



# Application Delivery Fundamentals: Java

## Java Excellence

**accenture**>technology

Module 3: Introduction to Java

[illegible]

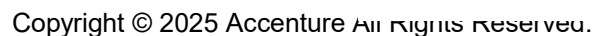
-

-

# Estrutura do programa fonte

**“Estrutura do programa fonte”**

- # Java



- 

Refer to the MainSample.java sample code

214869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820



↓

Hello World

```
variável local:
tipo: String
```

— Ponto-e-vírgula

argomento

o de classe Chamada de

```
println()
```

Copyright © 2025 Accenture All Rights Reserved

served.

# Java Source File Structure

## declaration order

### 1. Package declaration

Usado para organizar uma coleção de classes relacionadas.

### 2. Import statement

Usado para fazer referência a classes declaradas em outros pacotes.

### 3. Class declaration

Um arquivo de origem Java pode ter várias classes, mas apenas uma classe pública é permitida.

```
/*
 * Created on Jun 25, 2008
 *
 * First Java Program
 */
package sef.module3.sample;

import java.lang.*;

/**
 * @author SEF
 */
public class MainSample{

    public static void main(String[] args) {
        // print a message
        System.out.println("Welcome to Java!");
    }

}
```



```

/*
 * Created on Jun 25, 2008
 *
 * First Java Program
 */
package sef.module3.sample;
import java.lang.*;

/**
 * @author SEF
 */
public class MainSample{

    public static void main(String[] args) {
        // print a message
        System.out.println("Welcome to Java!");
    }
}

```



## Class

- ## Class
- A classe é o componente fundamental de todos os programas Java..
  - Todo programa Java inclui pelo menos uma definição de classe pública
  - Uma definição de classe contém todas as variáveis e métodos que fazem o programa funcionar. Isso está contido no corpo da classe indicado pelas chaves de abertura e fechamento.
  - O nome da declaração de classe pública deve ser o mesmo que o nome do arquivo (diferencia maiúsculas de minúsculas).

```

/*
 * Created on Jun 25, 2008
 *
 * First Java Program
 */
package sef.module3.sample;
import java.lang.*;

/**
 * @author SEF
 */
public class MainSample{

    public static void main(String[] args){
        // print a message
        System.out.println("Welcome to Java!");
    }

}

```

# Java Source File Structure

## Braces

- Chaves são usadas para agrupar instruções ou bloco de códigos.
- A chave esquerda ( { ) indica o início de um corpo de classe, que contém variáveis e métodos da classe;
- A chave esquerda também indica o início de um corpo do método;
- Para cada chave esquerda que abre uma classe ou método, você precisa de uma chave direita correspondente ( } ) para fechar a classe ou o método;
- Uma chave direita sempre fecha sua chave esquerda mais próxima;.

```
/*
 * Created on Jun 25, 2008
 *
 * First Java Program
 */
package sef.module3.sample;
import java.lang.*;

/**
 * @author SEF
 */
public class MainSample{

    public static void main(String[] args){
        // print a message
        System.out.println("Welcome to Java!");
    }

}
```

```
Syntax: public static void main( String[] args ) {  
    //Main method implementation goes here  
}
```



Refer to the MainSample.java sample code.

# The 'main( )' Method

## main() method

Esta linha inicia o método main (). Esta é a linha na qual o programa começará a executar;

