



# Java Excellence

## Application Delivery Fundamentals: Java

**accenture**<sup>></sup>technology

### Module 4: Classes e Objetos – Parte 1

# Introdução

## “Programação Orientada a Objetos”

# Orientação a objetos

A POO é uma técnica que organiza o código em torno de objetos, que encapsulam tanto dados (atributos) quanto os comportamentos (métodos) que operam nesses dados.

**Programação orientada a objetos é mais do que apenas uma técnica de codificação.**

**É uma maneira de pensar e estruturar soluções de software que promove a clareza, a eficiência e a adaptabilidade, tornando-se um pilar fundamental da engenharia de software moderna.**

# Orientação a objetos

## Programação Orientada a Objetos



# Introdução

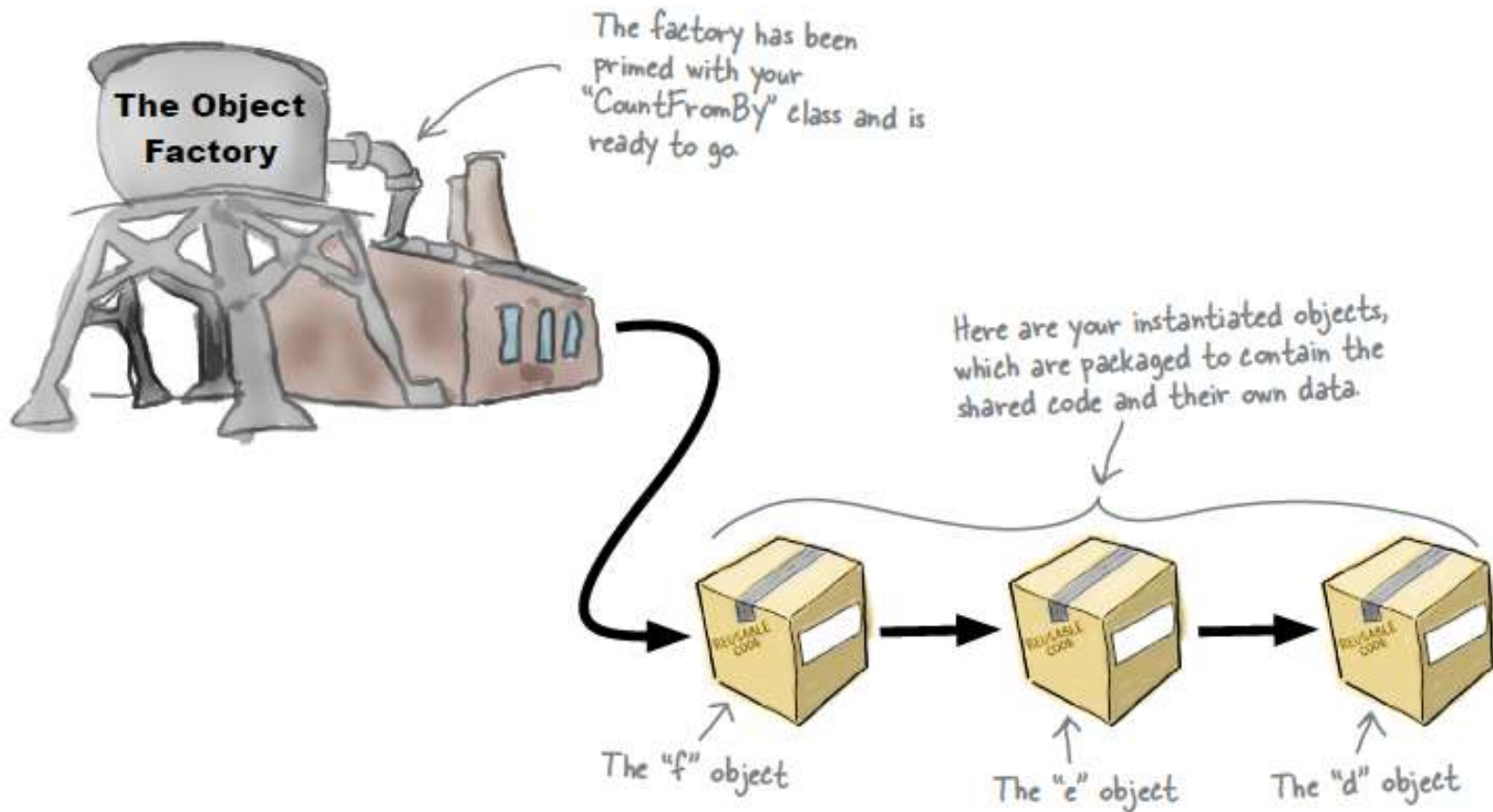
**“Classes and Objects”**

# Classes vs. Instances

- **CLASSE:**
  - Uma classe é um **modelo** para **criar** um objeto.
  - Uma classe tem **atributos** e **métodos**.
- 
- **OBJETO:**
  - Objeto é a **materialização** de uma classe.
  - Todos os objetos que pertencem a uma classe são **instâncias** dessa **classe**.

[illegible]

# Classes vs. Instances





# Visão geral da programação orientada a objetos

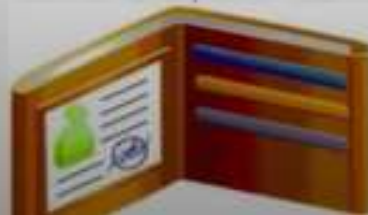
- A Programação Orientada a Objetos é um padrão de programação que utiliza objetos e suas interações para projetar e implementar aplicativos;
- Objetos são entidades que servem como os blocos de construção básicos de um aplicativo orientado à objetos;

# Overview of Object-Oriented Programming

## Objeto:

### The Customer Properties and Behaviors

- Properties
  - Name
  - Address
  - Age
  - Order number
  - Customer number
- Behaviors
  - Shop
  - Set address
  - Add item to cart
  - Ask for a discount
  - Display customer details



# Atributos e Comportamento

- Variáveis declaradas como parte da classe representam os dados contidos pelas instâncias dessa classe;
- Os valores armazenados por essas variáveis representam os 'atributos' ou 'estados' atuais do objeto
- Os métodos declarados como parte da classe representam as operações que as instâncias da classe podem executar;
- Diz-se que essas operações são o 'comportamento' do objeto

# Informações dentro de um objeto

Cada objeto define três tipos básicos de informações:

## Identity

- Um objeto deve descrever os recursos que permitirão a seus usuários diferenciá-lo de outros objetos. Precisa ter uma identidade. Mesmo que dois objetos compartilhem os mesmos recursos, cada objeto tem uma identidade única.

