

Algoritmos e Programação II

Slide 03

Prof. Jânio E. Teixeira Júnior





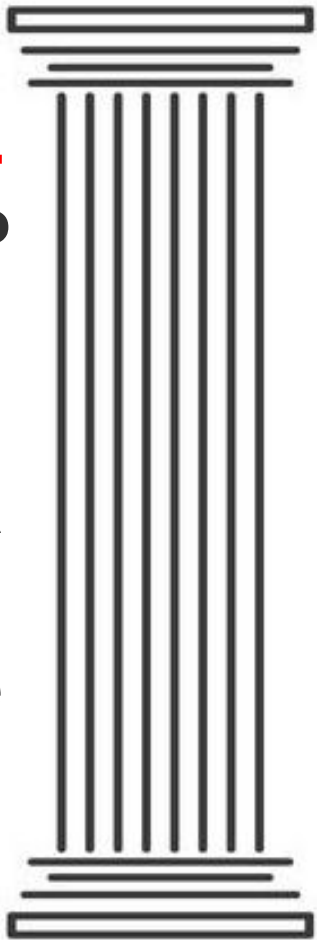
Roteiro

- Revisão – Abstração
- Construtores
 - keyword ***this***
- Destrutores
 - *Garbage Collector*
- Encapsulamento
 - Qualificadores de Acesso
 - Métodos Get e Set
- Trabalho

Orientação a Objetos – Abstração de Dados

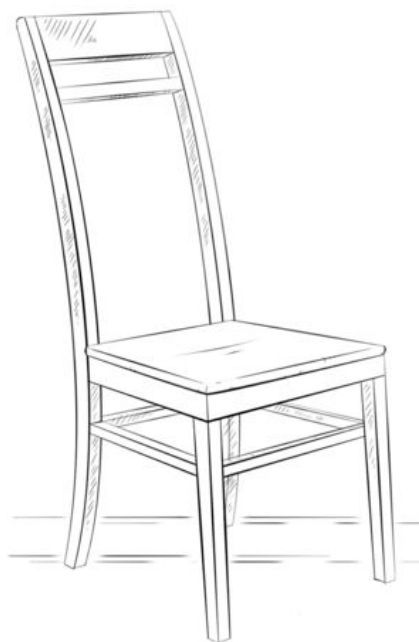
4

P
i
l
a
r
e
s



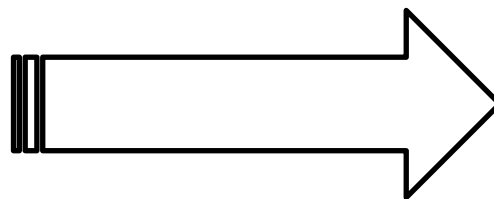
1. **Abstração**
2. Encapsulamento
3. Herança
4. Polimorfismo

