



POO

Método Construtor

Prof. Alcides Calsavara

PUCPR

Conceitos



1. Referência **this**
2. Método construtor

Referência **this**

- ✧ Palavra-chave da linguagem Java
- ✧ Termo usado na implementação de um método para referenciar o objeto da correspondente classe que é o alvo da chamada do método

```
public class Estudante {  
    private int matricula;  
    private String nome;  
    private int ano_de_entrada;  
  
    public void definir_matricula(int m)  
    { matricula = m; }  
  
    public void definir_nome(String n)  
    { this.nome = n; }  
  
    public void definir_ano_de_entrada(int ano_de_entrada)  
    { this.ano_de_entrada = ano_de_entrada; }  
}
```

```
public class Estudante {  
    private int matricula;  
    private String nome;  
    private int ano_de_entrada;
```

```
    public void definir_matricula(int m)  
    { matricula = m; }
```

```
    public void definir_nome(String n)  
    { this.nome = n; }
```

uso opcional (não há ambiguidade)

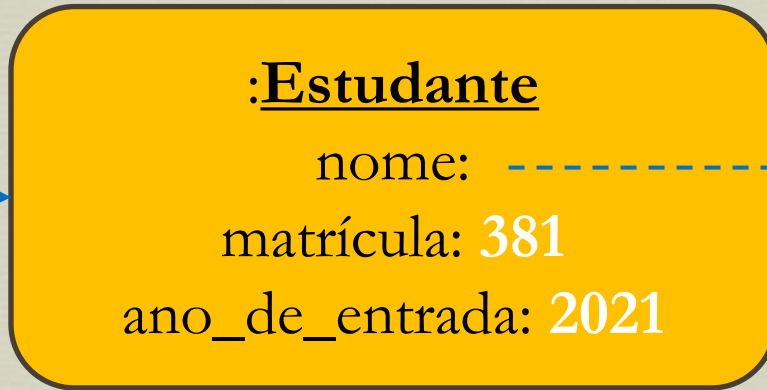
```
    public void definir_ano_de_entrada(int ano_de_entrada)  
    { this.ano_de_entrada = ano_de_entrada; }
```

```
}
```

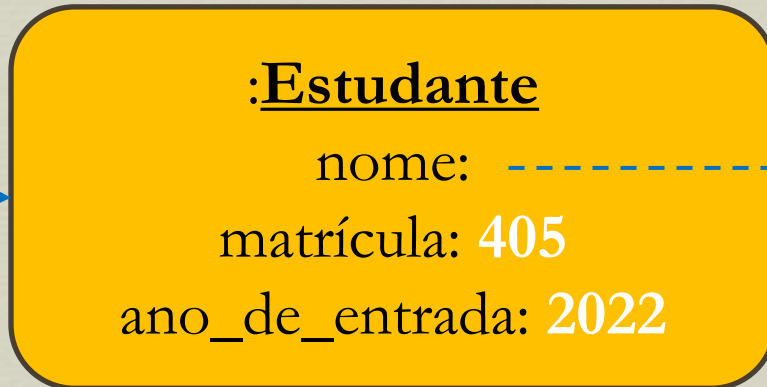
uso obrigatório

```
public class TesteEstudante {  
    public static void main(String[] args)  
    {  
        Estudante a = new Estudante();  
        a.definir_nome( "Candido Portinari");  
        a.definir_matricula( 381 );  
        a.definir_ano_de_entrada( 2021 );  
  
        Estudante b = new Estudante();  
        b.definir_nome( "Di Cavalcanti");  
        b.definir_matricula( 405 );  
        b.definir_ano_de_entrada( 2022 );  
    }  
}
```

a



b



```
public class Imovel {  
    private int codigo;  
    private double area_terreno;  
    private double area_construida;  
    private int bairro;  
  
    public void definir_estado_inicial(int cd, double at, double ac, int br)  
    {  
        codigo = cd;  
        area_terreno = at;  
        area_construida = ac;  
        bairro = br;  
    }  
}
```



```
public class Imovel {  
    private int codigo;  
    private double area_terreno;  
    private double area_construida;  
    private int bairro;  
  
    public void definir_estado_inicial(int cd, double at, double ac, int br)  
    {  
        this.codigo = cd;  
        this.area_terreno = at;  
        this.area_construida = ac;  
        this.bairro = br;  
    }  
}
```