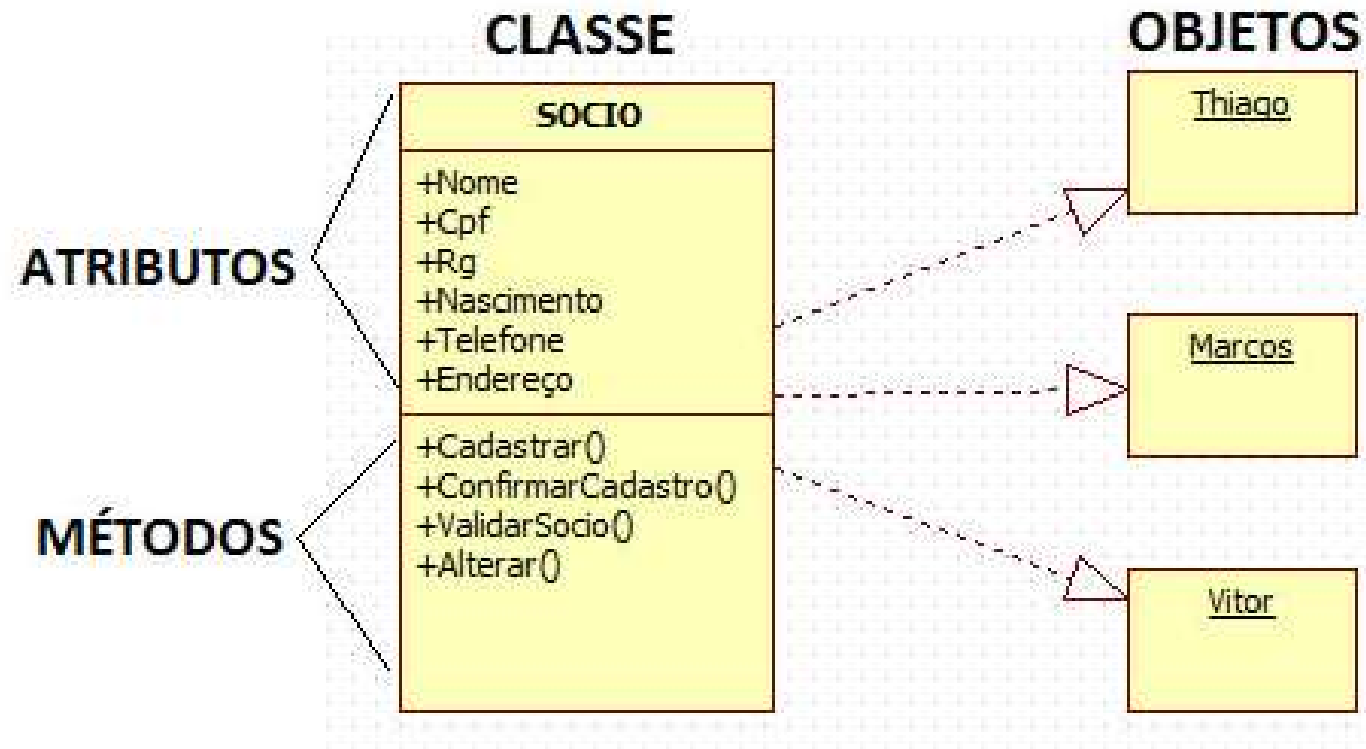
## Attributes

- Um objeto deve ser capaz de se descrever. Esse tipo de informação é armazenado nos atributos de um objeto que forma a estrutura do objeto..

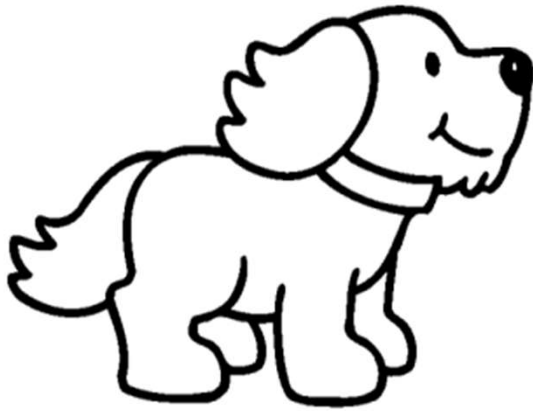
## State

- Um objeto deve ser capaz de descrever sua condição atual, chamada estado. Às vezes, o estado do objeto é representado pelos valores de cada um de seus atributos.

# Classes vs. Instances



# Classes vs. Instances



## ▶ Atributos

- Raça: Poodle
- Nome: Rex
- Peso: 5 quilos

## ▶ Método

- Latir
- Comer
- Dormir



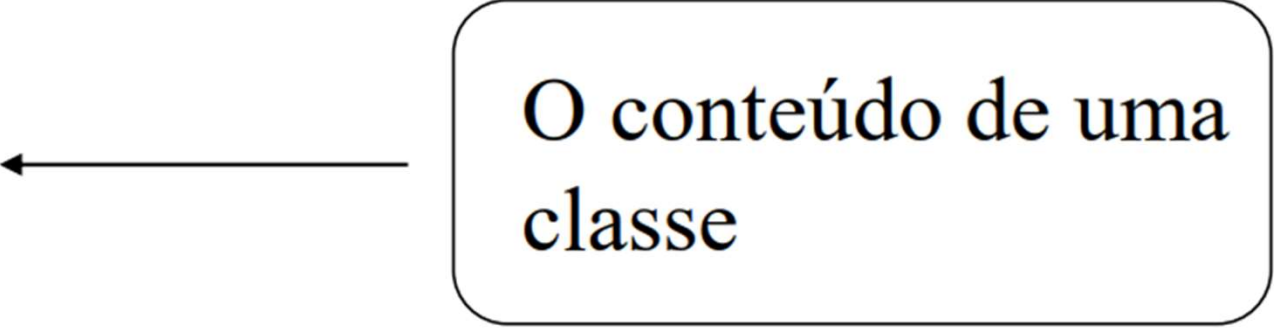
- Potência: 500cc
- Modelo: Honda
- Ano: 1998

