

**INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PARAÍBA
Campus João Pessoa**

Programação Orientada a Objetos

Fausto Maranhão Ayres

6 Atributo e Método de classe (estático)

Atributo e Método de classe

- Pertencem a classe e não a instância da classe
- São declarados como **static**
- São acessados junto com o nome da classe

Ex:

```
double area = Math.PI * Math.pow(raio,2) ;
```

↑ ↑
Atributo Método
estático estático

```
int idade = Integer.parseInt(...);
String s = String.format(...);
Collections.sort(...) ;
JOptionPane.showMessageDialog(...);
System.out.println(...);
```

Declaração **static**

■ Atributo static

```
System.out.println()
```

```
public class System{  
    public static PrintStream out;  
    ...  
}
```

■ Método static

```
Integer.parseInt(...)
```

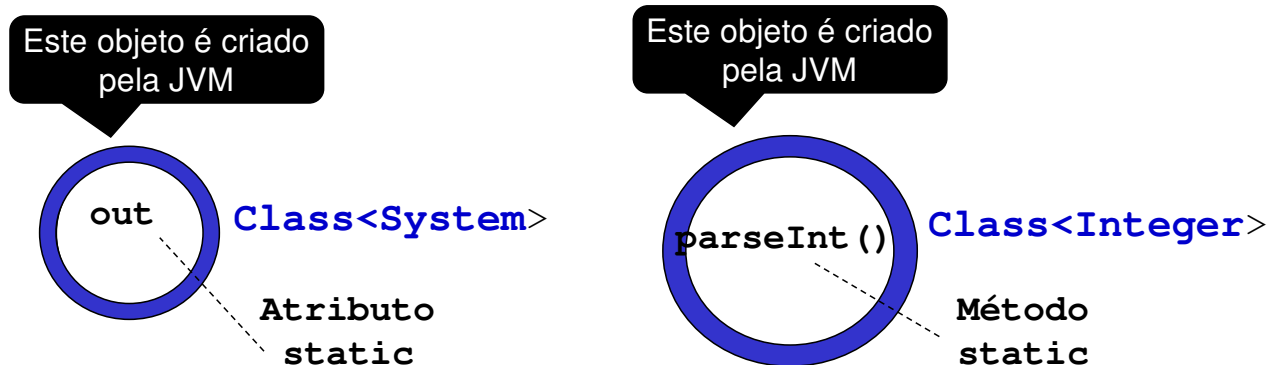
```
public class Integer{  
    public static int parseInt(String valor) {  
        ...  
    }  
}
```

fausto.ayres@ifpb.edu.br

3

Como funciona?

- Para cada classe, a JVM instancia automaticamente um objeto **Class<classe>** reunindo os atributos e métodos **static** da classe



- Para acessa-los basta citar
System.out...
Integer.parseInt(...)

fausto.ayres@ifpb.edu.br

4

Exemplo de Atributo estático

■ Salário mínimo

```
public class Empregado {  
    private int n;  
    public static double minimo = 1000;  
  
    public Empregado(int n) {  
        this.n = n;  
    }  
    public double getSalario() {  
        return n * minimo;  
    }  
}
```

O nome da classe pode ser omitido internamente

fausto.ayres@ifpb.edu.br

5

Teste

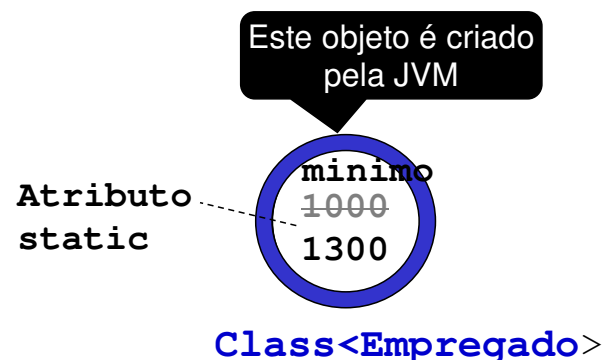
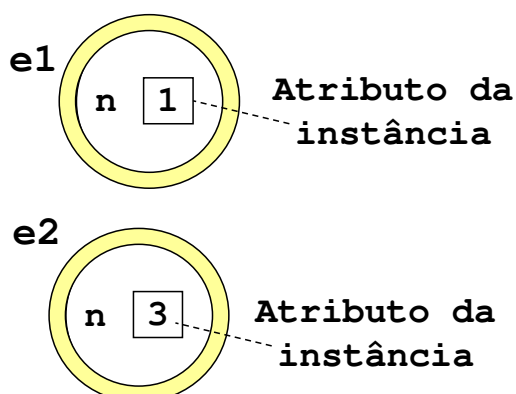
```
Empregado e1= new Empregado(1);           // 1 salario  
Empregado e2= new Empregado(3);           // 3 salários
```

```
System.out.println(e1.getSalario());        //1000  
System.out.println(e2.getSalario());        //3000
```

```
Empregado.minimo = 1300;
```

