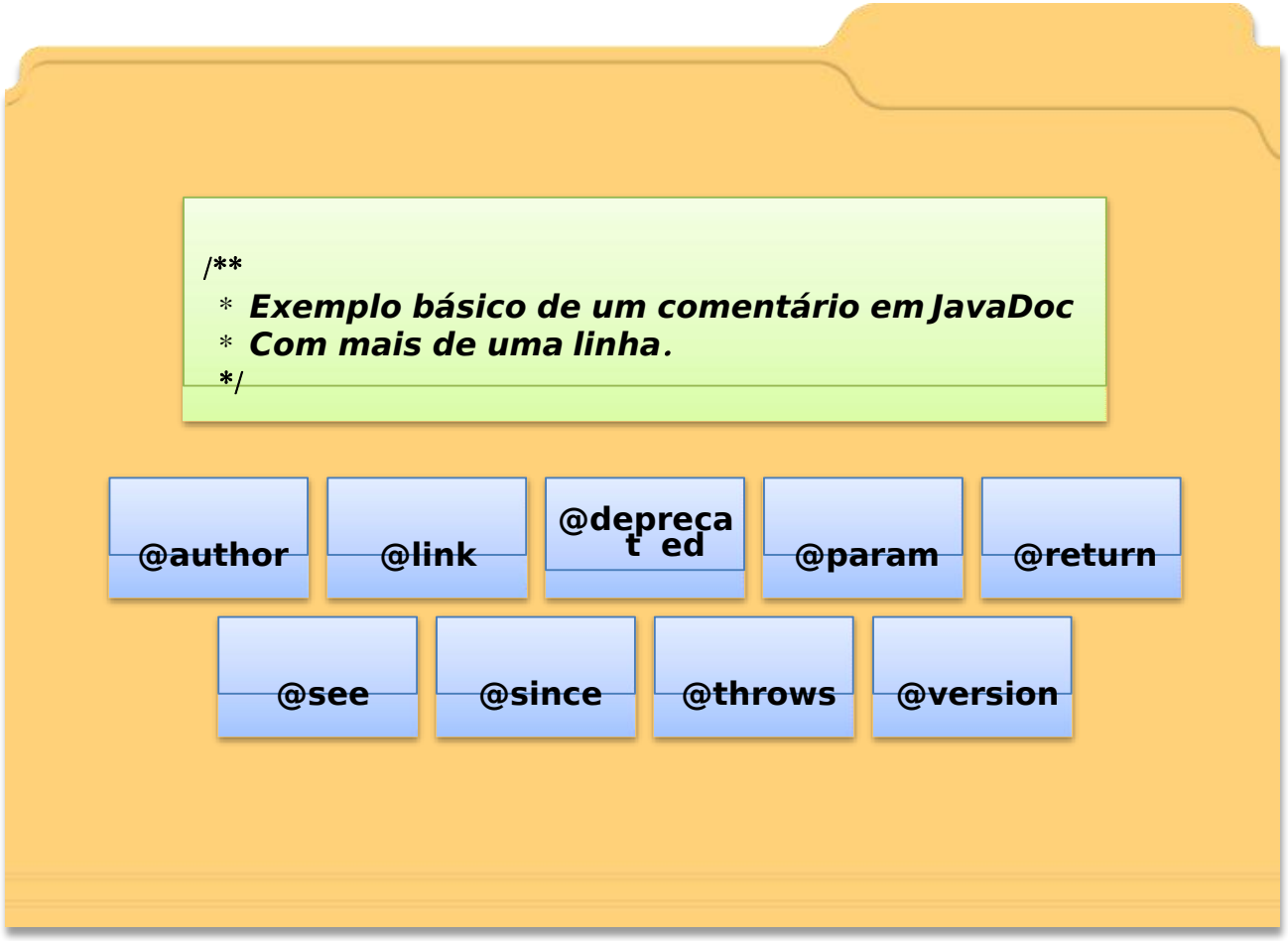
O método main()