## String args[]

Declara um parâmetro chamado args, que é uma matriz de String. Representa argumentos da linha de comando;

```
/*
 * Created on Jun 25, 2008
 *
 * Hello World Program
 */
package sef.module3.sample;

/**
 * @author SEF
 */
public class MainSample {

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

## Terminating character

O ponto e vírgula (;) é o caractere de terminação de qualquer instrução java.





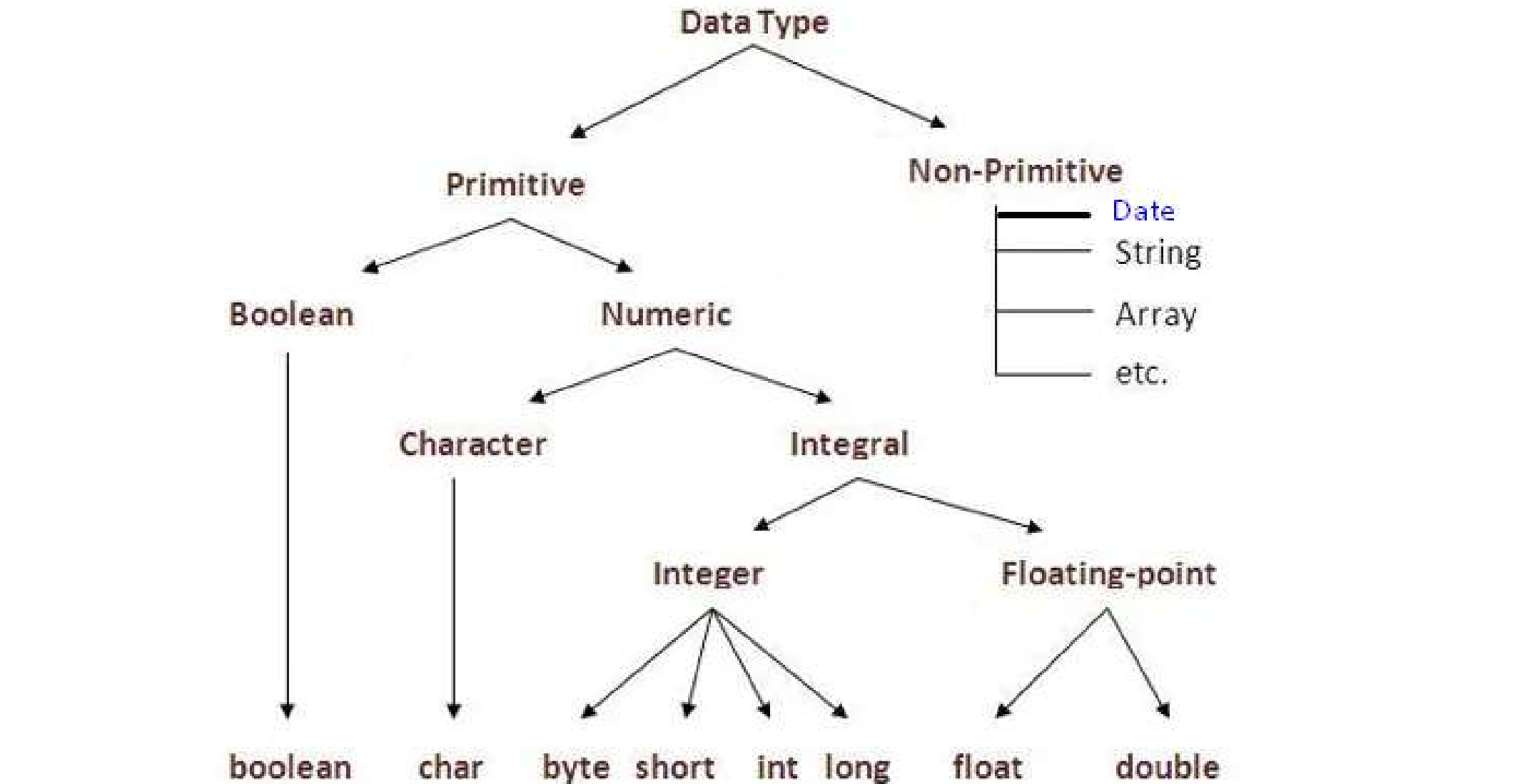


[illegible]

Copyright © 2025 Accenture All Rights Reserved.







# Primitive Data Types

Data Type	Size(bits)	Range			Default
Keyword		Minimum	Maximum		