- Acelerar
- Frear
- Abastecer

# Classes vs. Instances

## O que tem uma Classe?

```
public class NomeDaClasse
{
    Campos
    Construtores
    Métodos
}
```



O conteúdo de uma classe

# Classes vs. Instances

## O que tem uma Classe?

- Os **campos** armazenam dados para uso de cada objeto.
- Os **construtores** permitem que cada objeto seja configurado adequadamente quando ele é criado.
- Os **métodos** implementam o comportamento dos objetos.



# Overview of Object-Oriented Programming

## Classe:

Class Name = Substantivos

Fields = Adjetivos

Methods = Verbos

Class declaration

```
1 public class Customer {  
2     public String name = "Junior Duke";  
3     public int    custID = 1205;  
4     public String address;  
5     public int    orderNum;  
6     public int    age;  
7  
8     public void displayCustomer() {  
9         System.out.println("Customer: "+name);  
10    }  
11 }
```

Fields  
(Properties)  
(Attributes)

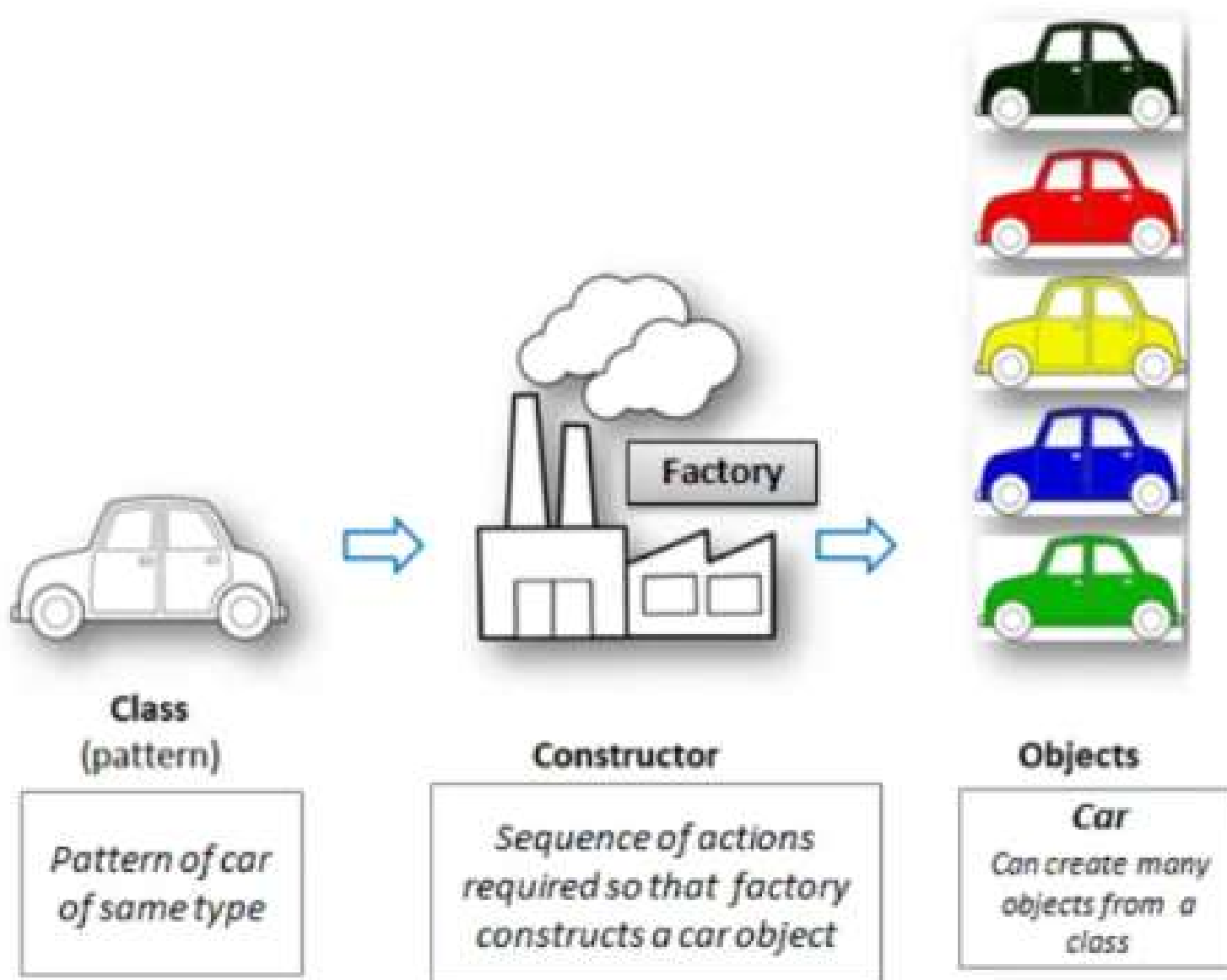
Methods  
(Behaviors)