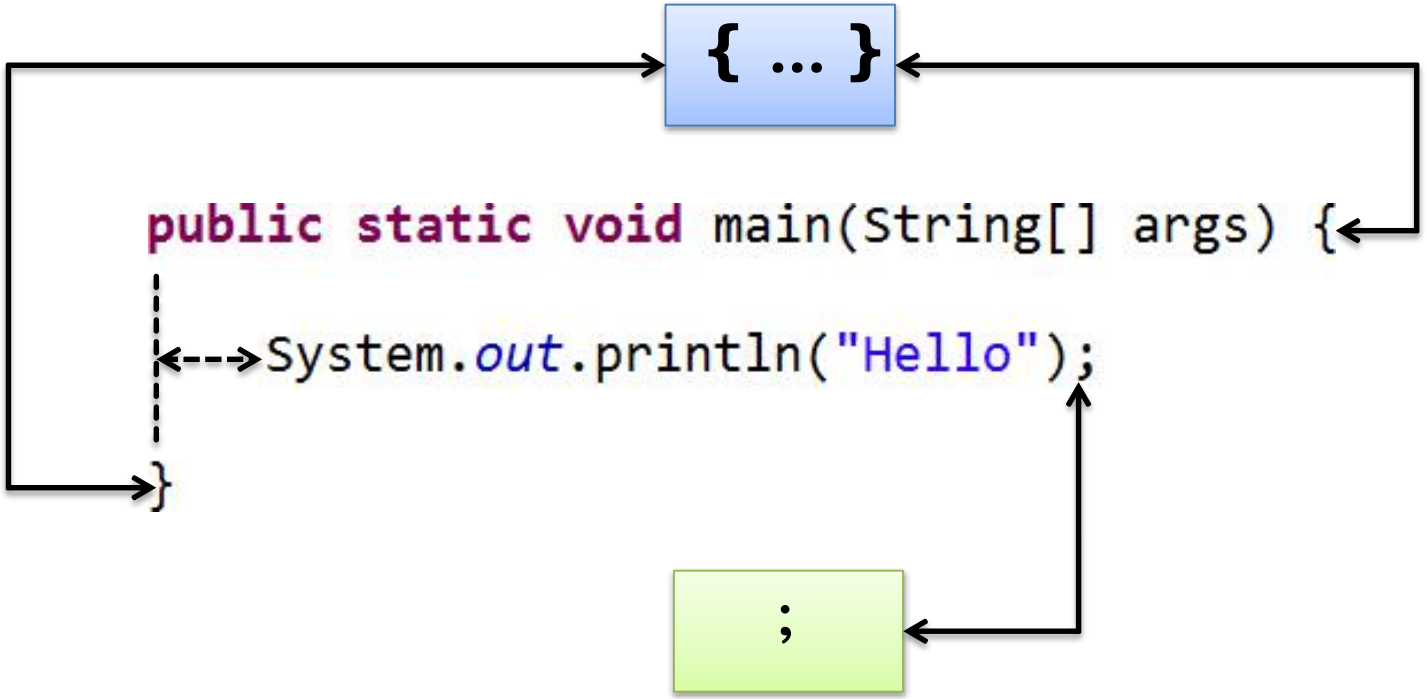
Baby Steps

Java Orientado a Objetos
Identificadores, palavras-chave e tipo

JavaDoc


/**
* Exemplo básico de um comentário em JavaDoc
* Com mais de uma linha.
*/
@author
@link
@deprecated
@param
@return
@see
@since
@throws
@version