Orientação a Objetos – Abstração de Dados



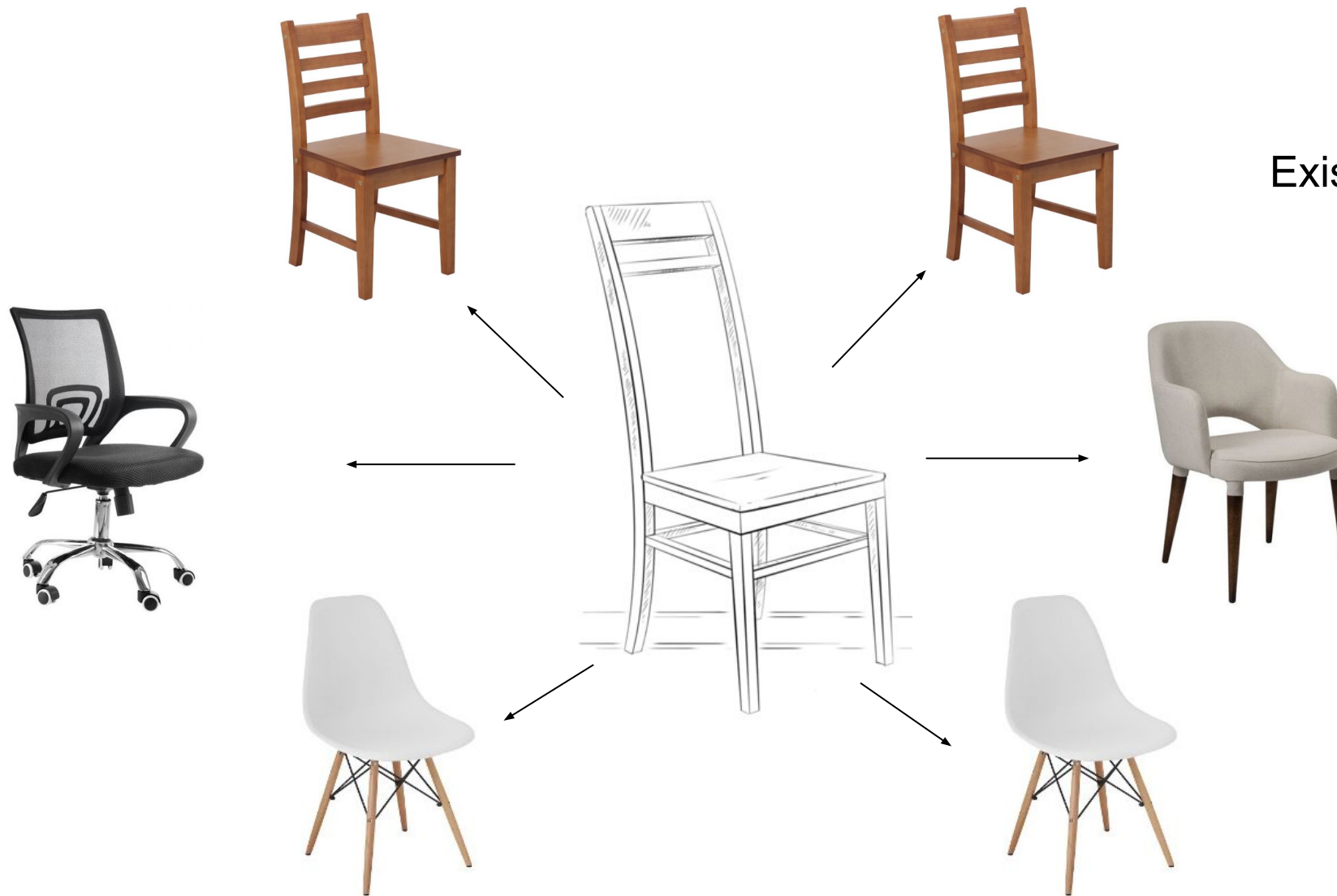
Classe
(Abstrato)

Através de uma classe,
cria/ instancia um objeto



Objeto
(Concreto)

Orientação a Objetos – Abstração de Dados





Orientação a Objetos – UML

- Classe: Cadeira
- Atributos (características):
 - Nome, qtdPes, cor, preco.
- Métodos (ações):
 - possuiEstoque;
 - obterPreco;
 - obterCor.

Cadeira
+ nome: String + qtdPes: Integer + cor: String + preco: Double
+ possuiEstoque(): Boolean + obterPreco(): Double + obterCor(): String



Construtores



Construtores

- Definição: “Alocar o espaço de memória para receber os dados conforme a estrutura definida pela classe”.
- Os construtores são um tipo especial de métodos;
- Geralmente um construtor é acionado através da palavra-chave **new**;
- Para implementar um construtor deve-se utilizar o mesmo nome da classe que ele pertence;
- Toda classe possui ao menos um construtor;
 - Caso não exista um construtor descrito na classe, essa classe recebe um construtor *default*.
- Ex.: Carro **ferrari** = **new** Carro();



Construtores – Criando um construtor

```
public Carro() { // construtor padrão
    super(); // chamando o construtor da classe Object
}
public Carro(String nome1, double velocidade1, double potencia1,
    double peso1) {
    super(); // chamando o construtor da classe Object
    nome = nome1;
    velocidade = velocidade1;
    potencia = potencia1;
    peso = peso1;
}
```



