

Ambientação com a Sintaxe de Java: parte 1

Prof. Gustavo Wagner

FATEC-PB

Programas em Java

- Texto do programa: arquivo com extensão “.java”
- Arquivo contém ao menos uma **classe**
- Nome do arquivo deve ser IGUAL ao nome da classe principal (pública)

Programas em Java

- Programa Java compilado: arquivo com extensão .class (BYTECODE)
- O arquivo .class executa sendo interpretado pela máquina virtual Java

Programas em Java

Estrutura de um programa Java

```
package test.example;
```

Declaração do pacote

```
import java.awt.Button;
```

```
import java.util.*;
```

Importação de pacotes

```
public class MinhaClasse { ... }
```

**Declaração da classe
(onde o programa será
realmente escrito)**

Programas em Java

- Para programas simples, vamos usar uma **classe-programa**
- Possui um programa principal: **função main**

```
public static void main(String []  
args) {  
    ...  
}
```

- O que tem no main
 - Declaração de variáveis
 - Operações
 - Lógica do programa

Variáveis

- Guarda um valor
 - de um tipo específico
- Declaração deve possuir um nome e um tipo
- Nome: identificador
 - Deve iniciar com letra
- Possui escopo definido
 - Não pode repetir nome no mesmo escopo

The diagram shows a code snippet `int a;` on a yellow background. A blue line points from the word **tipo** to the text `int`. A red line points from the text **identificador** to the text `a;`.

```
int a;
```

tipo **identificador**

Variáveis

- Inicializando variáveis

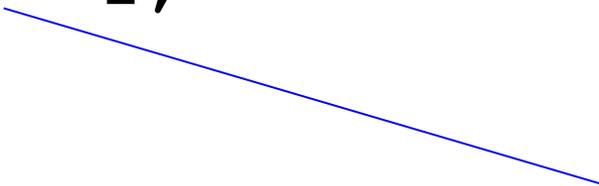
- Java obriga inicialização de variáveis locais!

```
int a = 2;
```

```
int a;
```

```
a = 2;
```

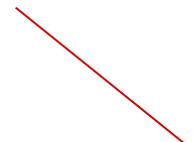
**Comando de
atribuição**



- Constantes

```
final int NÚMERO = 2;
```

**Boa prática:
constantes como
maiúsculas**



Variáveis em Java: dicas

- Java é sensível ao contexto
- Usar nomes com significado
- Boas práticas
 - Nomes de variáveis começam com letra minúscula.
 - `double numero;`
 - Caso o nome seja uma palavra composta, não colocar `_`. A primeira letra da palavra seguinte fica maiúscula
 - `int anoDePublicacao;`

Operadores

- Realizam uma operação sobre 1, 2 ou 3 operandos
- Operador unário: apenas 1 operando
 - Prefixo: operador op
 - Pós-fixado: op operador
- Operador binário: 2 operandos
 - Infixo: op operador op

Operadores aritméticos

- $+$ $-$ $*$ $/$ $\%$
- O operador $/$ é também utilizado para calcular divisões inteiras
 $1/2 = 0$
- O operador $\%$ calcula o resto de uma divisão inteira
 $1\%2 = 1$

Operador de concatenação

- **+** (aplicado a Strings)

```
String nomeCompleto = nome +  
sobrenome;
```

A concatenação também faz uma
conversão implícita para String

```
mensagem = "Este é o cliente número"  
+ x;
```

```
System.out.println("Total: " +  
total);
```

Imprime na tela... 11

Comparação e lógicos

- Operadores de comparação

`>`, `<`, `>=`, `<=`, `==`, `!=`

- Operadores booleanos

- Operações lógicas

`&&` (AND)

`||` (OR)

`!` (NOT)

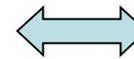
Atribuição

= (atribuição padrão)

+=, -=, *=, /= (atribuições especiais)

```
x =  
0;
```

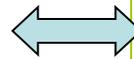
```
x +=  
1;
```



```
x = x + 1;
```

```
a = b = c =  
-1;
```

```
y -= k;
```



```
y = y - k;
```