# Concept of Constructor

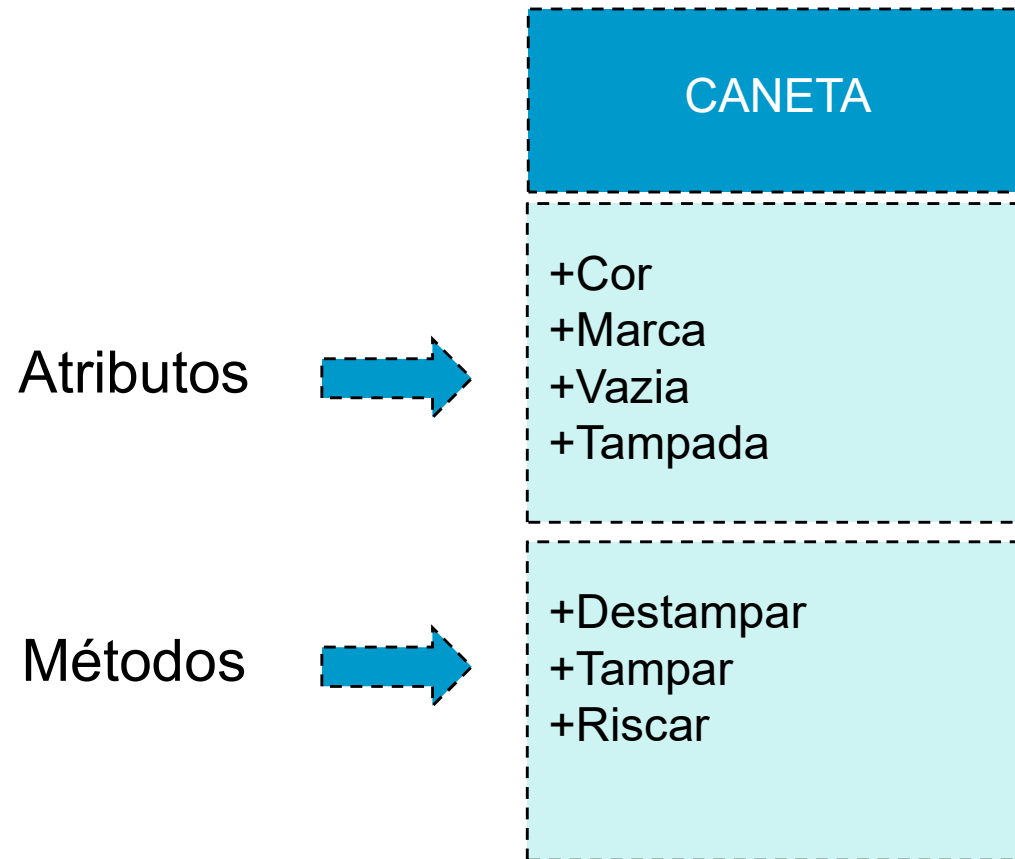
- Construtores são métodos que definem o estado inicial de um objeto;
- O nome do construtor deve ser o mesmo que o nome da classe;
- O construtor não pode retornar um valor, nem mesmo nulo;
- Construtores não podem ser herdados;
- Se a classe não declarar explicitamente um construtor, o padrão será um construtor no-parameter, do-nothing;

```
public class Cube1{  
    private int tamanho;  
  
    Cube1(int tamanho)  
    {  
        this.tamanho = tamanho;  
    }  
}
```

