



# PROGRAMAÇÃO ORIENTADA A OBJETO - JAVA

PROFESSOR: DANILO FARIAS  
SEMESTRE: 2020.1

23/05/2022

# PROGRAMAÇÃO ORIENTADA A OBJETOS



23/05/2022

# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Introdução

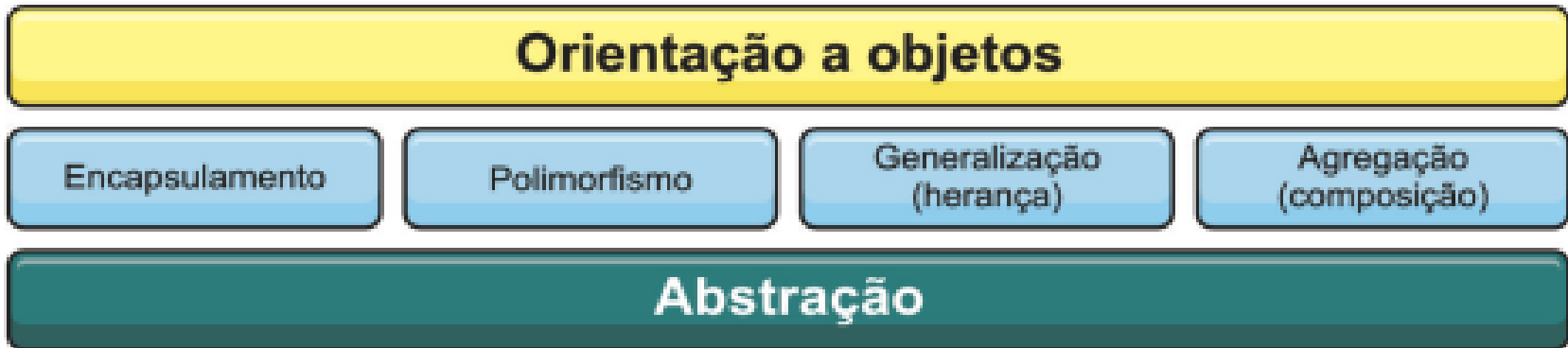
- A POO visa **aproximar** a maneira como construímos os programas de computador do mundo real
- O mundo real é composto por entidades que interagem, que trocam serviços entre si para realizar suas tarefas
- Na proposta de orientação a objeto, estas entidades são chamadas de **objetos**

# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Introdução (Cont.)

- Principais pilares do paradigma orientado a objetos



23/05/2022



# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Objetos

- Objetos são a chave para se compreender a tecnologia orientada a objetos.
- Você olha ao seu redor e tudo o que vê são *objetos*:



# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Objetos

- Os objetos do mundo real têm duas características em comum:
  - Estado e
  - Comportamento.
- O **estado** de um objeto revela seus dados importantes.
  - Uma pessoa tem: idade, peso, altura, cor de cabelo, cor da pele.
- O **comportamento** de um objeto, são as ações que aquele objeto pode exercer ou executar.
  - Uma pessoa pode: andar, falar, ouvir, pular.

# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Objetos

- São instâncias de classes, com valores diferentes
- Representa uma entidade do mundo real
- Cada objeto tem:
  - Identidade
  - Estado
  - Comportamento
- Uma porta tem:
  - Estado: aberto, fechado, ...
  - Comportamento: abrir, fechar, ...

# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Objetos (Cont.)

- Um objeto pode representar
  - Entidades concretas: pessoas, livros, discos
  - Entidades abstratas: transações bancárias, sessão, uma chave de criptografia

Exemplo:



Fonte: <http://img.alibaba.com/>





# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Objetos

- Objetos podem ser:
  - Concretos
    - carro, livro, nota fiscal...
  - Abstratos:
    - poupança, conta corrente, venda, pessoa jurídica...



# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Objetos

- Em OO, os objetos do mundo real são modelados e representados no mundo computacional, por meio de objetos de software.
- Os objetos de software, assim como os objetos do mundo real, também possuem estado e comportamento.
- Um objeto de software mantém seu estado nas variáveis e implementa seu comportamento através de seus métodos (função ou procedimento).

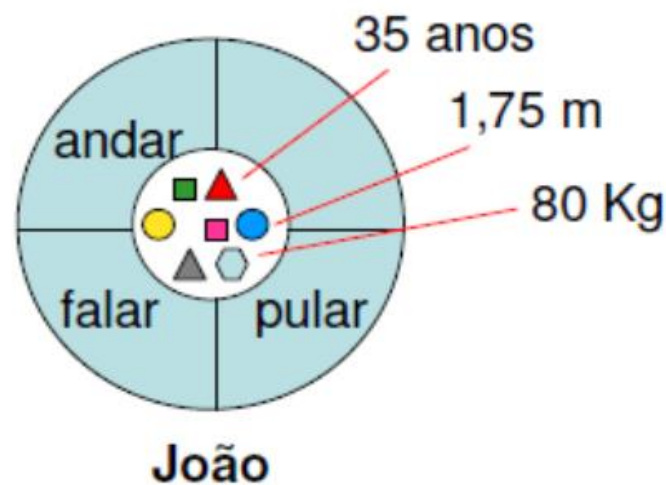
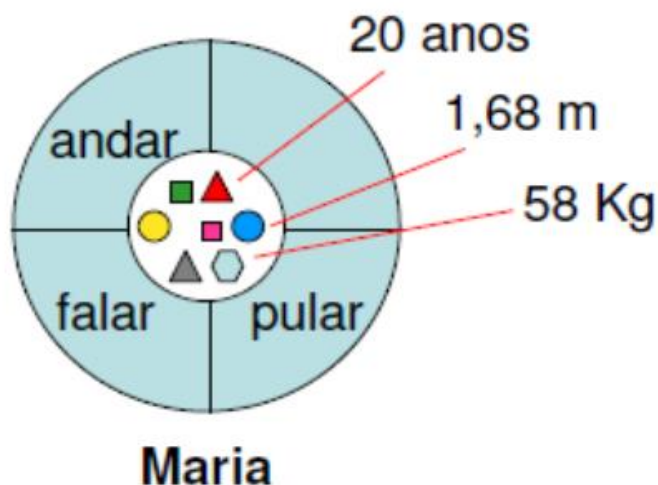
Um **objeto** é um pedaço de software que possui **variáveis** (estado) e **métodos** (comportamento).

# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Objetos

- Um sistema pode conter:
  - Um ou vários objetos ativos (**instâncias**).
- As diferentes **instâncias** possuem seu próprio estado.



Cada instância de pessoa possui um **estado** diferente em particular e **métodos** (comportamento) que operam sobre o próprio estado.



# PROGRAMAÇÃO ORIENTADA A OBJETOS

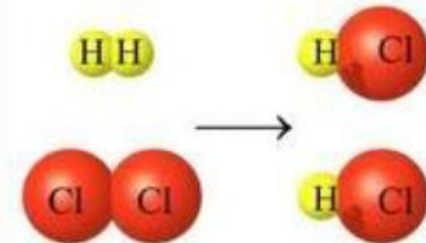


## Objetos

*“Um objeto representa uma **entidade**, unidade ou item identificável, individual, real ou abstrato, com um papel bem definido no **domínio do problema**.” [Booch, 91]*



Objeto físico



Objeto não físico

# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Classe



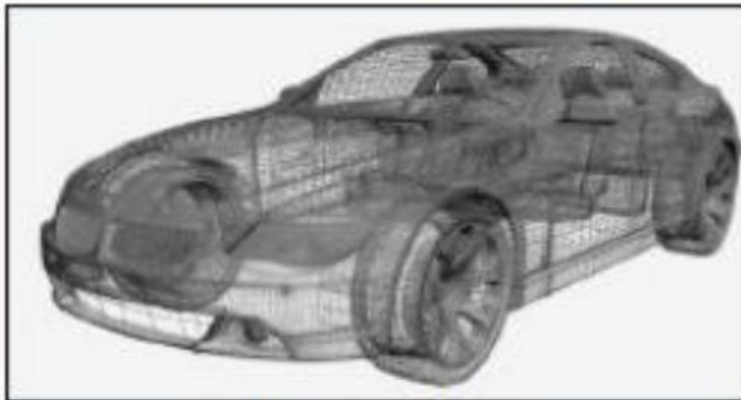
Uma classe é um modelo (protótipo) que define as variáveis (estado) e os métodos (comportamento) comuns a todos os objetos do mesmo tipo.



# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Classe

- Um **carro** em particular é uma instância da classe de objetos conhecida como **carros**;
  - Os **carros** possuem estado (cor, potência do motor, combustível) e comportamento (ligar, acelerar, frear, mudar marcha) em comum.



**Classe carro**

**Objeto carro**



# PROGRAMAÇÃO ORIENTADA A OBJETOS

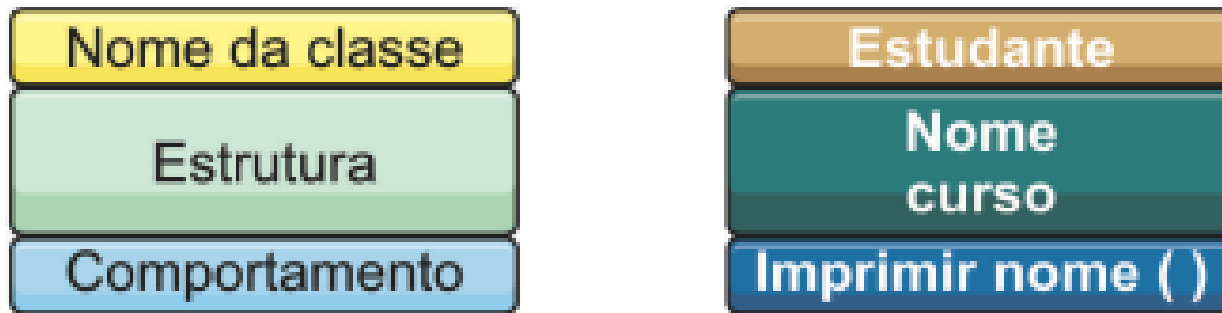
UML (do inglês, *Unified Modeling Language*)

- Representação bastante difundida para representar graficamente classes em um modelo orientado a objetos
- Utilizada para representação de muitas etapas durante a análise de um sistema
- Formada por vários diagramas
- Um deles é o diagrama de classes, em que classes são representadas graficamente
- Para facilitar nossa explicação sobre os conceitos OO, utilizaremos a notação UML para descrever as classes nesta disciplina