Operadores Unários

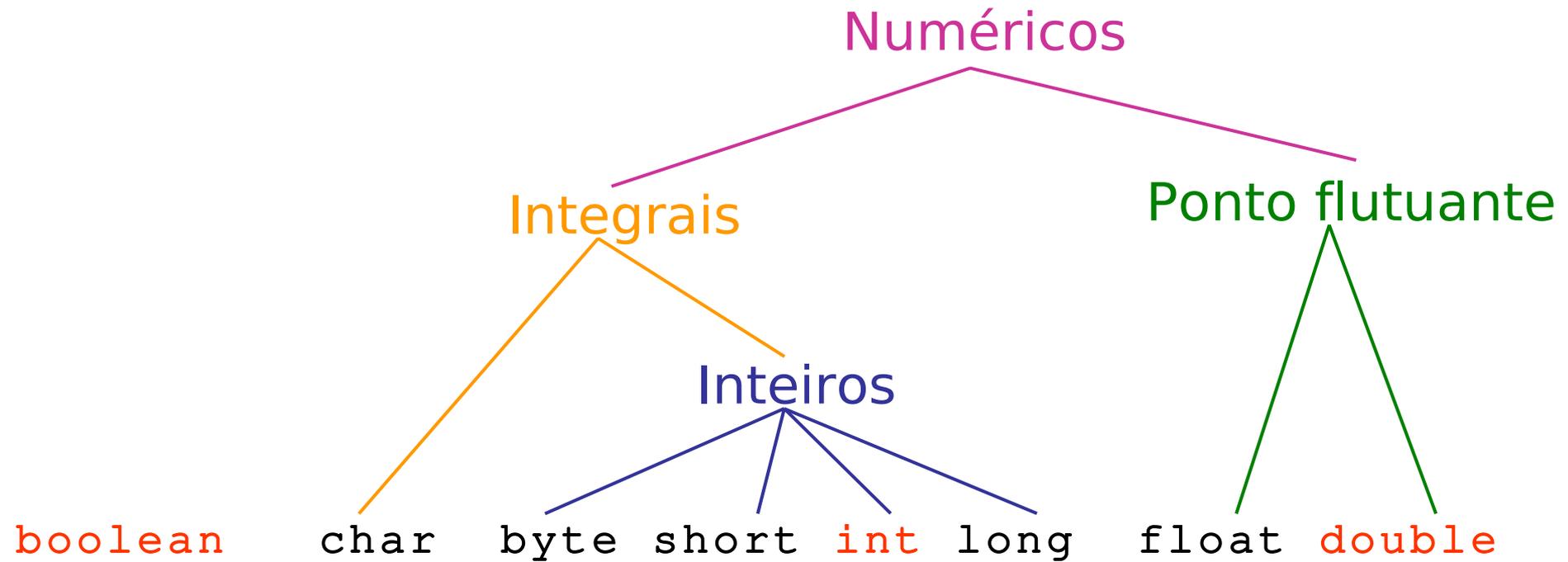
- **++**, **--**
 - **++x** primeiro incrementa o valor de x para a expressão
 - **x++** incrementa o valor de x depois da expressão

`y = 2 + ++x;`

`y = 2 + x++;`

**Qual o valor de x e y
depois de cada atribuição?**

Os tipos primitivos de Java



Tipos de Dados Primitivos

Tipo	Descrição	Tamanho
(inteiros)		
byte	Inteiro de 8 bits	1 byte
short	Inteiro de 16 bits	2 bytes
int	Inteiro de 32 bits	4 bytes
long	Inteiro de 64 bits	8 bytes
(Números Reais)		
float	Ponto flutuante precisão simples	4 bytes IEEE 754
double	Ponto flutuante precisão dupla	8 bytes IEEE 754
(Outros Tipos)		
char	Caracter	16 bits - Unicode
boolean	true ou false	true ou false (1 bit)

Valores dos tipos:

exemplos

Tipo	Exemplo
(inteiros)	
byte	100
short	350
int	2873, 034, 0xF
long	8864L
(Números Reais)	
float	23.746F
double	23.746
(Outros Tipos)	
char	'c', '\u4567'
boolean	true, false

Conversões (cast) de tipos primitivos

Conteúdo não cabe numa
xicara menor (cast explícito)



Conteúdo cabe numa xicara
maior (cast implícito)

Conversões (cast) de tipos primitivos



long (8 bytes)



int (4 bytes)



short (2 bytes)



byte (1 byte)

Inteiros



double (8 bytes)



float (4 bytes)

Ponto flutuante

Cont. Conversões

- Cast

- Conversão de tipos
- Necessário para atribuir um tipo maior a um menor

(<tipo>) <expressão>

- Converte a expressão para o tipo indicado entre parênteses

```
int a = 1234 }  
long b = a; }
```

**conversão
implícita**

```
int c = (int) b; }  
short c = (short) }  
a;
```

**conversão
explícita (cast)**

Promoções em expressões aritméticas

- Tipos de menor precisão são automaticamente convertidos para tipos de maior precisão (**promoção aritmética**)

```
int + short*byte - double
```

```
int + short*short - double
```

```
int + short - double
```

```
int + int - double
```

```
int - double
```

```
double
```

Exercícios

- Criar uma classe em Java com o método main
- Dentro do Main, voce deve declarar variáveis a(int), b(double), c(float), d(char), e e(boolean) e tentar realizar atribuições:
 - a para c (ou seja, c=a)
 - d para b
 - b para a
 - d para e