# Concept of Constructor



# Classes vs. Instances



**i** Refer to the Caneta.java and PricipalCaneta.java sample code.

# Criando e Manipulando Instâncias da Classe

- Instâncias de uma classe podem ser criadas usando a palavra-chave '**new**'. Esse processo é chamado de "instanciação".
- A palavra-chave '**new**' cria o objeto com base na classe especificada e retorna uma referência ao objeto recém-criado.

**NEW**



*Pessoa* *persona* = *new* *Pessoa* ("Zé Mané", 25, 'M');



- A referência é então recebida por um identificador apropriado.

# Criando e Manipulando Instâncias da Classe (cont.)

- Os nomes dos identificadores são usados para acessar os atributos do objeto.



```
persona.setAge(23);
```

```
persona.getName();
```



- O acesso é feito através do uso da notação 'ponto'.

# Custom Classes

- Um aspecto primário de uma linguagem de programação orientada a objetos é a capacidade de definir classes personalizadas;
- Uma declaração de classe é o componente básico de um aplicativo Java;
- Todo o código é criado e anexado a uma declaração de classe. Não há código que não pertença a uma classe;

# Custom Classes (cont.)

- Uma classe Java denota uma categoria de objetos e atua como um modelo para a criação de tais objetos;
- As classes declaram a que pacote pertencem;
- Classes declaram quais classes importar de outros pacotes;
- Os campos (também conhecidos como variáveis ou atributos) se referem às propriedades da classe;
- Métodos (também conhecidos como operações) se referem a comportamentos que a classe exibe;
- O arquivo de origem que contém a declaração de classe deve ter o mesmo nome que sua classe pública;



# Package Structure and the Class Path

- Packages são coleções de classes.
- Packages são usados para agrupar classes com base na funcionalidade ou relacionamento.
- As classes podem ser atribuídas a um determinado pacote usando o método *package* keyword:

```
package package_name;  
package sef.module4.samples;
```