# PROGRAMAÇÃO ORIENTADA A OBJETOS

UML (do inglês, *Unified Modeling Language*)

- Em UML, uma classe é representada como descrito na figura a seguir:



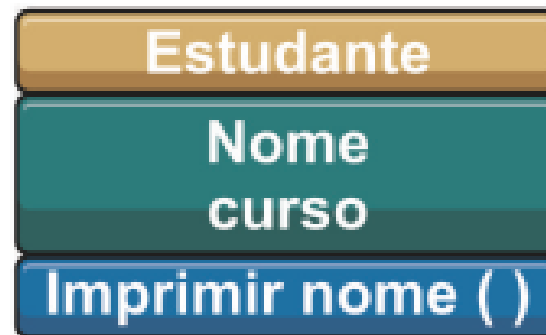
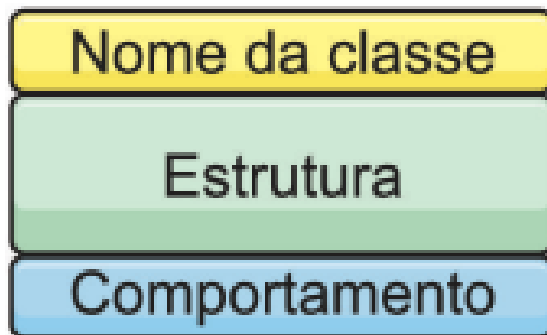
## Diagrama de Classe



# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Classe

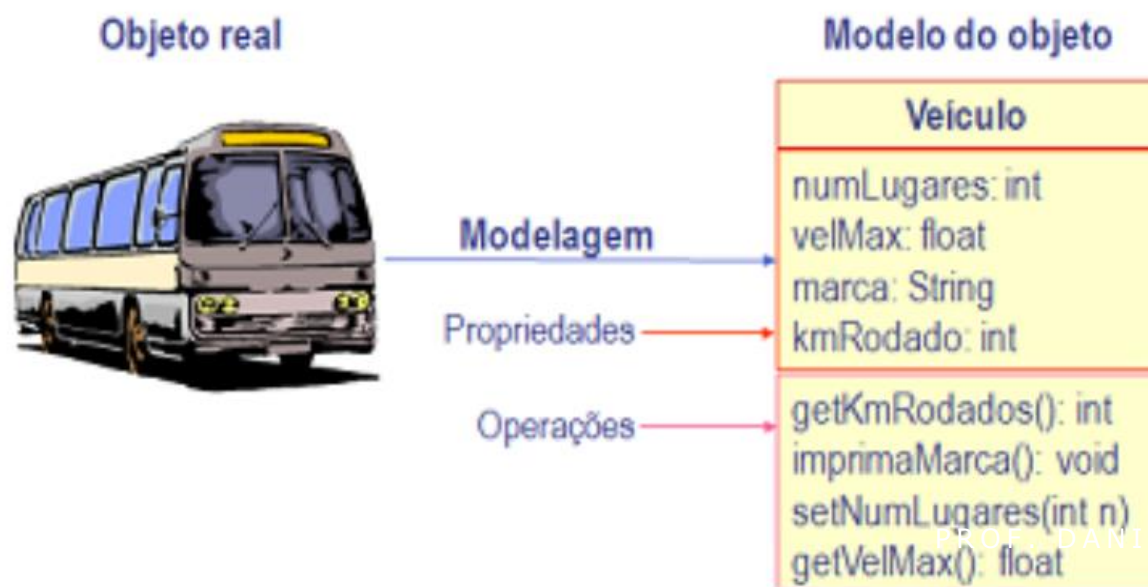
- Especifica a estrutura e o comportamento dos objetos
- São como “moldes” para a criação de objetos



# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Classe

- Programar em OO é basicamente definir classes para os elementos do Domínio do Problema em questão;
- Entidades do Domínio do Problema se transformam em objetos de software através da criação e implementação de classes.



# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Classe

- Para construir classes é preciso, primeiramente identificá-las:
  - Procurando por Entidades externas (outros sistemas):
  - Coisas, Lugares, Ocorrências ou eventos, Papéis (funções) desempenhados por alguém num sistema, Unidades de uma instituição, etc.



# OBJETOS X CLASSES

**São estruturas relacionadas, mas bem distintas:**

- Objeto é uma instância de uma classe;
- Objeto pertence a uma classe;
- Os objetos de uma mesma classe se diferenciam pelo conteúdo dos seus atributos;
- Objetos existem apenas na memória, quando o programa está em execução;
- Classe define a estrutura de um objeto:
  - Atributos e métodos;
- Classe define o método de se construir (instanciar) um novo objeto:
  - Construtores;

# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Atributos

- São os dados internos do objeto
- Só podem ser declarados no corpo de uma classe Java e são visíveis por qualquer outro membro desta classe
- Representam o estado do objeto



# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Mensagem

- Os objetos precisam interagir e comunicar-se entre si;
  - Eles se comunicam através de mensagens.