Usando a referência this

uso opcional (não há ambiguidade)

```
public class Imovel {  
    private int codigo;  
    private double area_terreno;  
    private double area_construida;  
    private int bairro;
```

Melhorando os identificadores
dos parâmetros

```
public void definir_estado_inicial(int codigo, double area_terreno,  
    double area_construida, int bairro)  
{  
    this.codigo = codigo;  
    this.area_terreno = area_terreno;  
    this.area_construida = area_construida;  
    this.bairro = bairro;  
}  
}
```

uso obrigatório

Como criar um objeto da classe Imovel?

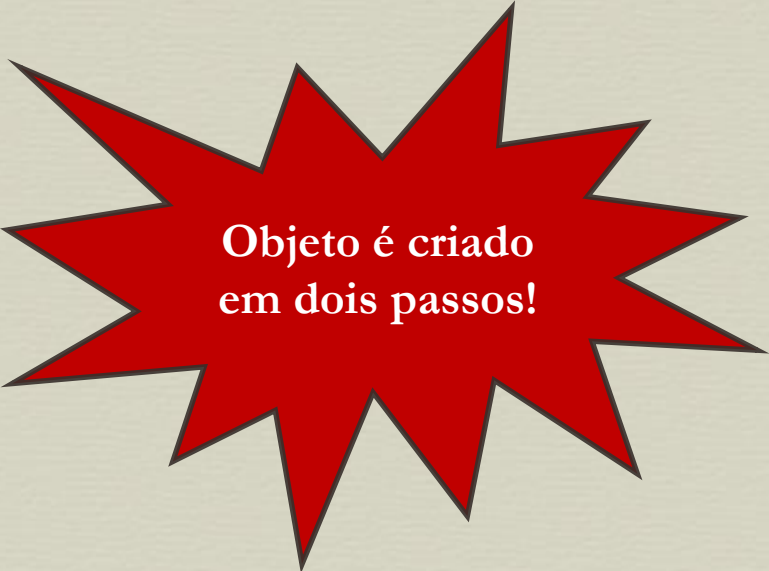
```
Imovel casa = new Imovel( );
```

```
casa.definir_estado_inicial( 34, 450, 200, 4 );
```

Como criar um objeto da classe Imovel?

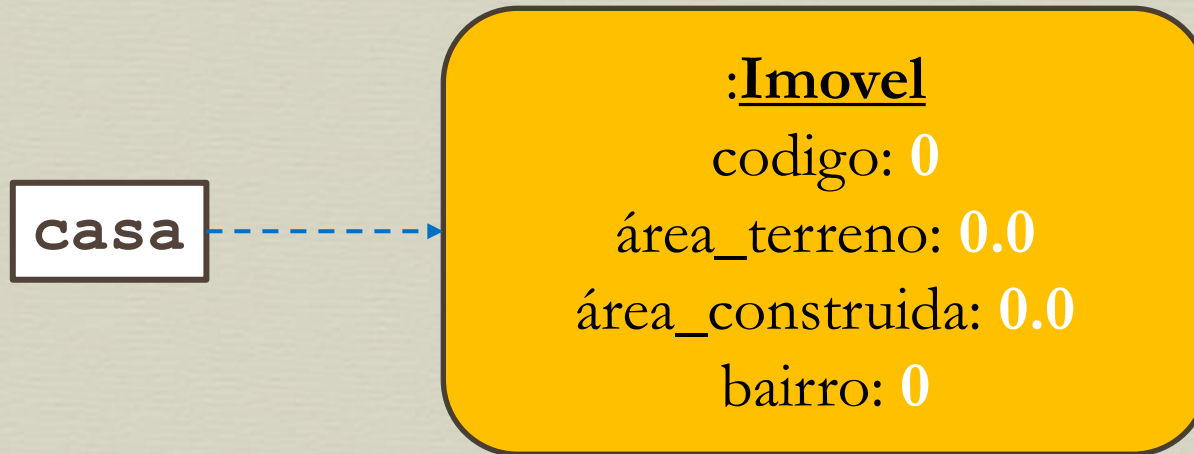
```
Imovel casa = new Imovel( );
```

```
casa.definir_estado_inicial( 34, 450, 200, 4 );
```