boolean	1 bit	false	true		false
byte	8 bits	-128	127		0
short	16 bits	-32,768	32,767		0
char	16 bits	'\u0000' (0)	'\uFFFF' (65535)		'\u0000'
int	32 bits	-2,147,483,648	2,147,483,647		0
long	64 bits	$-2^{63}$	$2^{63} - 1$		0
float	32 bits	32-bit IEEE 754 Floating Point			0.0
		$\sim 1.4e^{-045}$	$\sim 3.4e^{+038}$		
double	64 bits	64-bit IEEE 754 Floating Point			0.0
		$\sim 4.9e^{-324}$	$\sim 1.8e^{+308}$		



[illegible]

23

[illegible]

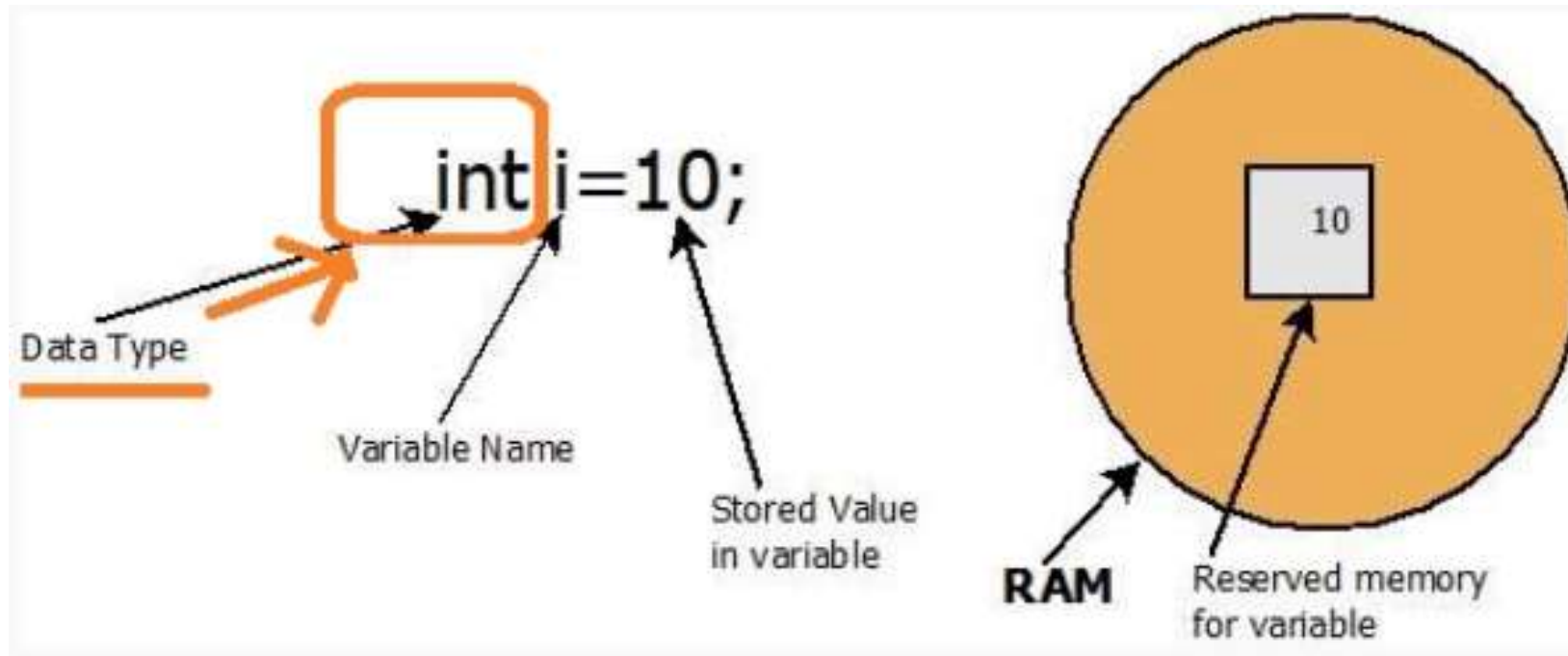
\_\_\_\_\_



[illegible]

- Syntax:** `<data type> <identifier_name>;`
- Examples:**
- ```
int myInteger;  
String myFirstName;  
Date theDateToday;