O nome da classe é obrigatório

```
System.out.println(e1.getSalario());        //1300  
System.out.println(e2.getSalario());        //3900
```

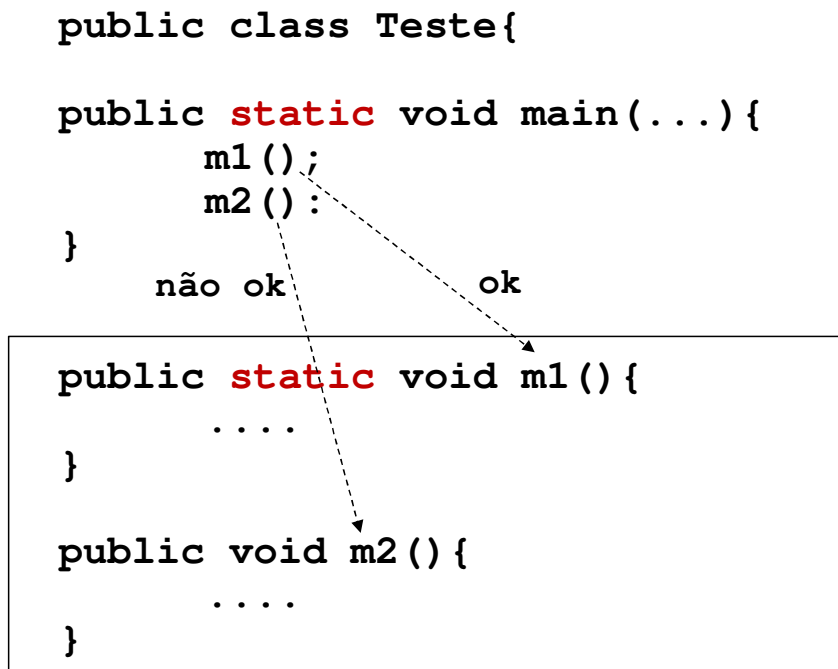


fausto.ayres@ifpb.edu.br

6

Restrição

- **Método estático não pode referenciar** atributos/métodos **não-estáticos** da sua classe



Classes utilitárias

Classes utilitárias

- São classes **não instanciáveis**, contendo atributos e métodos **static**

Exemplo:

```
public class Math {  
    private Math(){}           //private  
  
    public static double sqrt(double a) {...  
    public static double pow(double a, double b) {...  
    public static final double PI = 3.1415;  
    ...  
}
```

fausto.ayres@ifpb.edu.br

9

Exemplo: Util.java

```
public class Util {  
    private Util(){}  
  
    public static boolean validarCPF(String cpf){...}  
    public static boolean validarEmail(String email){...}  
    public static String tirarBranco(String s){...}  
    public static String LINK = "www.ifpb.edu.br";  
    ...  
}
```

Teste:

```
boolean b = Util.validarCPF("88800099911");  
String s = Util.tirarBranco(" frase com brancos ");  
System.out.println(Util.LINK);
```

fausto.ayres@ifpb.edu.br

10

Exemplo: Moeda.java

```
public class Moeda {  
    private Moeda() {}  
    public static double cotacaoDolar = 5.0;  
    public static double cotacaoEuro = 6.0;  
    ...  
    public static double realToDolar(double real) {  
        return real/cotacaoDolar;  
    }  
    public static double dolarToReal(double dolar) {  
        return dolar*cotacaoDolar;  
    }  
}
```