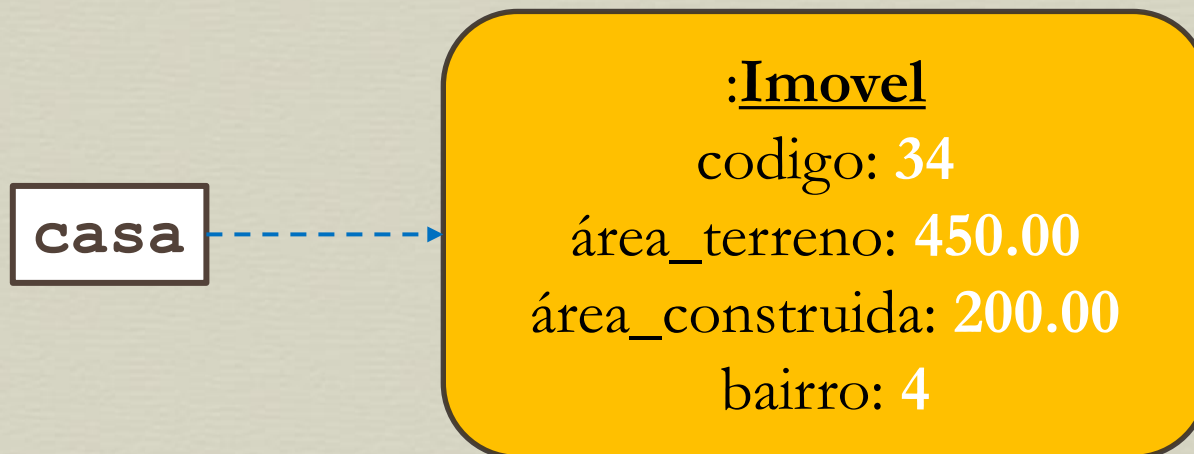


**Objeto é criado
em dois passos!**

```
Imovel casa = new Imovel( );
```



```
casa.definir_estado_inicial( 34, 450, 200, 4 );
```



Método Construtor

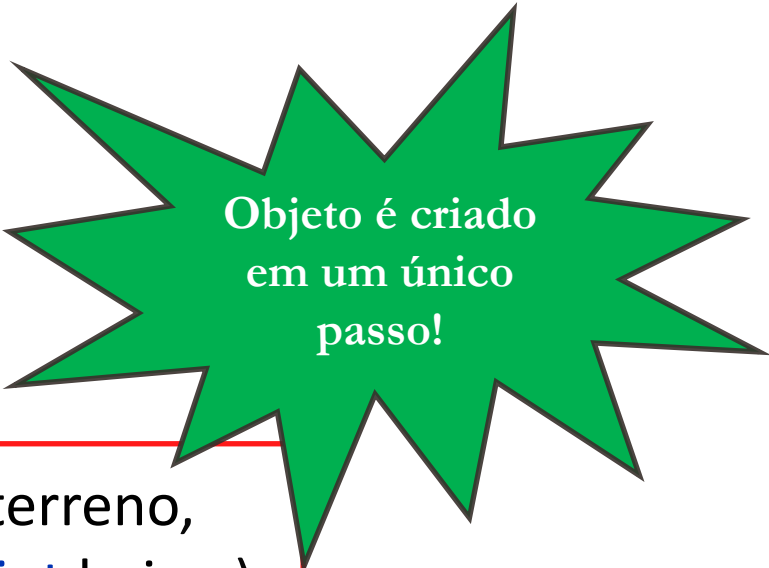
- ✧ Método executado na criação de um objeto (**new**)
- ✧ Em Java, o método construtor é caracterizado da seguinte forma:
 - ✧ possui nome idêntico ao nome da classe
 - ✧ não define retorno
 - ✧ normalmente, tem visibilidade **public**

```
public class Imovel {  
    private int codigo;  
    private double area_terreno;  
    private double area_construida;  
    private int bairro;
```

```
    public Imovel(int codigo, double area_terreno,  
                  double area_construida, int bairro)  
    {  
        this.codigo = codigo;  
        this.area_terreno = area_terreno;  
        this.area_construida = area_construida;  
        this.bairro = bairro;  
    }  
}
```

```
Imovel casa = new Imovel( 34, 450, 200, 4 );
```

```
public class Imovel {  
    private int codigo;  
    private double area_terreno;  
    private double area_construida;  
    private int bairro;
```