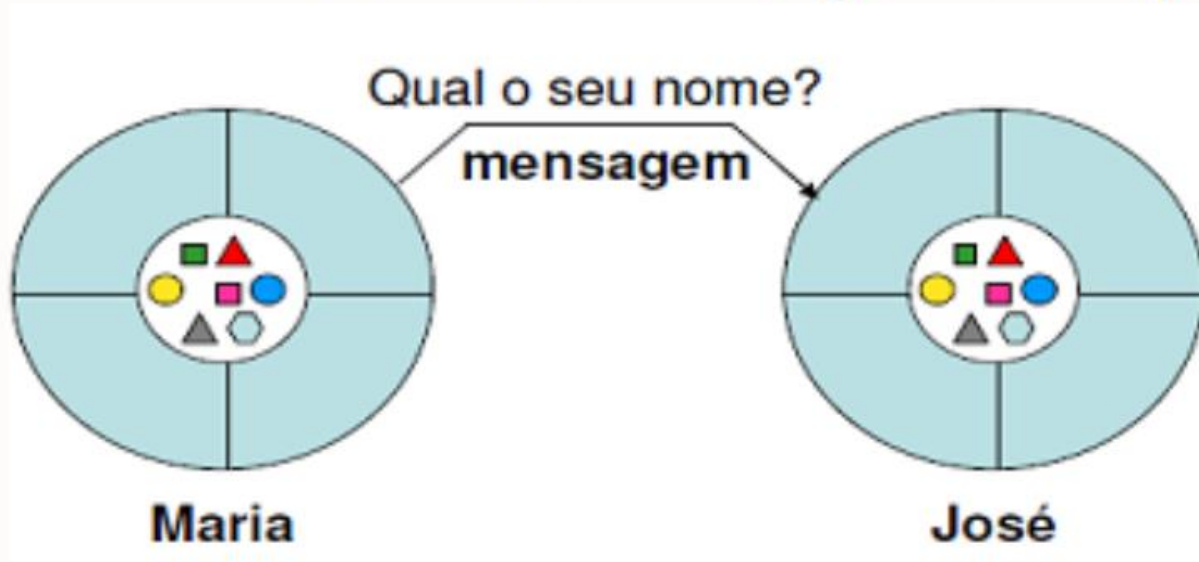


# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Mensagem

- Quando um objeto A quer se comunicar com um objeto B é enviada uma mensagem de A para B.

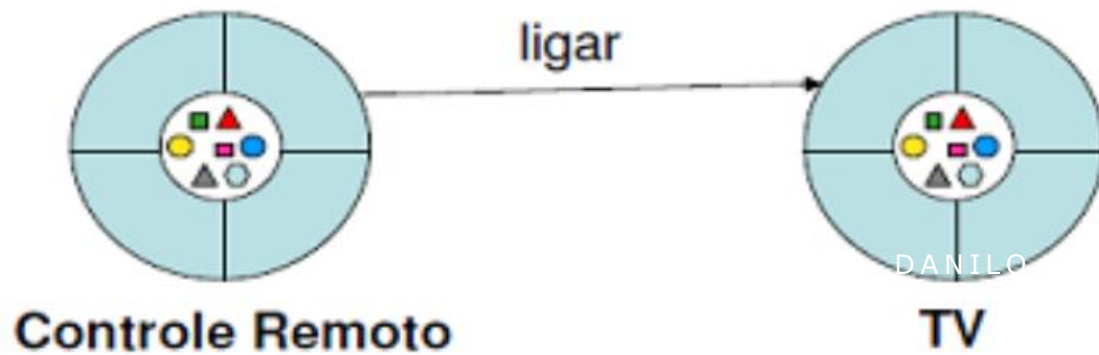


- Enviar uma mensagem significa executar um método.
- Se A envia uma mensagem para B  
→ O objeto A está executando um método do objeto B.

# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Mensagem

- As mensagens são compostas por três partes:
  - Parte 1  
Objeto a quem a mensagem é endereçada
  - Parte 2  
Nome do método a ser chamado
  - Parte 3  
Parâmetros que o método recebe



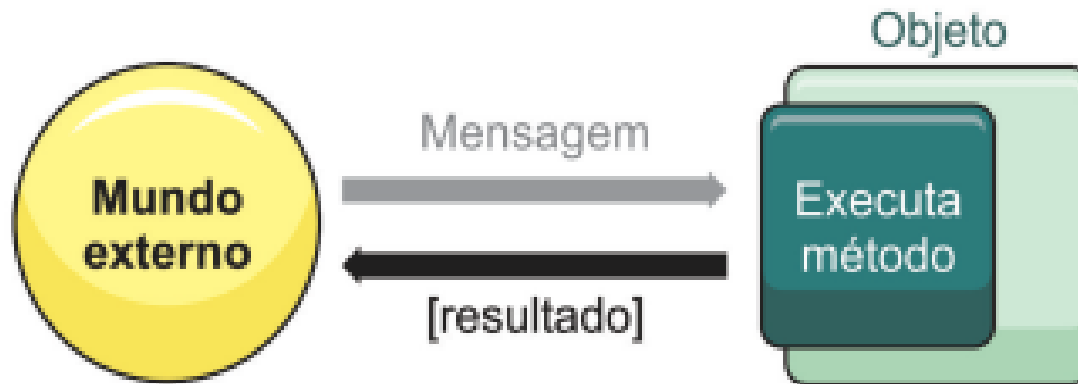


# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Método

- É uma operação que realiza ações e modifica os valores dos atributos do objeto responsável pela sua execução



# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Classe

```
public class NomeDaClasse {  
    CorpoDaClasse  
}
```

O corpo de uma classe pode conter

- atributos
- métodos
- construtores (inicializadores)
- outras classes...

# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Classe

```
public class Conta {  
  
    // Atributos  
    private Cliente cliente;  
    private double saldo;  
    private double numero;  
  
    // Métodos  
    //sacar  
    public void sacar(double valor){  
        if (this.saldo >= valor) {  
            this.saldo = this.saldo - valor;  
        }else{  
            // Inconstitucional...  
            System.out.println("Saldo insuficiente.");  
        }  
    }  
}
```



# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Classe

```
public class <nome> {  
    public static void main (<parametros>)  
    {  
        <declarações>  
        <comandos>  
    }  
}
```

Onde, **main**: método por onde se inicia a execução  
**public**: parâmetro de acesso  
**static**: indica que main se aplica à classe  
**void**: indica que main não retorna um valor

# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Abstração

- Habilidade de modelar características do mundo real
  - Ignorando detalhes não relevantes
  - Visando representar uma realidade complexa num modelo mais simplificado
- *“Uma abstração depende mais do observador do que do objeto observado”*
- Só representamos características que nos interessem para determinado contexto



# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Agregação / Composição

- A ideia de agregação ou composição estabelece a criação de novas classes de objetos a partir da junção de classes já existente



# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Agregação / Composição

- Há uma notação especial em UML para representar agregação
  - Um losango vazado é anexado à extremidade de um caminho de associação ao lado do agregado (o todo) para indicar agregação

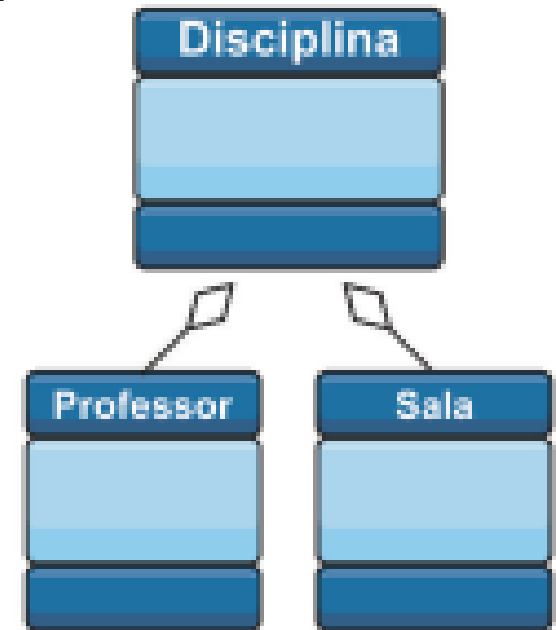


# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Agregação / Composição

- Os objetos contidos podem existir sem serem parte do objeto que os contém
- Exemplo:
  - Um professor continuará a existir, mesmo que não haja o conceito Disciplina



**Losango vazado**

23/05/2022

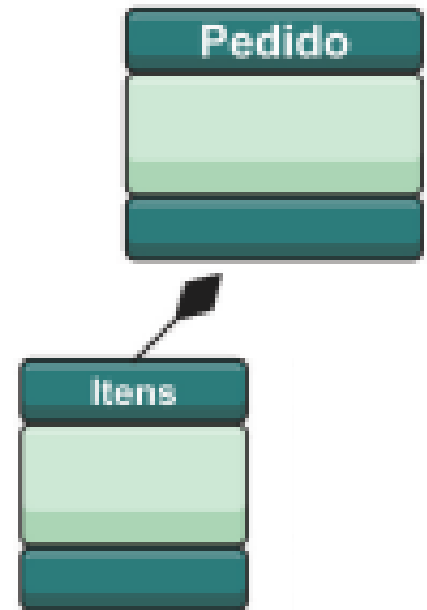




# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Agregação / Composição

- O objetos contidos não fazem sentido fora do contexto do objeto que os contém
- Exemplo:
  - Um pedido contém itens vinculados. Se você destruir o pedido, os itens são destruídos junto, pois eles não tem sentido fora do pedido



**Losango preenchido**

# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Associação

- As entidades envolvidas apresentam existências independentes, mas existe uma ligação entre elas
- Consiste na descrição genérica de uma ou mais ligações entre as classes, permitindo que objetos de uma classe utilizem recursos de outro objetos