Ponto-e-Vírgula, Blocos e Espaço



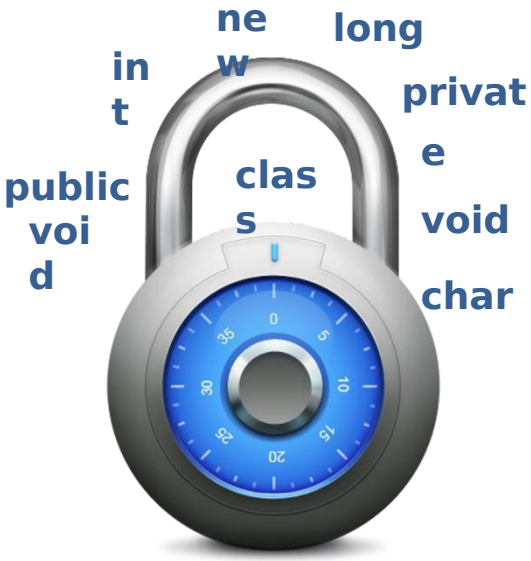
Identificadores e Palavras Reservadas

case-sensitive

hello ≠ **Hello**

n u m e r o
n ú m e r o
a_com_b
var01
_100

n ^ m e r o
0001var
a-com-b
v a r / 0 1 ,
..100

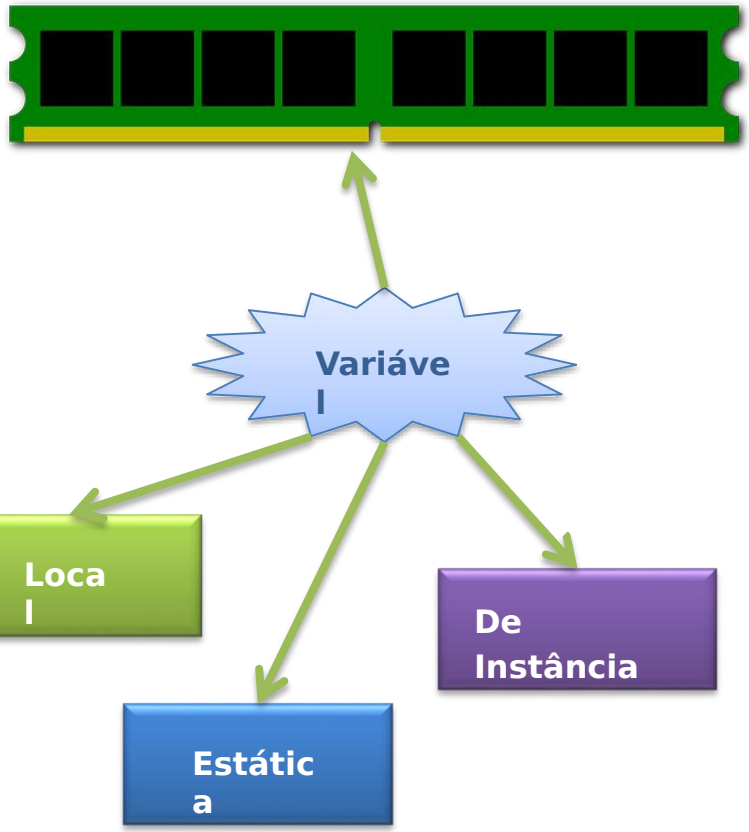


Variáveis, Declaração e Atribuição

Atribuindo Variável

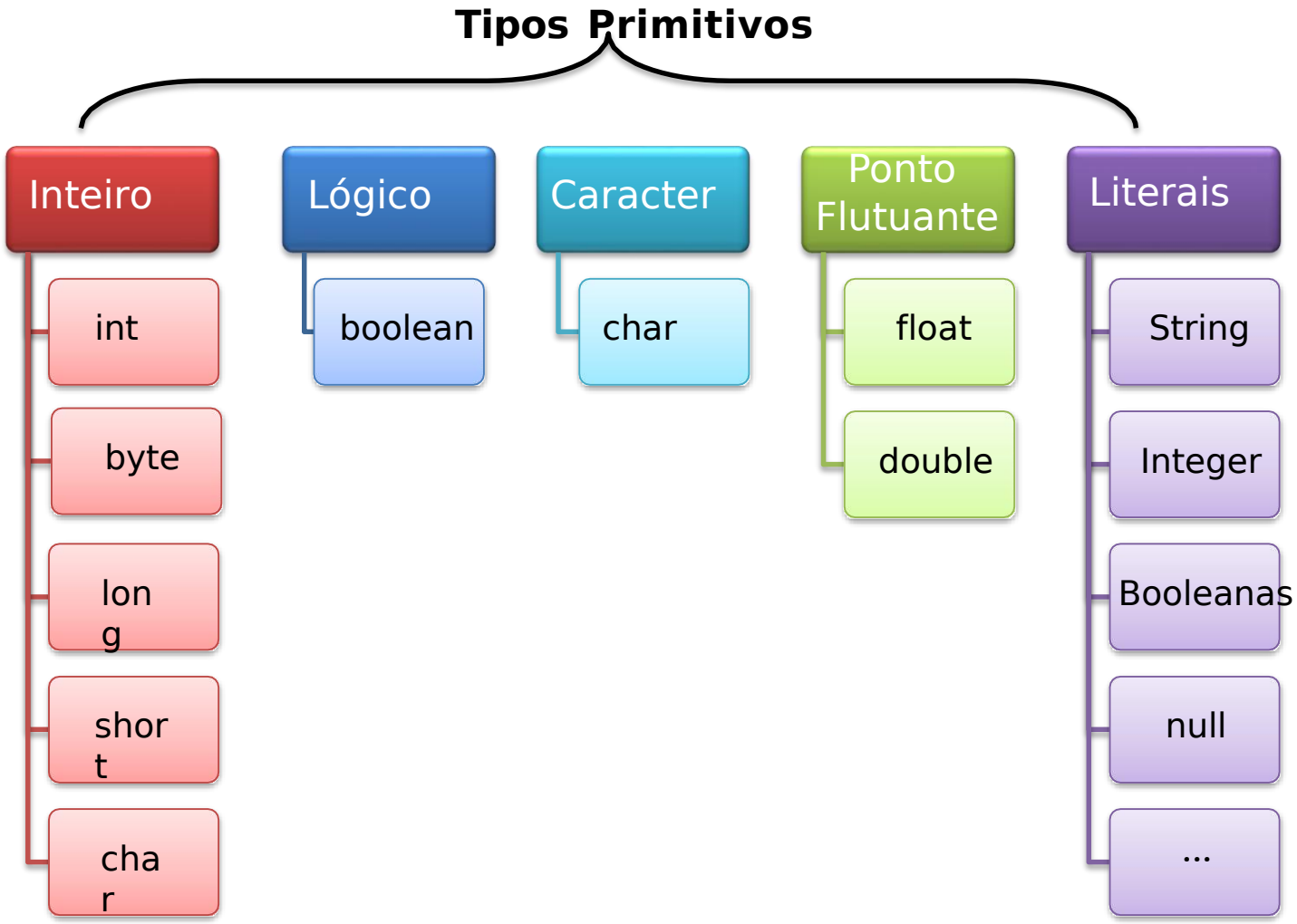
<tipo do dado> <nome> = [valor inicial];