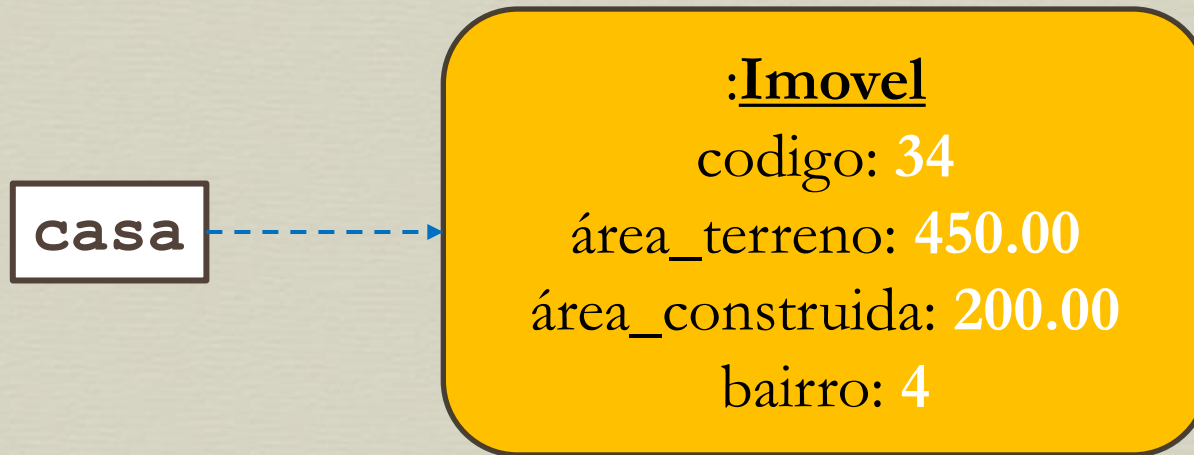
Objeto é criado
em um único
passo!

```
    public Imovel(int codigo, double area_terreno,  
                  double area_construida, int bairro)  
    {  
        this.codigo = codigo;  
        this.area_terreno = area_terreno;  
        this.area_construida = area_construida;  
        this.bairro = bairro;  
    }  
}
```

```
Imovel casa = new Imovel( 34, 450, 200, 4 );
```



```
Imovel casa = new Imovel( 34, 450, 200, 4 );
```



Construtor default



Método executado quando a classe não define um construtor próprio.

```
Imovel casa = new Imovel( );
```

Os atributos do objeto criado ficam com os valores padrão.

O construtor default fica **indisponível** quando uma classe define um construtor próprio.

```
public class Jogador {  
  
    private String nome;  
    private int pontuacao;  
  
    public void definir_nome(String nome)  
    { this.nome = nome; }  
  
    public void aumentar_pontuacao(int ganho)  
    { pontuacao += ganho; }  
}
```

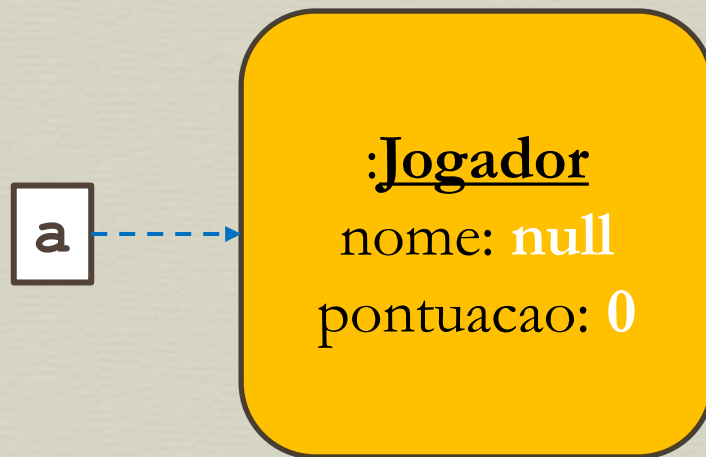
Classe sem método
construtor próprio

```
Jogador a = new Jogador( );
```