```

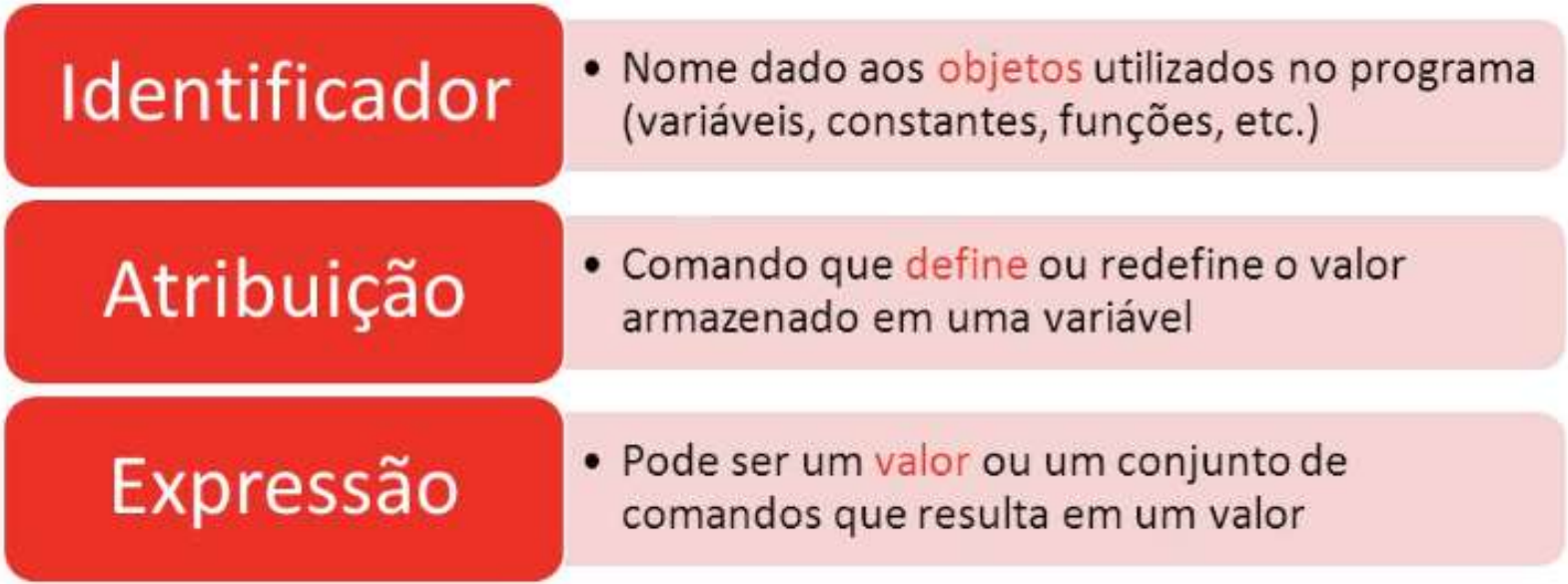
# Variáveis



[illegible]

- Syntax:** `<variable_name> = <the_value>;`  
**Example:** `myInteger = 0;`

- Copyright © 2025 Accenture All Rights Reserved.



**Syntax:** `<identifier_name> = <initial_value>;`  
**Example:** `myInteger = 0;`

**Syntax:**        <identifier\_name> = <initial\_value>;

**Examples:**    myFirstName = "Jason";

                  theDateToday = new Date( );

# Variáveis

- ❑ Não use espaços.
- ❑ Letras maiúsculas e minúsculas são diferentes.
  - ❑ Variáveis **Area** e **area** são distintas.

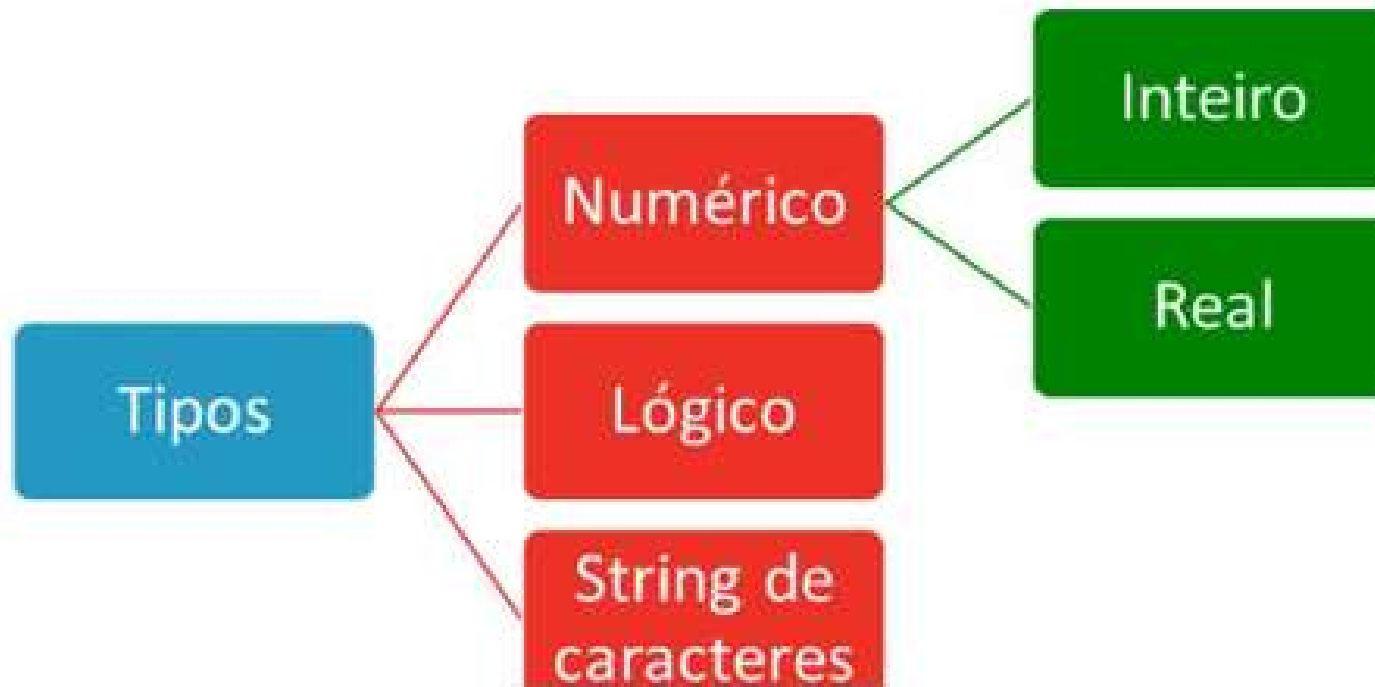
[illegible]