Keyword *this*

- Em Java, a palavra-chave (**keyword**) *this* é uma referência ao objeto atual cujo o método está sendo executado;
- Através dessa **keyword** podemos acessar também seus atributos e métodos;
- Por exemplo, em um construtor os seus parâmetros podem possuir o mesmo nome que os atributos da classe, dessa forma, a utilização do **keyword this** seria necessário. Segue o exemplo:

```
public Carro(String nome, double velocidade, double potencia) {  
    this.nome = nome;  
    this.velocidade = velocidade;  
    this.potencia = potencia;  
}
```



Destruutores



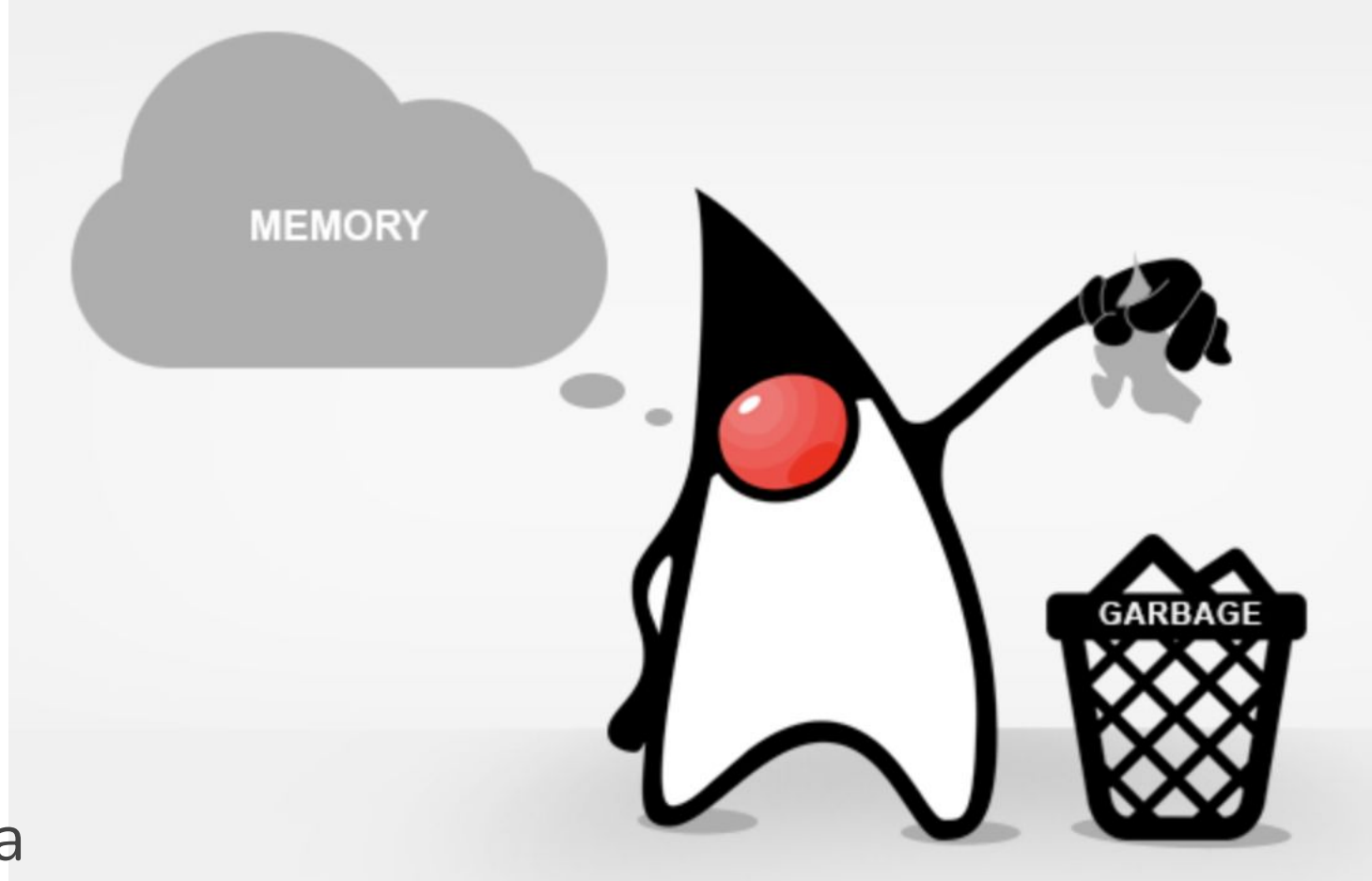
Destrutores

- É o método especial responsável por liberar o espaço de memória de um objeto. Ao executá-lo, toda operação em andamento é finalizada, por exemplo: fechar arquivos, encerrar a comunicação e liberar os recursos alocados do sistema. Eliminando qualquer vestígio da instância do objeto;
- **Como funciona do Java?**
- **// Deprecated no java 9**
- **protected void** finalize() {
 // codigo do destrutor
}