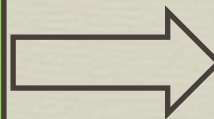
```
a.definir_nome("Edson");  
a.aumentar_pontuacao(10);  
a.aumentar_pontuacao(8);
```

Método construtor
default é executado.

```
Jogador a = new Jogador( );
```

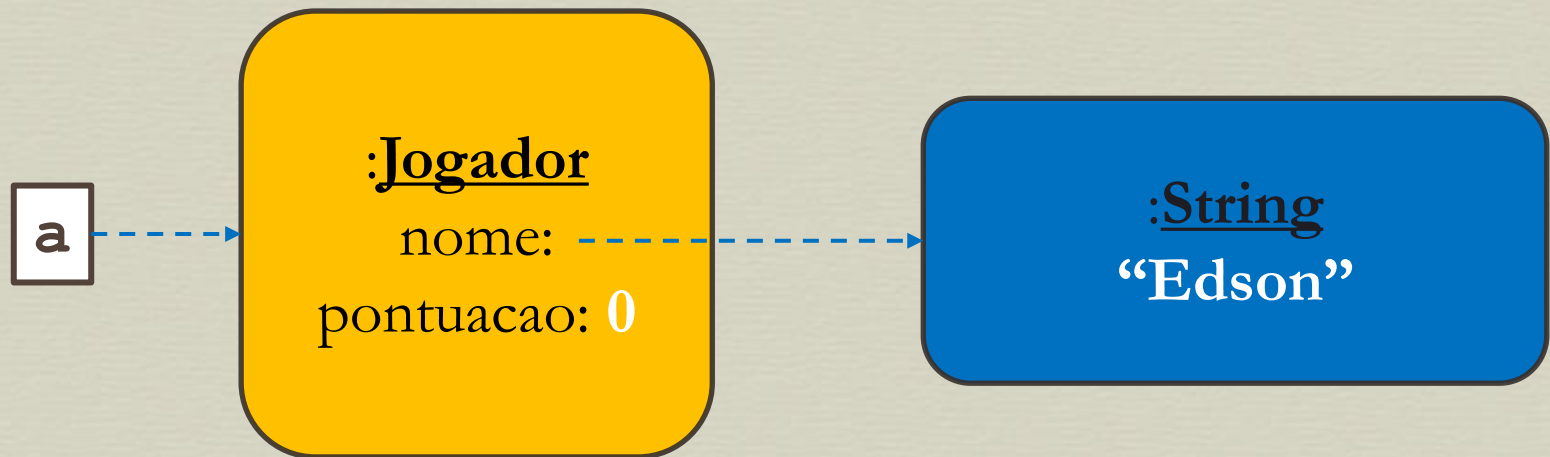


Método construtor default é executado.