- ```
>>> a = 6
>>> a
6
>>> a = 22
>>> a
22
```

- Copyright © 2025 Accenture All Rights Reserved.

# Variáveis

- ❑ O tipo define a **natureza dos dados** que a variável armazena.
- ❑ Tipos mais comuns





# Variáveis

## Inteiros (`int`)

- São números sem a parte fracionária.
- Exemplos: `1` | `0` | `-5` | `567`

## Reais (`float`)

- São números com parte fracionária.
- Também conhecidos como `ponto flutuante`.
- Exemplos: `1.0` | `3.1415` | `2.7182`

[illegible]

$1 \neq 1.0$

- ❑ Números inteiros e de ponto flutuante são representados de maneiras **distintas** na memória do computador.
- ❑ na maioria das linguagens de programação, utilizamos o **ponto** – e não a vírgula – como separador entre a parte inteira e a parte fracionária de um número ponto flutuante.

[illegible]

- Copyright © 2025 Accenture All Rights Reserved.

214869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820

```
>>> (15/3) == 5
True
```

# Variáveis

- ❑ Uma **string** (= corda) é uma cadeia de caracteres.
- ❑ Uma cadeia de caracteres é um sequência de símbolos, tais como letras, números, sinais de pontuação, etc., que formam textos em geral.

[illegible]

-

[illegible]

- ```
texto = " Amo Java"
```

- Copyright © 2025 Accenture All Rights Reserved.





[illegible]

- 



- `x = 1.3` ❌      `raio = 2.2` ✅

- |                                                     |                                                                                   |                                                                             |                                                                                     |
|-----------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| <pre>raio = 1.3<br/>Raio = 4.6<br/>RAIO = 7.9</pre> |  | <pre>raio_interno = 1.3<br/>raio_meio    = 4.6<br/>raio_externo = 7.9</pre> |  |
|-----------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------|-------------------------------------------------------------------------------------|

- área = 1.3 
area = 2.2 

[illegible]

- $$H = (A^2 + B^2)^{0.5}$$



$$H = ((A^{**2}) + (B^{**2}))^{**0.5}$$



[illegible]

- ```
nivel = 0.8      # nivel de combustivel (m)
altura = 2.3     # altura do tanque (m)
raio = 1.5       # raio da secao vertical (m)
volume = 0       # volume de combustivel (m3)
```

[illegible]

- 
- A woman in a dark business suit is standing and writing on a whiteboard. She is holding a marker in her right hand. Two men in business suits are seated at a glass table in the foreground, looking towards the whiteboard. The man on the left is holding a folder, and the man on the right has his hand to his chin in a thoughtful pose. The office has large windows in the background, letting in bright light.

[illegible]



# Variáveis: Escopo

- Refere-se a partes ou seções de um programa em que a variável tem valor e é considerada 'visible';

## Tipos de variáveis por escopo:

# Class Variables

Compartilhado por todas as instâncias de uma classe. Identificado pela palavra-chave **static**.

# Instance Variables

Pertencer a uma instância de uma classe. São exclusivos para cada instância;

# Local Variables

São acessíveis apenas dentro de sua localidade e geralmente declarados dentro de um método;

## instance method