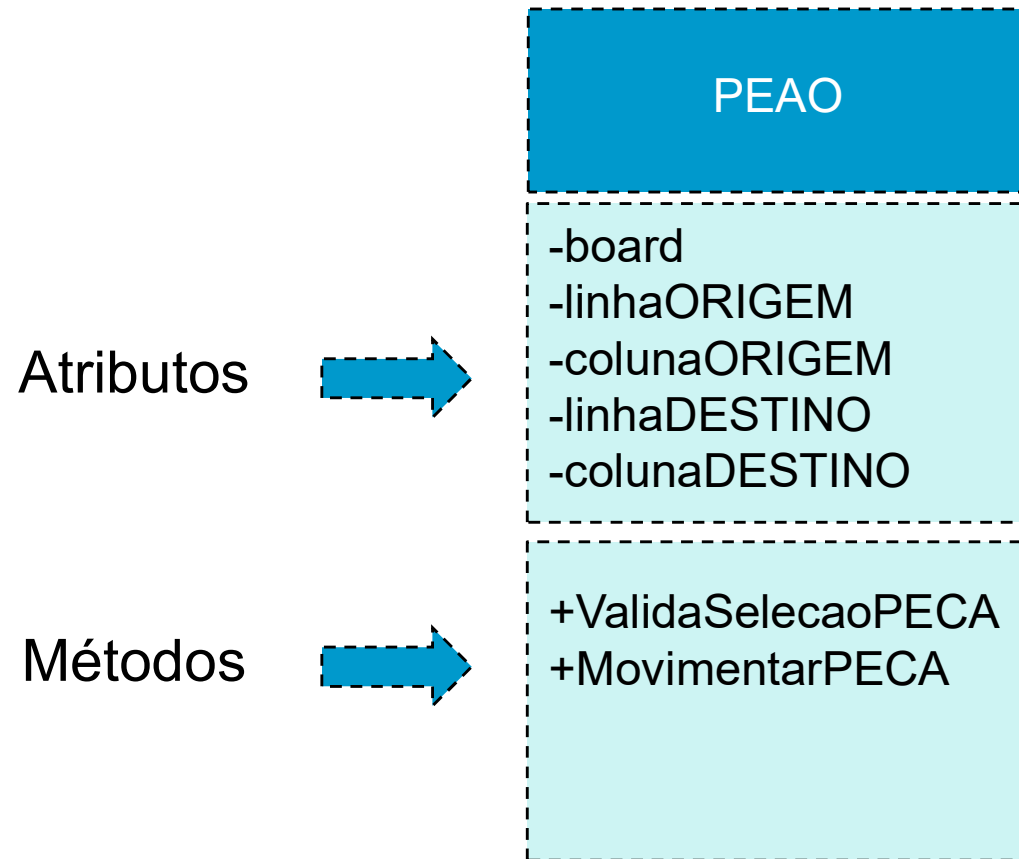
- A estrutura da package é representativa da estrutura de diretório em que a definição de classe está localizada.

# Importing Classes

- Um programa pode usar recursos externos importando-o;
- Os recursos importados podem ser uma única classe ou um pacote inteiro de classes;
- As classes importadas precisam estar visíveis no caminho da classe para serem acessíveis.

```
import package_name; ou import class-name;  
import java.util.*;  
import java.util.Calendar;
```

# Classes vs. Instances



# Activity – Create a Java File



## • Atividade 5

- Crie uma classe java **Gato.java**
- Com os atributos: **nome e idade**
- Com os métodos: **emitirSom;**
- No método constructor inicialize os atributos (não engessar os atributos);





# Activity – Create a Java File

## • Atividade 5

- Crie uma classe java **dog.java**
- Com os atributos: **nome e idade**
- Com os métodos: **emitirSom;**
- No método constructor inicialize os atributos (não engessar os atributos);



# Activity – Create a Java File



## • Atividade 5

- Crie uma **classe principal** java  
PrincipalAnimais.java com o método **main**
- Instancie a classe Gato e a classe Dog;
- Execute os métodos **emitirSom**;
- O Gato faz “miau”;
- O Dog faz “auau”;



# Activity – Create a Java File



## • Atividade 6

- Crie um Classe java **Carro.java**
- Com os atributos: **marca, CódigoCor, preço**
- Com os métodos, constructor: **Exibir, Ligar, Buzinar e Movimentar;**
- No método constructor inicialize os atributos;
- O método exibir exibe as atributos;
- No método Buzinar escreva “emitir som”.



# Activity – Create a Java File



## • Atividade 6

- Crie um classe java **CarroPrincipal.java** com o método **main**
- Instancie a classe **Carro**;
- Execute o método **Buzinar()**, porém o carro só buzina se estiver em movimento;
- Execute o método **Movimentar()**, porém o carro só anda se estiver ligado;