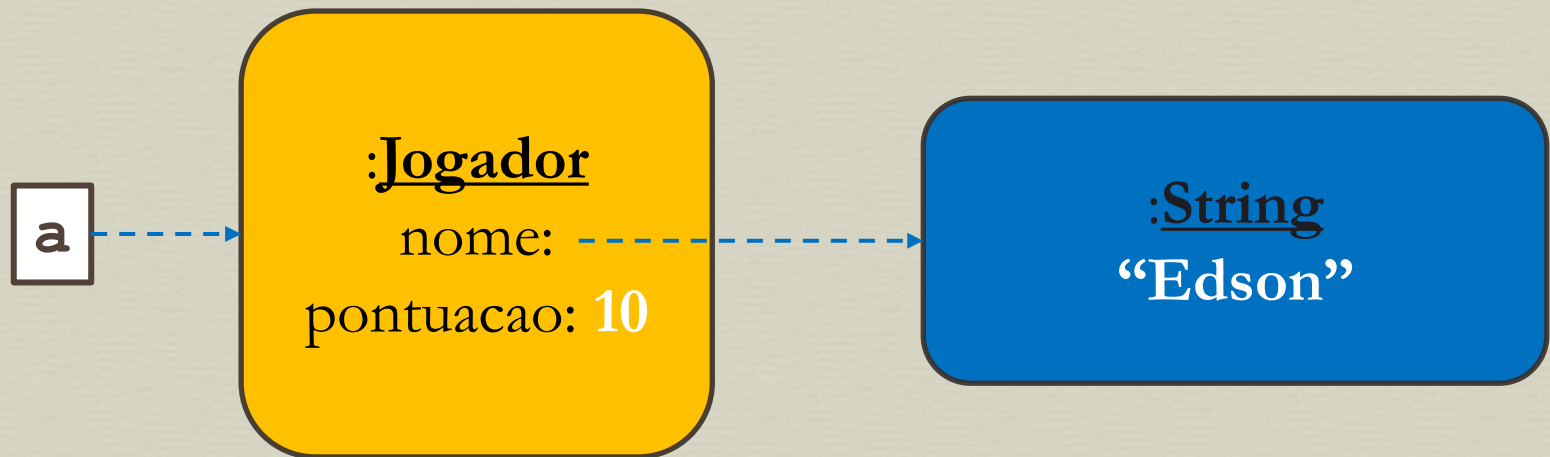


Atributos com valor padrão

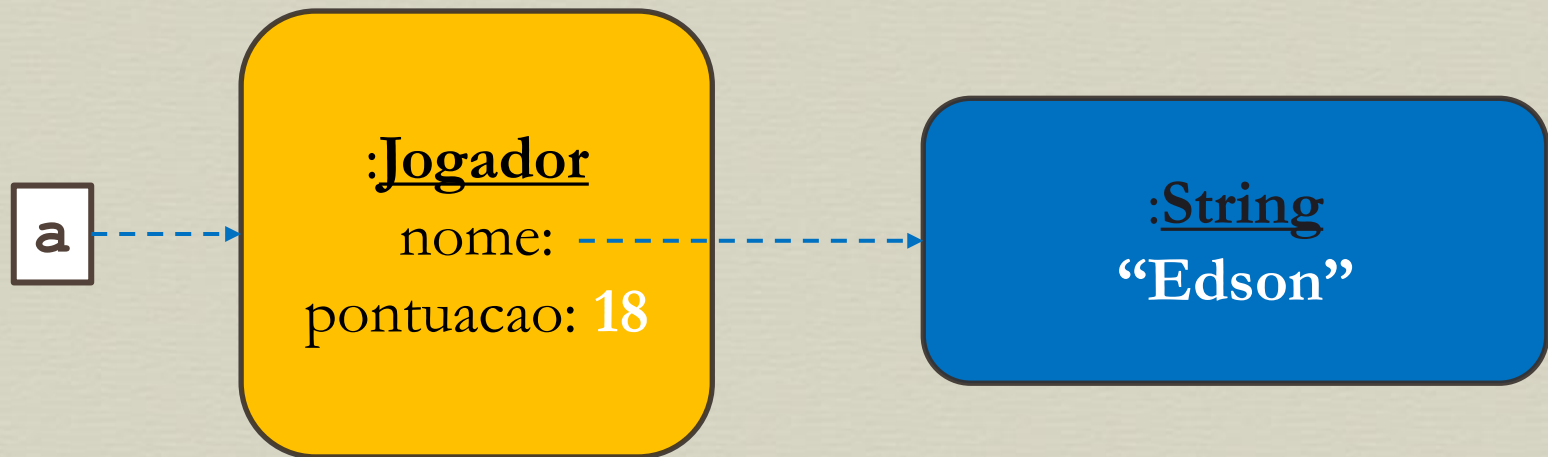
```
a.definir_nome("Edson");
```



```
a.aumentar_pontuacao(10);
```



```
a.aumentar_pontuacao(8);
```




```
public class Produto {  
    private int codigo;  
    private int estoque = 100;  
  
    public void definir_codigo(int codigo)  
    { this.codigo = codigo; }  
  
    public void vender(int quantidade)  
    { estoque -= quantidade; }  
}
```