- ```
1: class VerificaEscopo{
2:     int escopoA;
3:     public void metodo(int escopoB) {
4:         int escopoC;
5:     }
6:
7: }
```

# Variáveis: Escopo

## Class Variable

É uma variável cujo valor é comum a todos os objetos membros da classe.

## Local Variable

Declarados dentro de métodos e / ou sub-rotinas. *aString* é local no método *main*

## Instance Variables

Possui um valor diferente para cada objeto instanciado.

```
package sef.module3.sample;
```

```
public class VariableScope {
```

```
    /**
```

```
     * @param args
```

```
     */
```

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

```
        String aString = "This is a local  
        variable";
```

```
    }
```

```
}
```

```
class Employee {
```

```
    public static int totalCount = 0;
```

```
    private String myFirstName;
```

```
    private String myLastName;
```

```
    private int myAge;
```

```
}
```

O Código abaixo esta correto?

```
package unicesumar.ead.programacao;

public class Pessoa {

    private String nome;
    private int idade;

    private String falar(){

        String frase = "Olá";
        return frase;
    }

    public void andar(){
        System.out.println(nome + frase);
    }

}
```

# Variáveis: Escopo

## Shadowing

Variáveis globais ficam ocultas em trechos do Código que são escopo de outras variáveis com o mesmo nome;

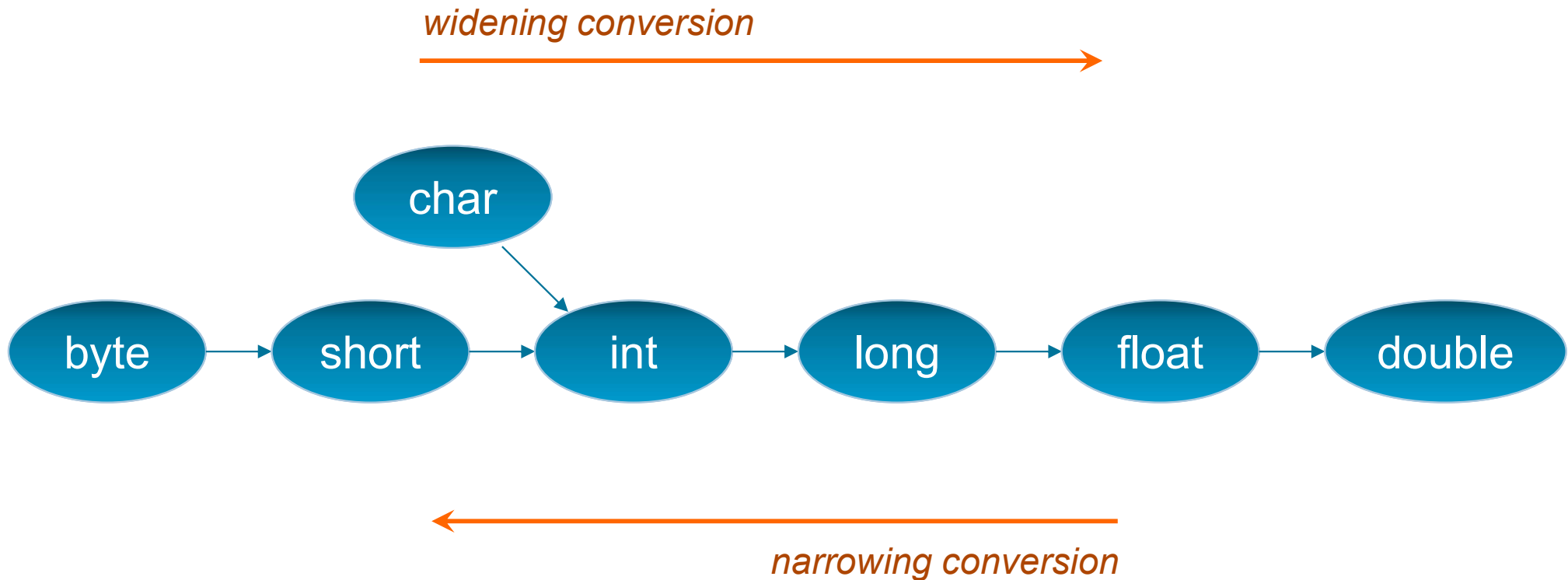
```
public class Parede {  
    private String cor;  
  