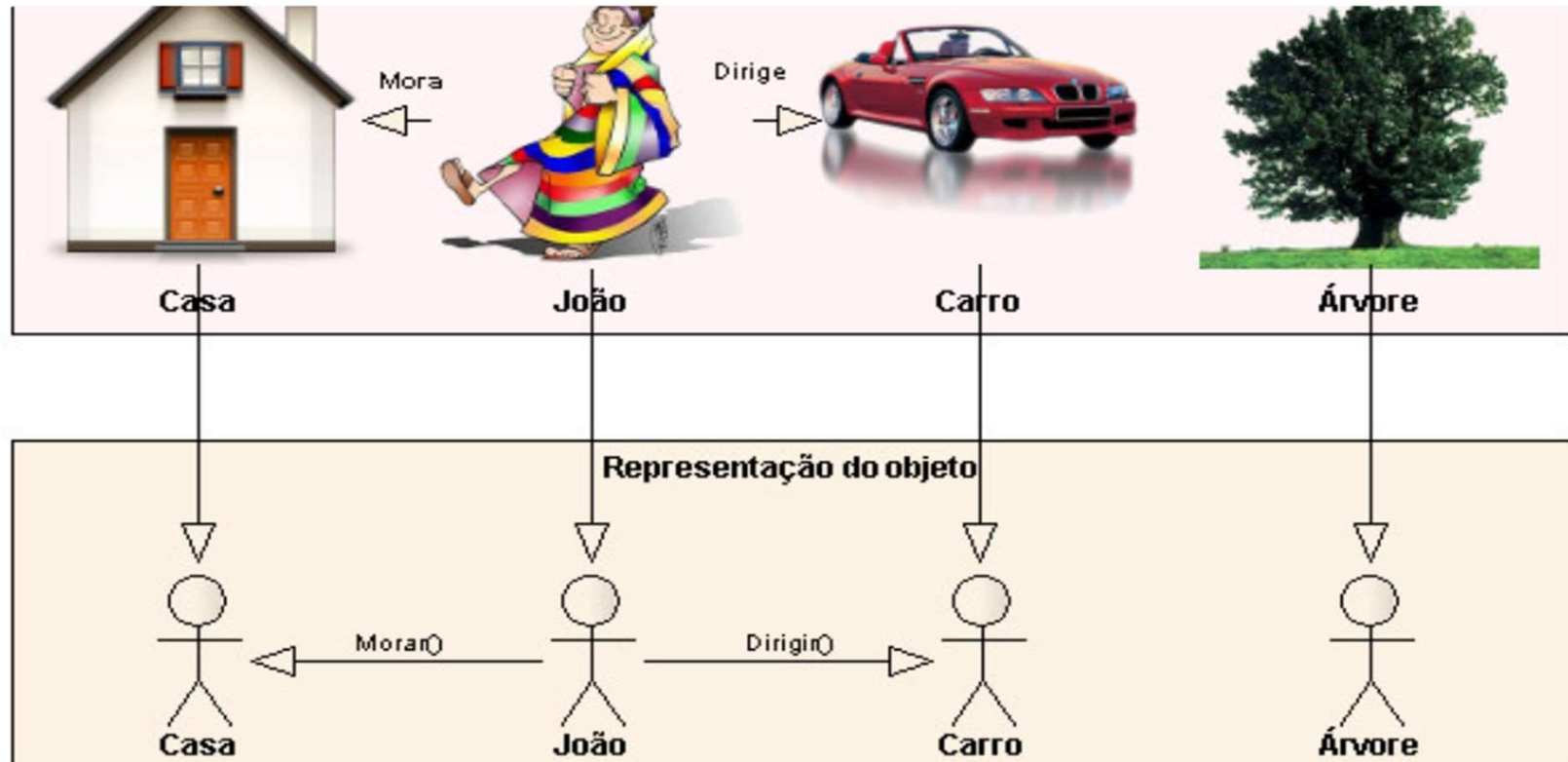




# Activity – Create a Java File

## • Atividade 7

Crie um classe java para cada objeto



# Perguntas:

## O que é uma classe?

Um modelo para criação de objetos;

## O que é uma instancia de classe?

Uma instância refere-se a um objeto que é membro de uma classe específica. Todos os objetos que pertencem a uma classe são instâncias dessa classe

## O que é um construtor?

Construtores são métodos que definem o estado inicial de um objeto. Tem o mesmo nome da classe;

# Questions and Comments

- What questions or comments do you have?