Declarando Variável

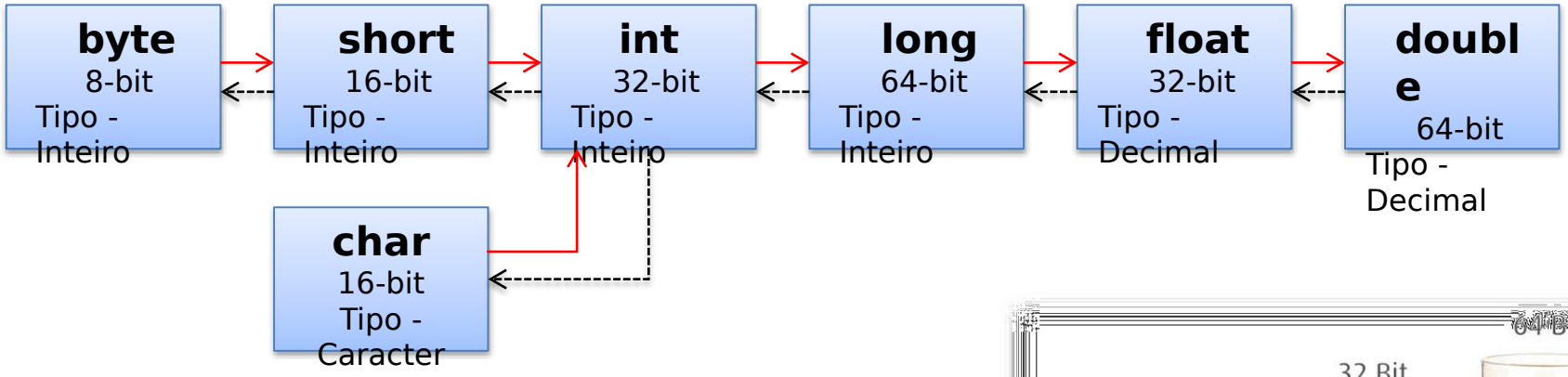


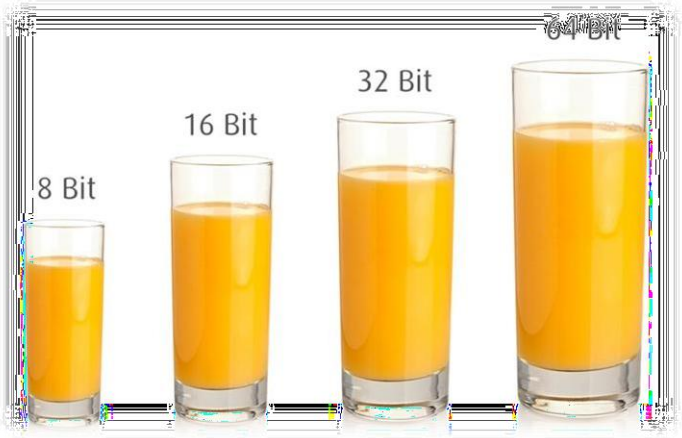
Variáveis devem ser declaradas e inicializadas antes de serem utilizadas.