# PROGRAMAÇÃO ORIENTADA A OBJETOS



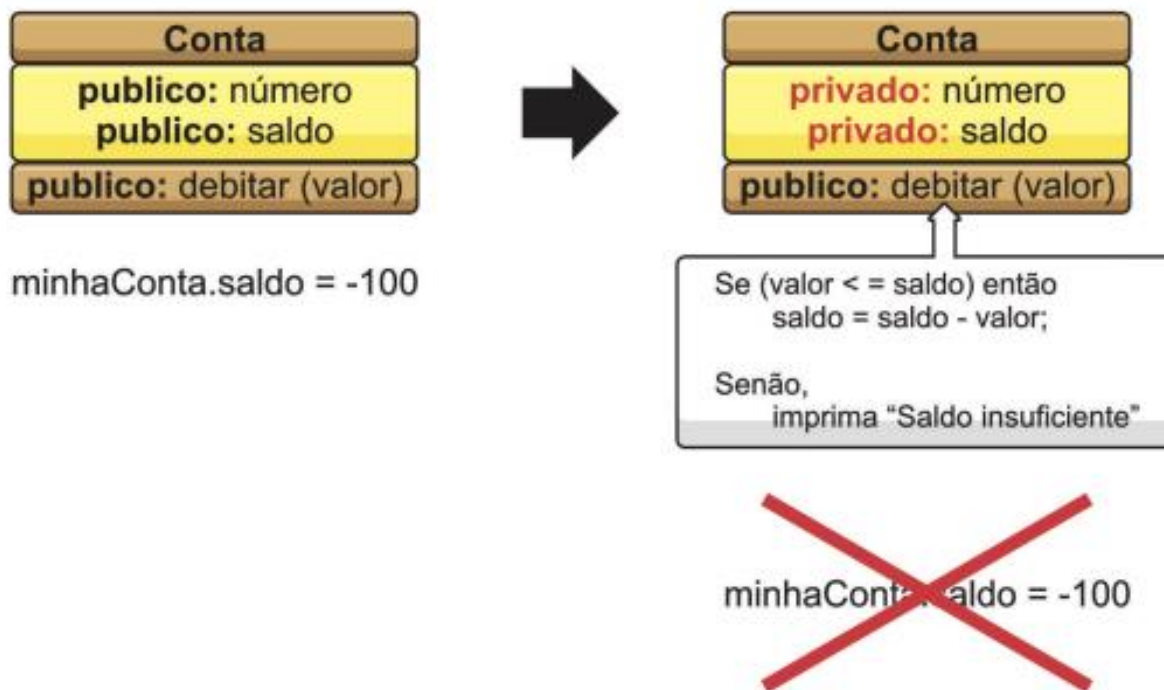
## Encapsulamento

- Consiste em separar os aspectos externos de um objeto dos detalhes internos de implementação do objeto
- Evita que objetos possuam grandes dependências entre si, de modo que uma simples mudança em um objeto possa trazer grandes efeitos colaterais e problemas para outros objetos
- Em POO, o acesso a componentes de um objeto é controlado e, especialmente, os atributos de um objeto só devem ser modificados pelos métodos do próprio objeto



# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Encapsulamento



# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Herança

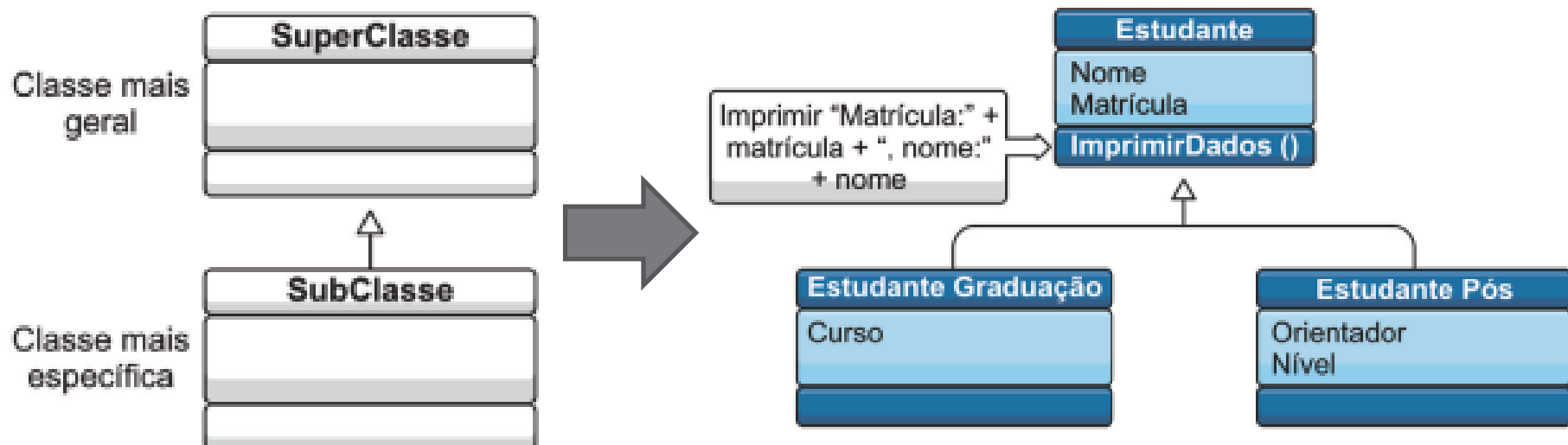
- É o mecanismo pelo qual pode-se definir uma nova classe de objetos a partir de uma classe já existente
- Esta nova classe poderá aproveitar o comportamento e possíveis atributos da classe estendida
- A classe sendo refinada é chamada de superclasse ou classe base, enquanto que a versão refinada da classe é chamada uma subclasse ou classe derivada