    public Parede() {  
        this.cor = "branca";  
    }  
  
    public void colorir(String cor) {  
        cor = "verde";  
    }  
}
```

O que acontece com o atributo de classe, ao executar o método colorir?

[illegible]

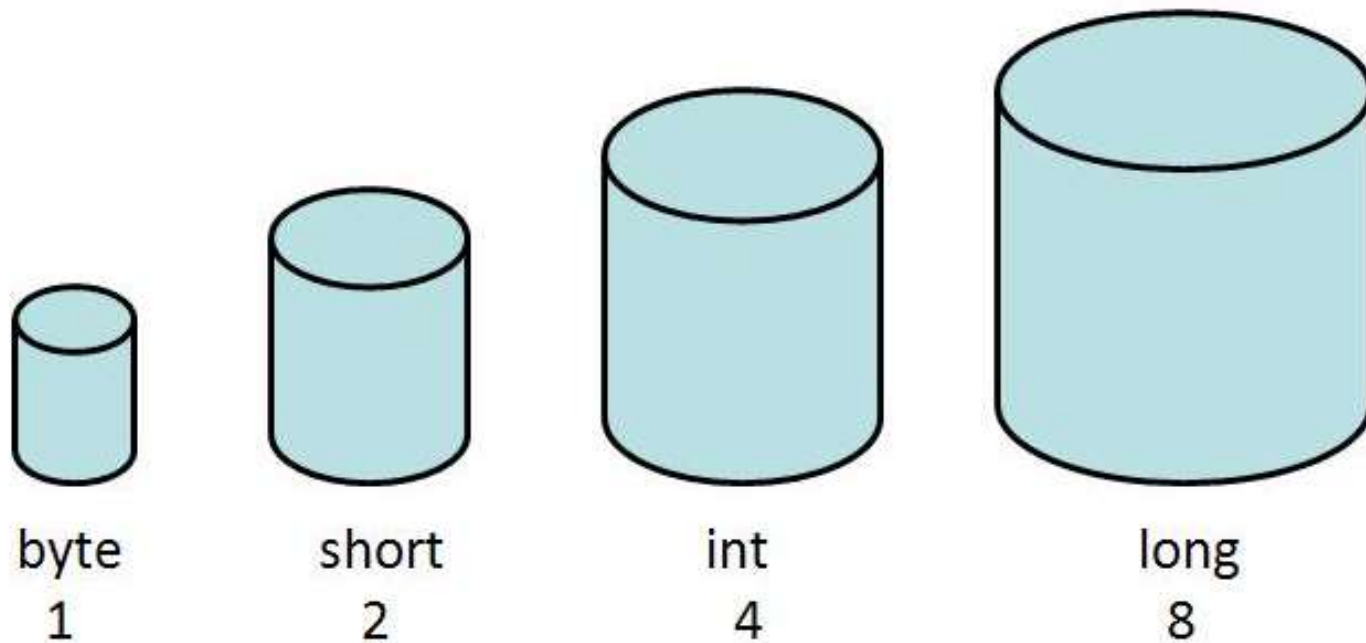


# Primitive Data Type Casting Flow



# Primitive Data Type Casting Flow

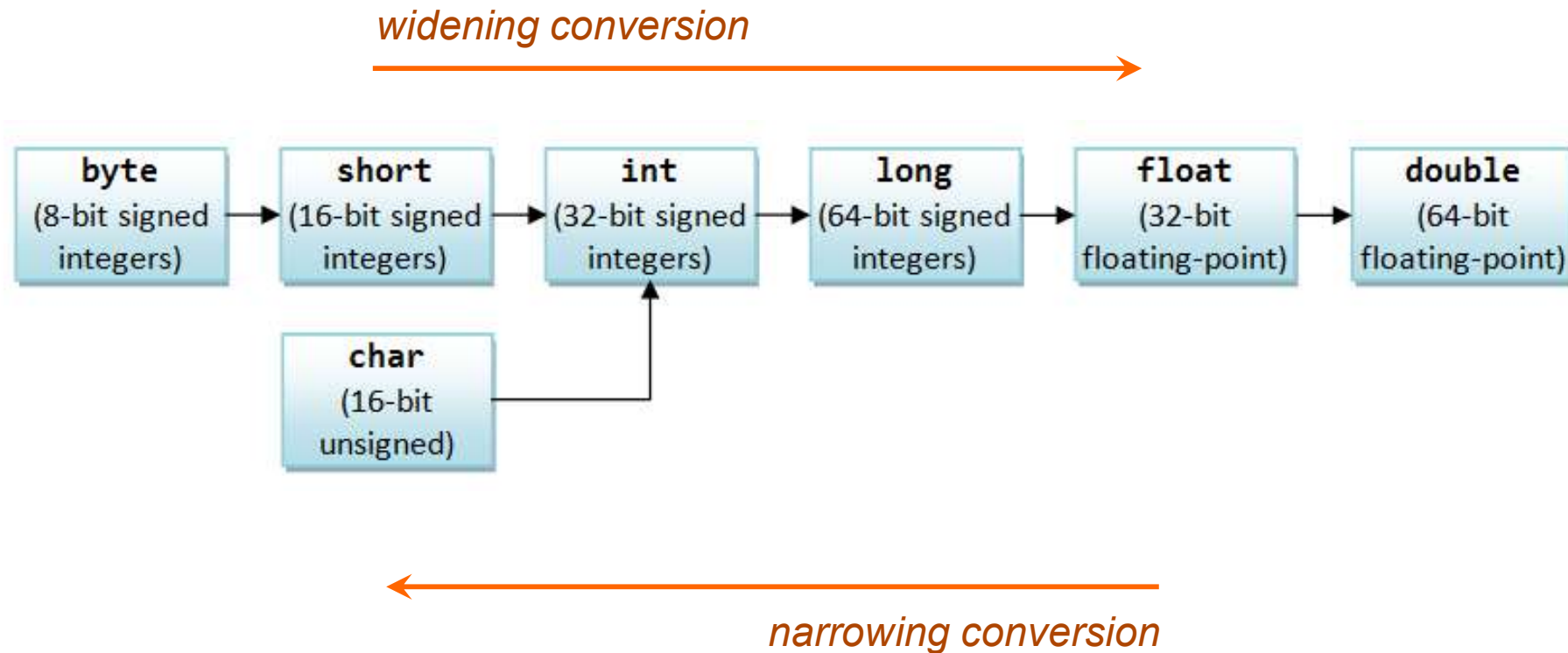
*widening conversion*



*narrowing conversion*



# Primitive Data Type Casting Flow



- Example A:**                    **int a = 1;**  
                                     **double b = a;**
- Example B:**                    **float x = 1.0;**  
                                     **double y = x;**

- Copyright © 2025 Accenture All Rights Reserved.

[illegible]

[illegible]