Tipos de Dados



Casting de Tipos Primitivos

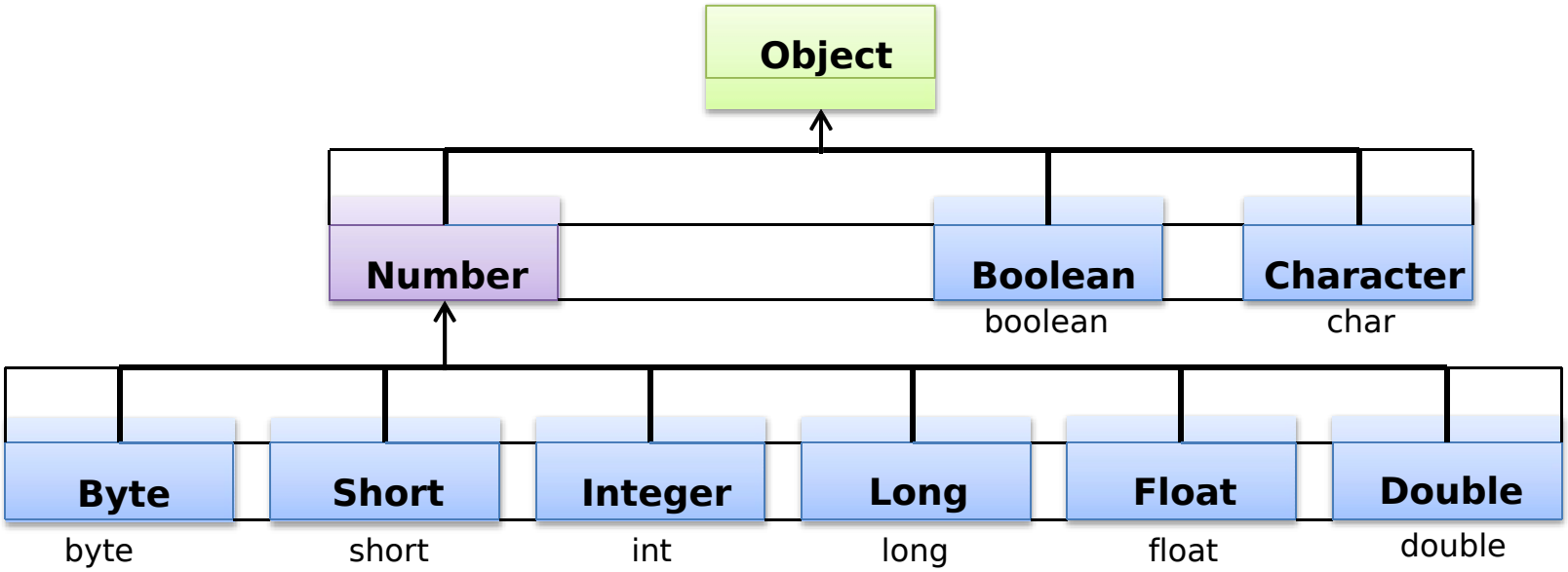




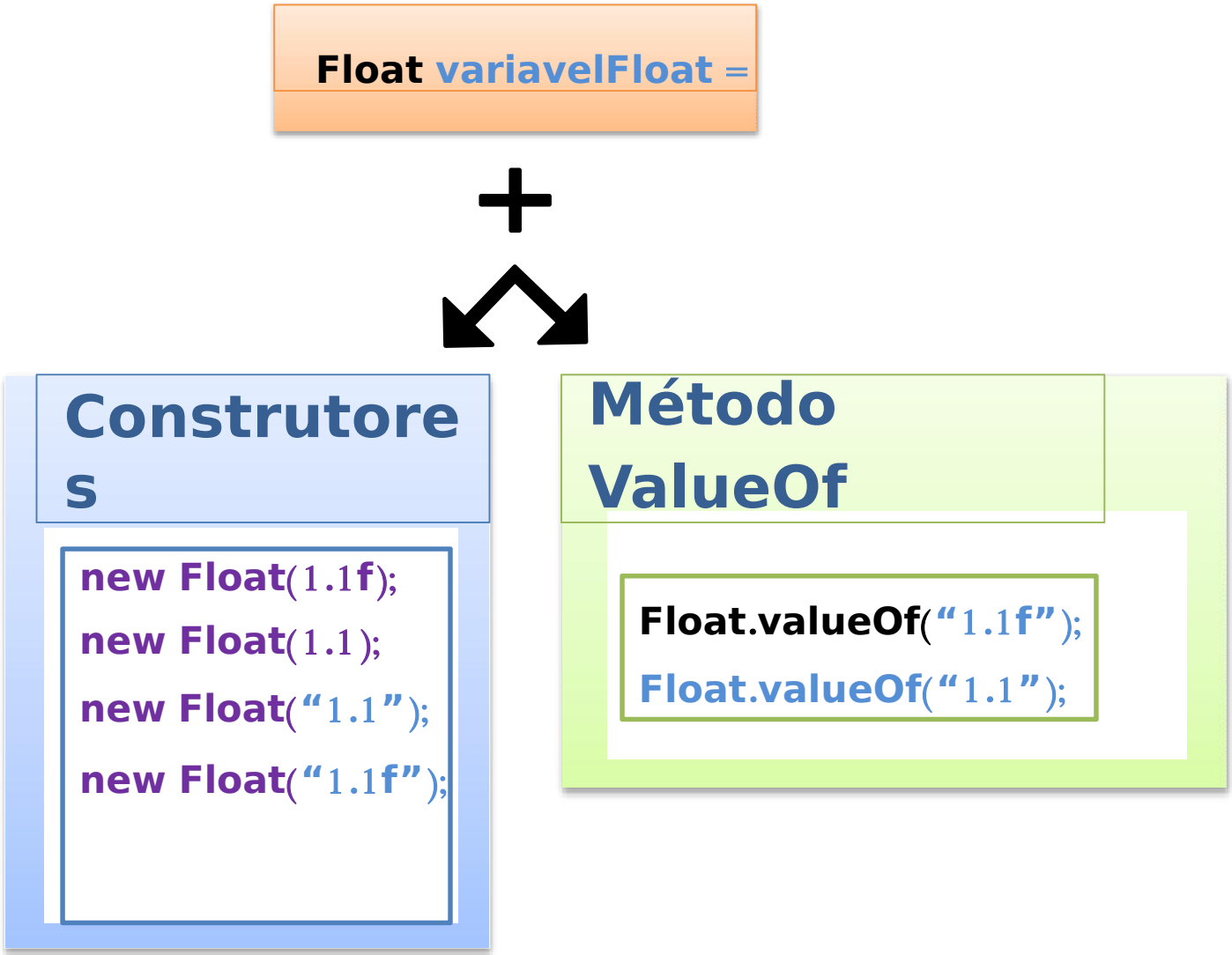
→ **Casting implícito (Automático)**

-----> **Casting explícito (Requer a utilizaçao de cast)**

Classes Wrapper (Empacotadoras)