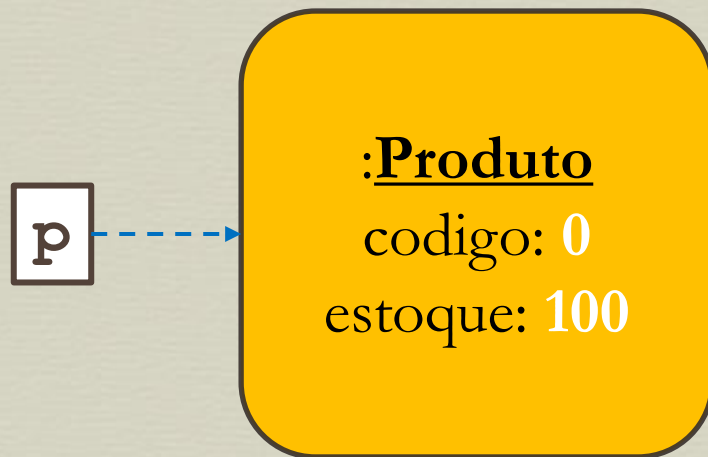
Classe sem método
construtor próprio

```
Produto p = new Produto( );
```

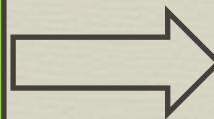
```
p.definir_codigo(7);  
p.vender(20);
```

Método construtor
default é executado.


```
Produto p = new Produto( );
```

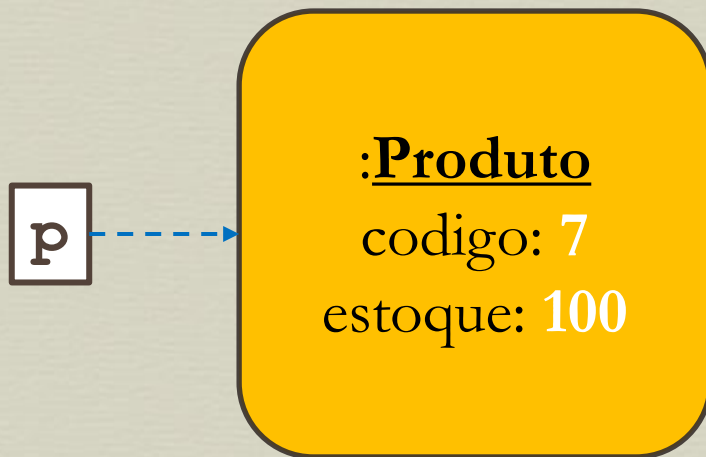


Método construtor
default é executado.



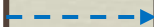
Atributos com
valor padrão

```
p.definir_codigo(7);
```



```
p.vender(20);
```

p



:Produto
codigo: 7
estoque: 80

Múltiplos Construtores

- ✧ Em Java, uma classe pode definir mais de um método construtor.
- ✧ Nesse caso, cada construtor deve possuir uma lista diferente de parâmetros.

```
public class Pessoa {  
    private String nome;  
    private int idade;
```

1

```
public Pessoa(String nome, int idade)  
{ this.nome = nome; this.idade = idade; }
```

2

```
public Pessoa(String nome)  
{ this.nome = nome; idade = -1; }
```

3

```
public Pessoa(int idade)  
{ this.nome = null; this.idade = idade; }
```

4

```
public Pessoa()  
{ nome = null; idade = -1; }
```

```
}
```

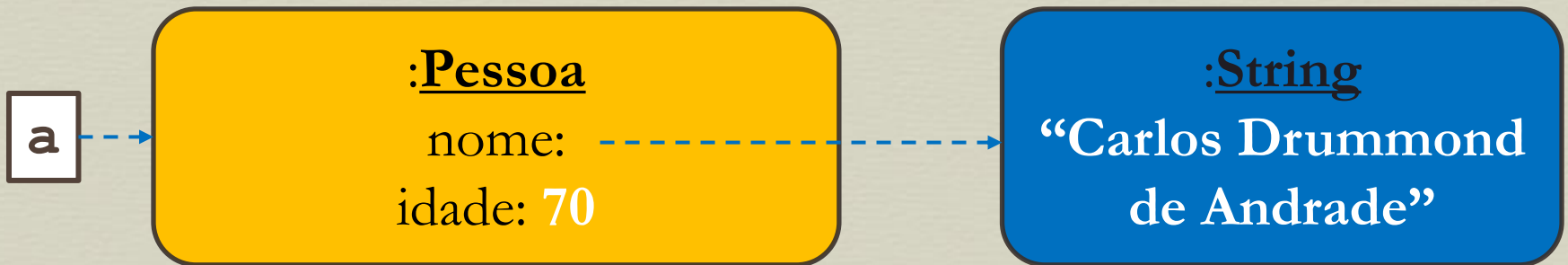
```
public class TestePessoa {  
    public static void main(String[] args)  
    {  
        Pessoa a = new Pessoa("Carlos Drummond de Andrade", 70);  
  
        Pessoa b = new Pessoa("Paulo Freire");  
  
        Pessoa c = new Pessoa(30);  
  
        Pessoa d = new Pessoa( );  
    }  
}
```