# PROGRAMAÇÃO ORIENTADA A OBJETOS

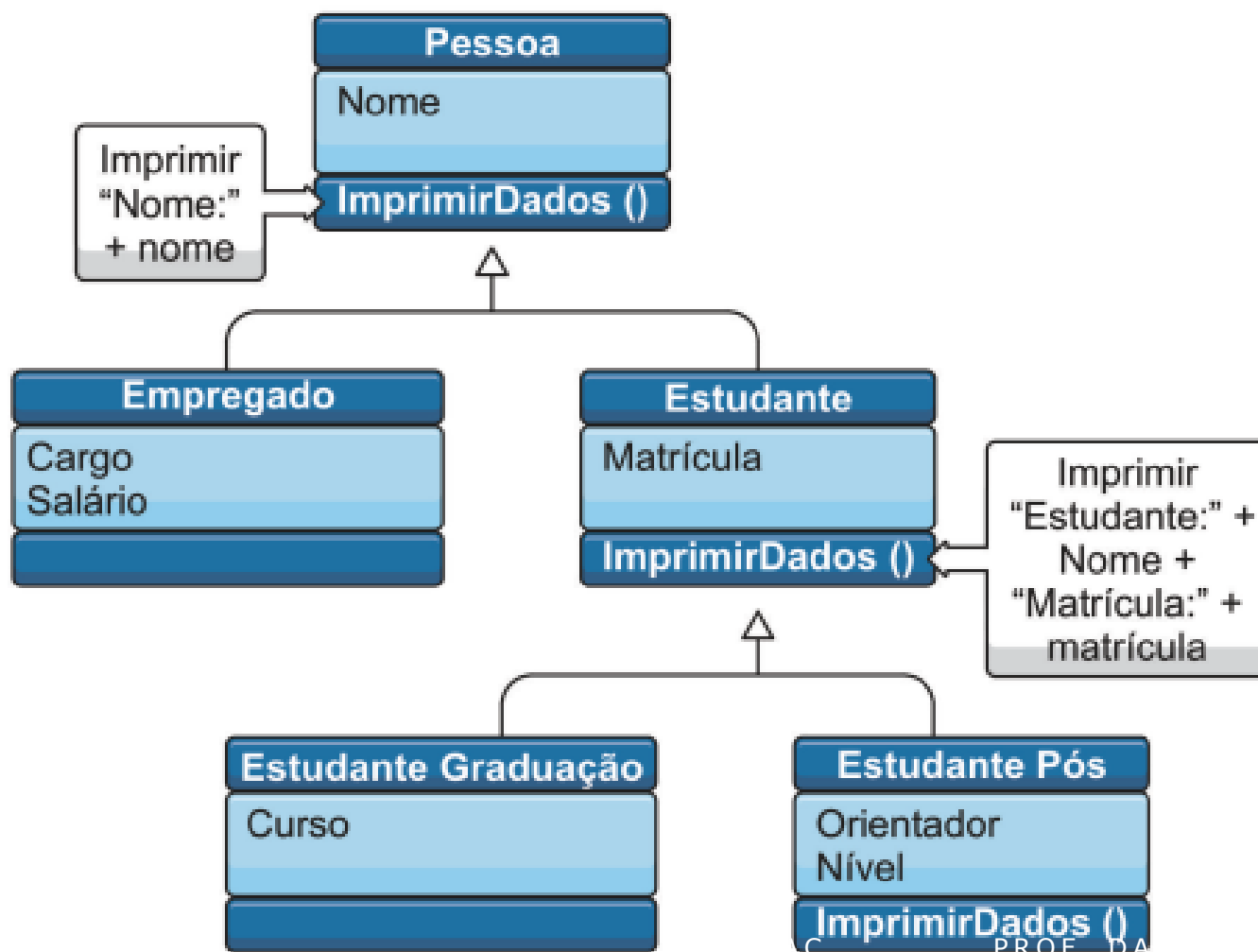


## Herança



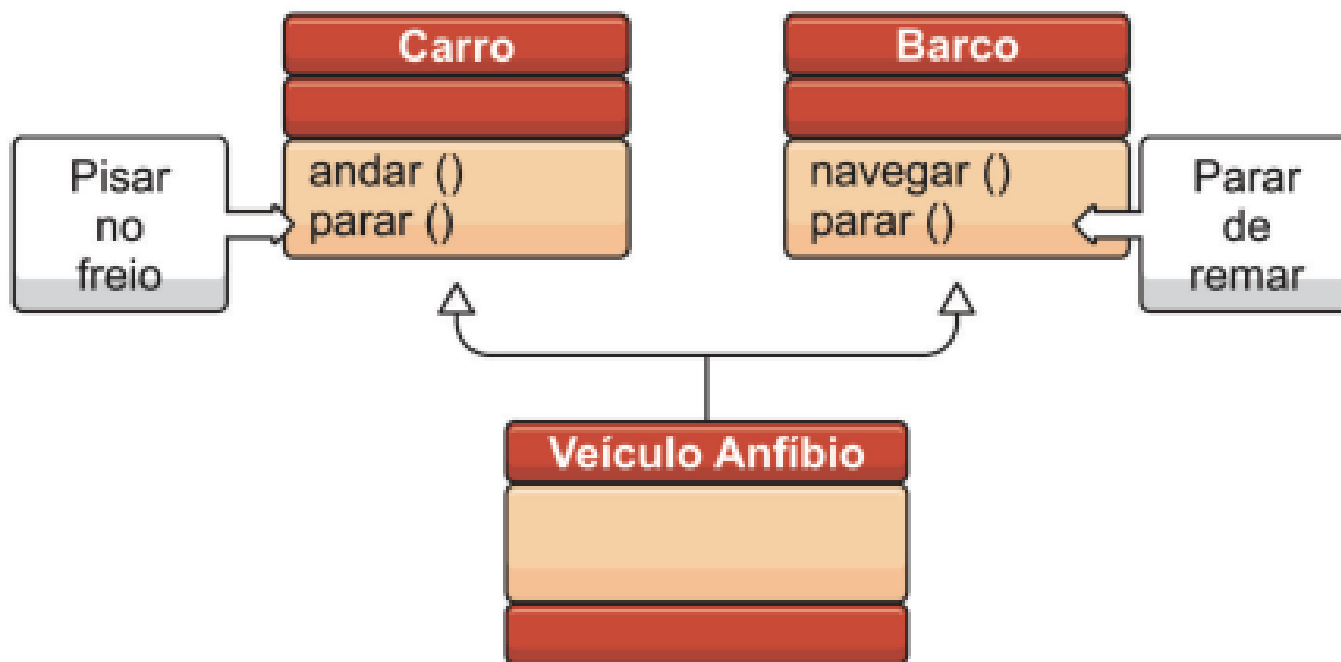
# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Herança::Múltiplo níveis