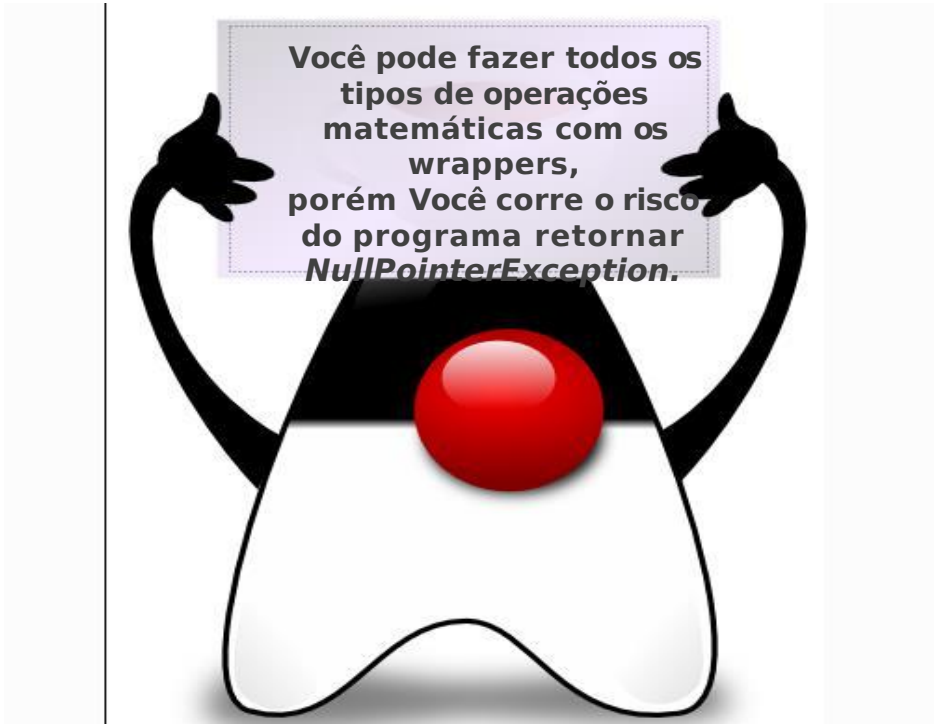


Construtores e método valueOf



AutoBoxing – Boxing and Unboxing

```
int i = 10;
Integer iRef = new Integer(i); // Boxing Explicito
int j = iRef.intValue(); // Unboxing Explicito
iRef = i; // Boxing Automatico
j = iRef; // Unboxing Automatico
```



Java Orientado a Objetos

Operadores

Operadores Aritméticos

*	var1 * var2	Multiplicação
+	var1 + var2	Adição
-	var1 – var2	Subtração
%	var1 % var2	Resto da divisão
/	var1 / var2	Divisão

Operadores Relacionais

!=

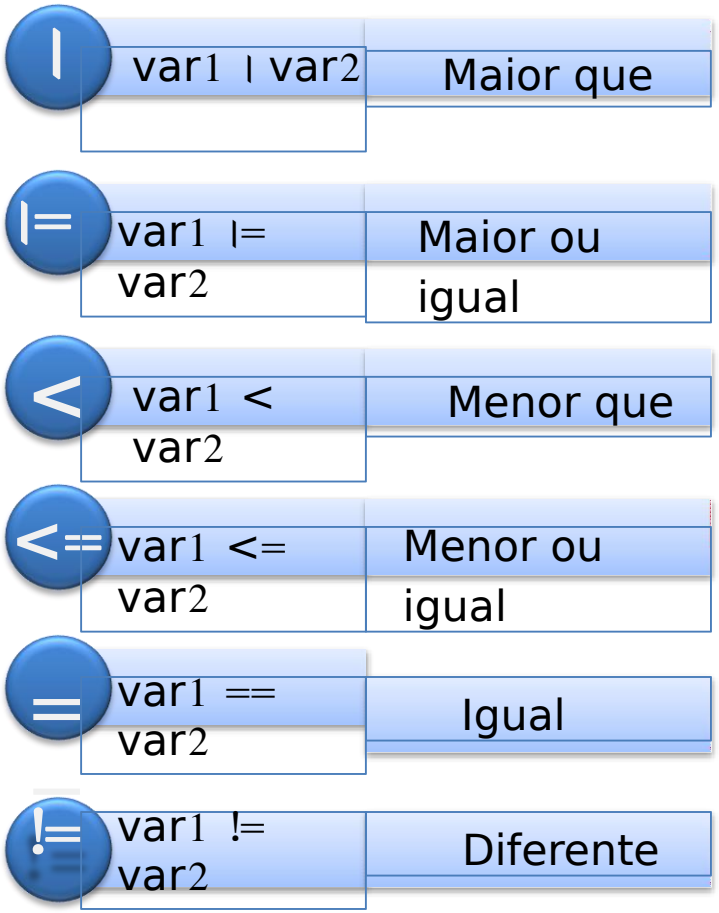
>

==

<

>=

<=



Operadores Lógicos

&&	var1 &&var2	‘E’ lógico (AND)
&	var1 & var2	‘E’ binário
||	var1 || var2	‘OU’ lógico (OR)
|	var1 | var2	‘OU’ binário
^	var1 ^ var2	OU’ exclusivo binário
!	var1 ! var2	Negação (NOT)

&& (e lógico) e & (e binário)

Condição 1	Operador	Condição 2	Resultado
true	&&	true	true
false	&&	true	false
true	&&	false	false
false	&&	false	false

A diferença básica do operador && para & é que o && suporta uma avaliação de curto- circuito (ou avaliação parcial), enquanto que o & não.