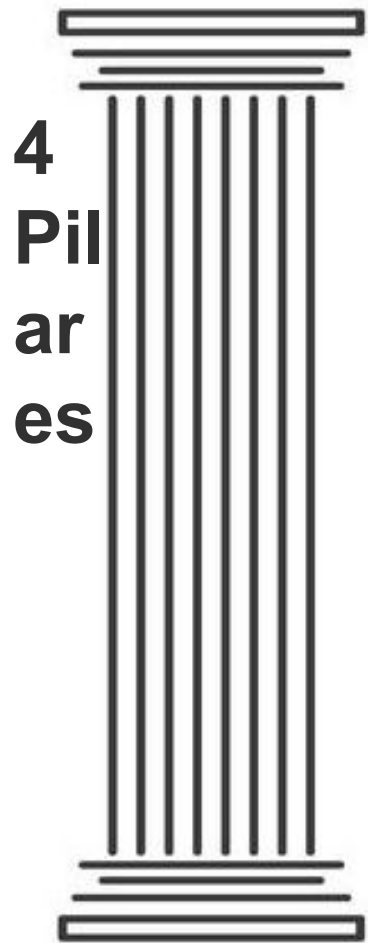
Garbage Collector

- Coletor de lixo;
- Liberação automática da memória;
- Responsável por eliminar da memória todos os objetos que não estão sendo utilizados;
- Boas práticas para a liberação de memória:
 - Setar o valor ***null*** aos objetos que não estão mais sendo utilizados.





Orientação a Objetos – Encapsulamento



1. Abstração
2. Encapsulamento
3. Herança
4. Polimorfismo



Encapsulamento

“Tem certas coisas na vida que você não precisa saber como funciona, você simplesmente a utiliza”.

Exemplo:





Encapsulamento

- **Encapsulamento** é a técnica que faz com que detalhes internos do funcionamento de uma classe permaneçam **ocultos** para determinados objetos que tenham acesso a ela;
- Podemos restringir o acesso na linguagem Java através dos **qualificadores de acesso** e **interfaces**.



Qualificadores de Acesso ou Modificadores de Acesso

- **private** (privado) – Apenas a própria classe tem acesso;
- **protected** (protegido) – Acessível para subclasses e classes do mesmo pacote;
- **public** (público) – Todas as classes tem acesso;
- **default** (padrão) – Acessível para todas as classes do mesmo pacote.



Métodos de acesso - Get e Set

- Padrão da linguagem Java para obter e modificar um atributo;

```
public class Carro {  
    private double velocidade;
```

```
    public double getVelocidade() {  
        return velocidade;  
    }
```

```
    public void setVelocidade(double velocidade) {  
        this.velocidade = velocidade;  
    }  
}
```



Exemplo UML dos qualificadores

- Legenda:
 - - private;
 - # protected;
 - + public;
 - ~ default.

Carro
<ul style="list-style-type: none">- nome : String# marca : String+ potencia : double~ velocidade : double- peso : double
<ul style="list-style-type: none">+ acelerar() : void+ frear() : void+ imprimir() : void

Trabalho



- Faça a abstração das informações de uma **máquina de refrigerante**;
- Modele a classe **MaquinaRefrigerante** (e outras, caso existam) em UML;
- Construa um programa em Java que simule a execução de uma máquina de refrigerante;
- A máquina de refrigerante deve, **por exemplo**:
 - realizar uma venda;
 - selecionar um refrigerante;
 - inserir o dinheiro;
 - devolver o troco.
- Aplique a técnica de encapsulamento em seus modificadores de acesso;
- Utilize construtores (Default não vale);
- Implemente também os métodos get e set;
- O programa deve ter um menu de navegação.



Algoritmos e Programação II

Slide 03

Obrigado.