# PROGRAMAÇÃO ORIENTADA A OBJETOS

## Herança múltipla



■ Java



# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Polimorfismo

- É originário do grego, e quer dizer “muitas formas”
- Significa que um mesmo tipo de objeto, sob certas condições, pode realizar ações diferentes ao receber uma mesma mensagem

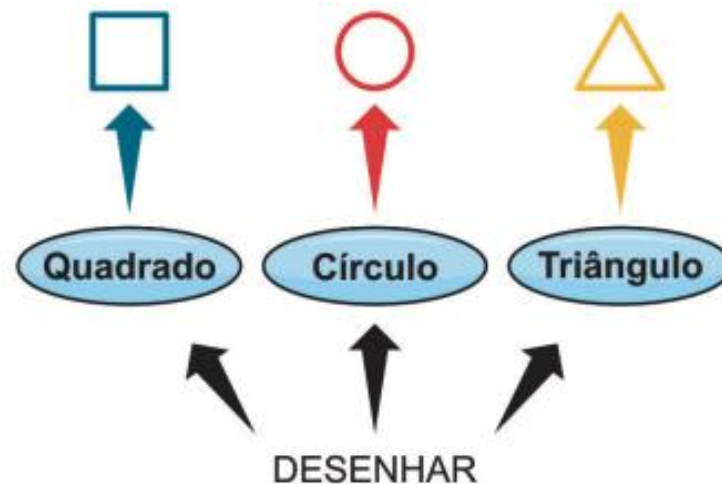


# PROGRAMAÇÃO ORIENTADA A OBJETOS



## Polimorfismo

- Permite o envio de uma mesma mensagem a objetos distintos, onde cada objeto responde da maneira mais apropriada

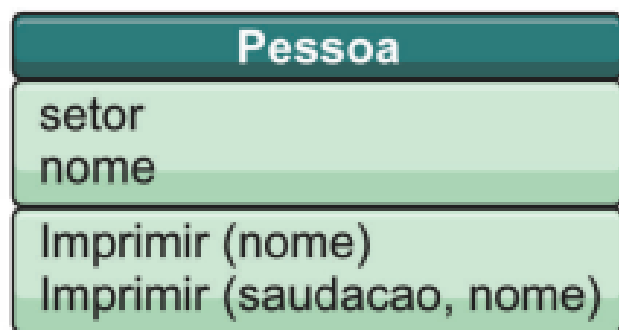


# PROGRAMAÇÃO ORIENTADA A OBJETOS

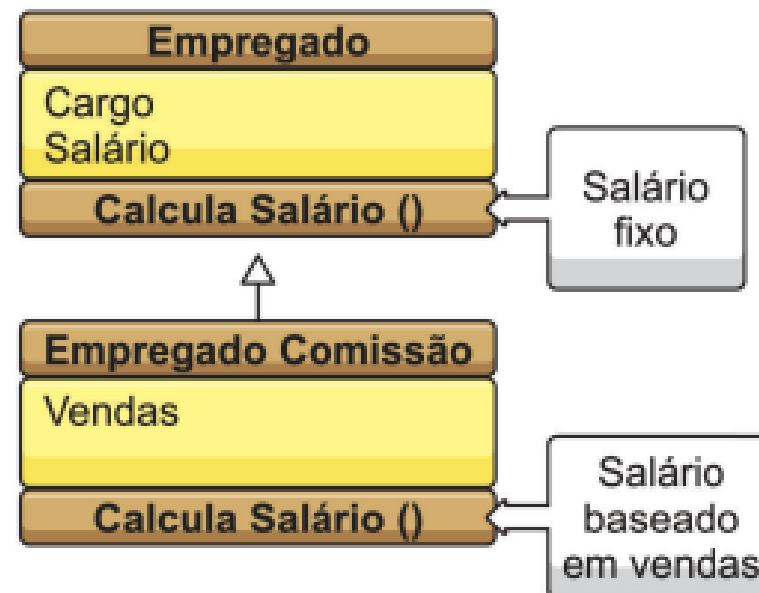


## Polimorfismo

- Tipos de polimorfismo



### Sobrecarga



### Sobreposição





# PROGRAMAÇÃO ORIENTADA A OBJETO - JAVA

PROFESSOR: DANILO FARIAS  
SEMESTRE: 2020.1

23/05/2022