Qual construtor é executado em cada caso?

```
Pessoa a = new Pessoa("Carlos Drummond de Andrade", 70);
```

1

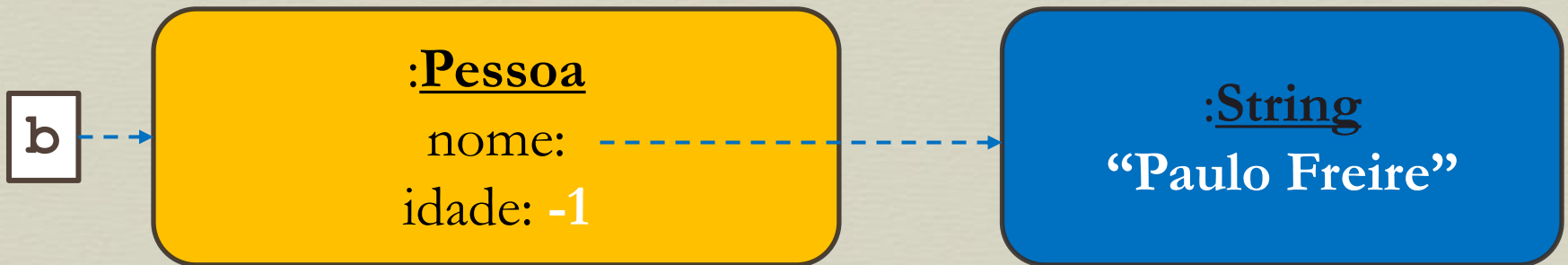
```
public Pessoa(String nome, int idade)  
{ this.nome = nome; this.idade = idade; }
```



```
Pessoa b = new Pessoa("Paulo Freire");
```

2

```
public Pessoa(String nome)  
{ this.nome = nome; this.idade = -1; }
```




```
Pessoa c = new Pessoa(30);
```

3

```
public Pessoa(int idade)  
{this.nome = null; this.idade = idade; }
```

c

:Pessoa
nome: null
idade: 30

```
Pessoa d = new Pessoa( );
```



4

```
public Pessoa( )  
{this.nome = null; this.idade = -1; }
```

d



:Pessoa
nome: null
idade: -1

a

:Pessoa

nome:
idade: 70

:String

“Carlos Drummond
de Andrade”

b

:Pessoa

nome:
idade: -1

:String

“Paulo Freire”

c

:Pessoa

nome: null
idade: 30

d

:Pessoa

nome: null
idade: -1

Cada construtor
permite criar um
objeto de uma
forma diferente

```
public class Arvore {  
  
    private int idade; // em anos  
    private double altura; // em metros  
  
    public Arvore(int idade, double altura)  
    { this.idade = idade; this.altura = altura; }  
  
    public Arvore(int idade) { this.idade = idade; }  
  
    public Arvore(double altura) { this.altura = altura; }  
}
```

```
Arvore a = new Arvore( 6, 2.5 );
```

```
Arvore b = new Arvore( 4 );
```

```
Arvore c = new Arvore( 4.2 );
```

```
Arvore d = new Arvore( 4.0 );
```

```
Arvore a = new Arvore( 6, 2.5 );
```



```
public Arvore(int idade, double altura)  
{ this.idade = idade; this.altura = altura; }
```

```
Arvore a = new Arvore( 4 );
```



```
public Arvore(int idade)  
{ this.idade = idade; }
```



```
Arvore a = new Arvore( 4.2 );
```



```
public Arvore(double altura)  
{ this.altura = altura; }
```



```
Arvore a = new Arvore( 4.0 );
```



```
public Arvore(double altura)  
{ this.altura = altura; }
```