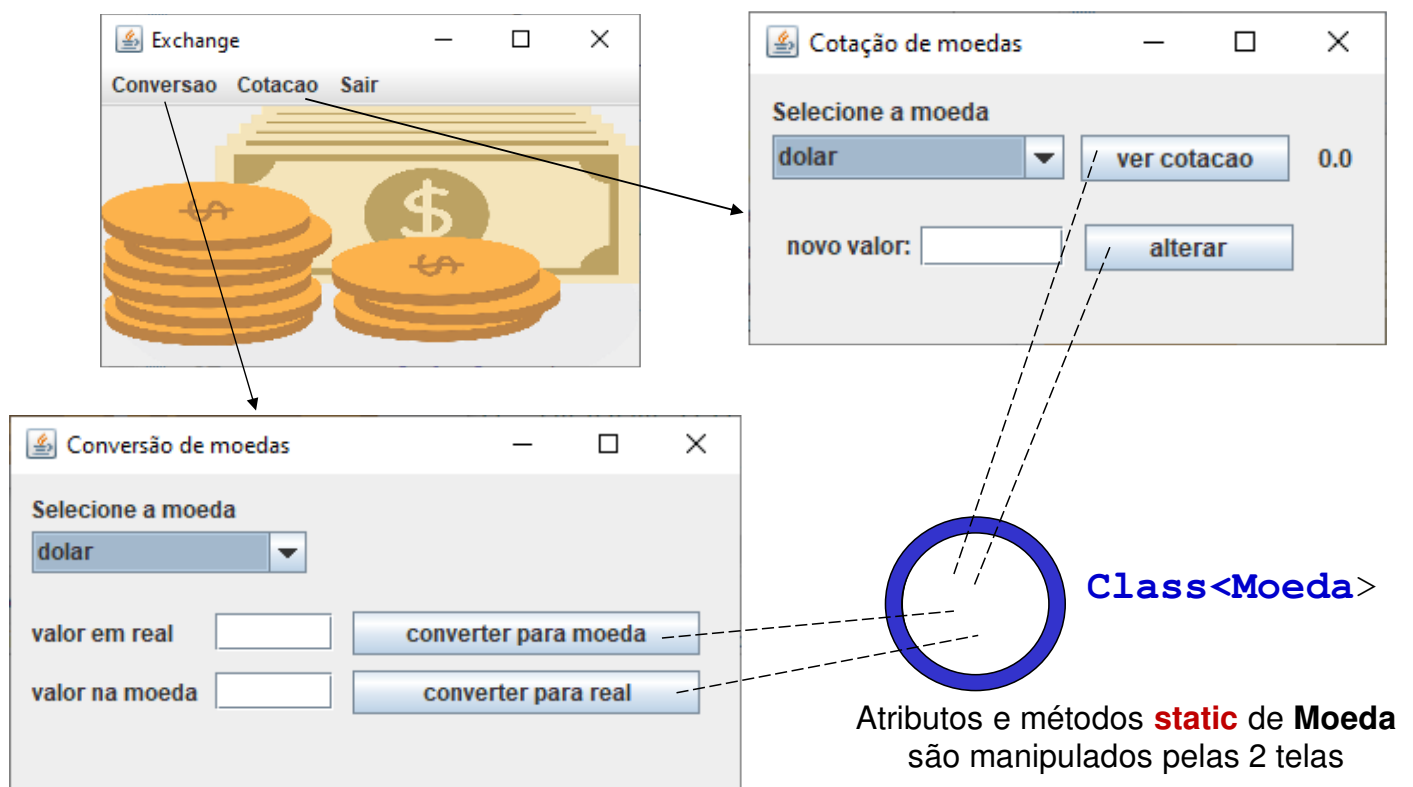
Teste:

```
System.out.println(Moeda.realToDolar(100));    //20.0  
Moeda.cotacaoDolar = 5.1;  
System.out.println(Moeda.realToDolar(100));    //19,6
```

fausto.ayres@ifpb.edu.br

11

Aplicação Gráfica



fausto.ayres@ifpb.edu.br

12