- ```
double d = 10;
int i;
i = (int) d
```

Type Cast Operator

# Reference Casting

- Refere-se à conversão de um tipo de dados de referência (um objeto) para outro tipo de dados de referência;

[illegible]

Copyright © 2025 Accenture All Rights Reserved. 62

$$A^*(Z+1)$$

- prefix

- postfix:

- ## Example

```
int y = 6;  
y--; // y is 5  
--y; // y is 4
```

Copyright © 2025 Accenture All Rights Reserved.



- | Descrição        | Operador                  |
|------------------|---------------------------|
| igual a          | == (dois sinais de igual) |
| maior que        | >                         |
| menor que        | <                         |
| maior ou igual a | >=                        |
| menor ou igual a | <=                        |
| diferente de     | !=                        |

```
boolean test = x < y;
```





# Java Operators

**NOTE:** This is a hidden slide.  
Please refer to the additional  
content in the notes section.

- This is a hidden slide.

[illegible]

- 
- A woman in a dark business suit is standing and writing on a whiteboard. She is holding a pen in her right hand. Two men in business suits are seated at a glass table in the foreground, looking towards the whiteboard. The man on the left is holding a folder, and the man on the right has his hand to his chin in a thoughtful pose. The office has large windows in the background, letting in bright light.



# Questions and Comments

- What questions or comments do you have?

