

O.O.

Conceitos

Paradigma Orientado a Objetos

Prof. Renato de Tarso

Orientação a Objetos

P.O.O X Procedural/Estruturado

Estruturado

- Código mais “simples”, o fluxo é mais visível;
- Basicamente com 3 estruturas:
 - **Sequência, Decisão e Iteração;**
- Não se tem a visão do todo;
- Menor potencial de Reuso e Manutenção;



Orientado a Objetos

- Código mais “limpo” e coeso, fluxo mais abstraído;
- Maior organização e escalabilidade;
- Alto poder de Reuso e Manutenção;

Orientação a Objetos

P.O.O X Procedural/Estruturado

Tabela - O.O X Procedural/Estruturado

Orientado a Objetos	Procedural/Estruturado
<i>Métodos</i>	<i>Procedimentos e funções</i>
<i>Instâncias de variáveis</i>	<i>Variáveis</i>
<i>Mensagens</i>	<i>Chamadas a procedimentos e funções</i>
<i>Classes</i>	<i>Tipos de dados definidos pelo usuário</i>
<i>Herança</i>	<i>Não disponível</i>
<i>Polimorfismo</i>	<i>Não disponível</i>

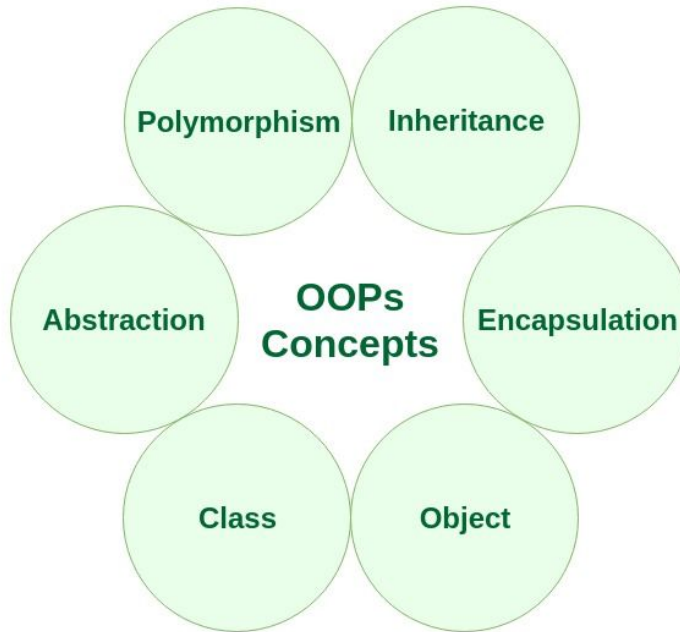
Fonte: [DevMedia](#)



Orientação a Objetos

Levar Entidades e Conceitos do mundo real, como Objetos, para programação.

Imagem - Conceitos O.O



Fonte: [GeeksforGeeks](https://www.geeksforgeeks.org/object-oriented-programming/)



Orientação a Objetos

Classes

- São padrões (molde/modelo/forma) que definem e operam Objetos.

Imagem - Classe = um Modelo, uma Definição



Fonte: Acervo próprio



Orientação a Objetos

Objeto

- Um objeto, é uma instância de uma Classe.

Imagem - Objeto P1

Objeto **P1** do tipo **Professor**

<u>p1: Professor</u>
- nome : "Cláudio" - matricula = "1234567" - dataContrato = 17/08/1990 - situacao = "Titular" - disciplina = "Matemática" - cargaHoraMax = 16
+ setDisciplina() + setCargaMax() + tirarFerias() + ministrarAula()

Classe **Professor** molda o objeto **P1**.

Professor
- nome: String - matricula: String - dataContrato: Date - situacao: Enum - disciplina: String - cargaHoraMax: Integer
+ setDisciplina() + setCargaMax() + tirarFerias() + ministrarAula()

3 Objetos
do Tipo **Professor**



Fonte: Acervo próprio



Orientação a Objetos

Abstração

- Facilita representações de conceitos, mas difere no nível de abstração ou no ponto de vista;
- A abstração depende do observador (consumidor);
- Foco em aspectos essenciais, ignorando aspectos irrelevantes;
- “Oculta” detalhes desnecessários de um objeto.



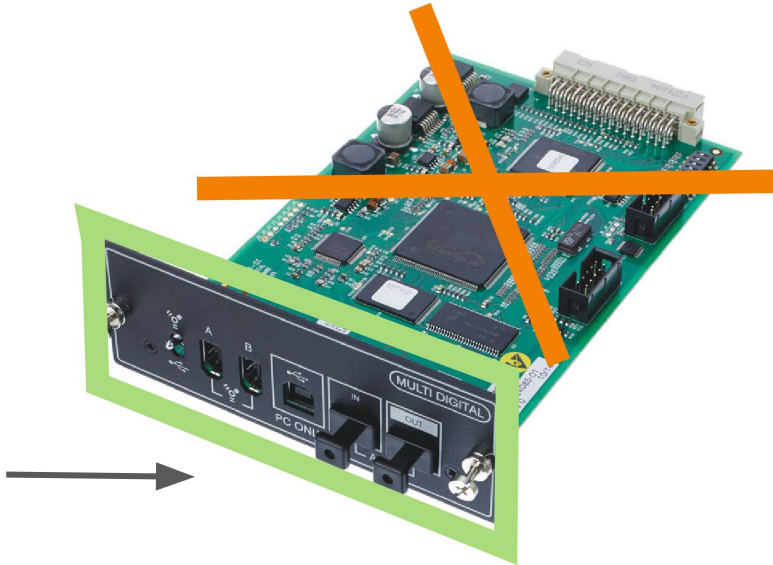
Orientação a Objetos

Abstração

- Simplifica, removendo a complexidade de objetos.

Imagem - Abstração Interface aparelho

A Interface
simplifica
(abstrai)
processos
eletrônicos
internos.



Fonte: Acervo próprio



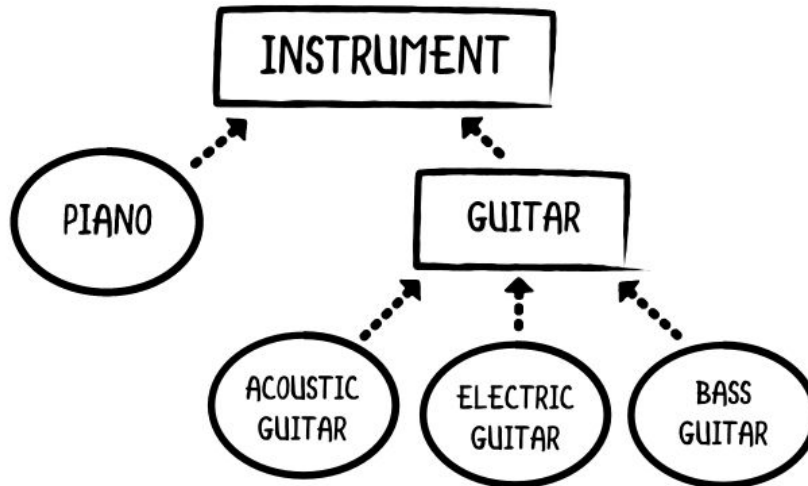
Orientação a Objetos

Abstração

- Generaliza o uso de objetos.
- Instrument > Guitar > Acoustic Guitar;



Imagem - Abstração Instruments



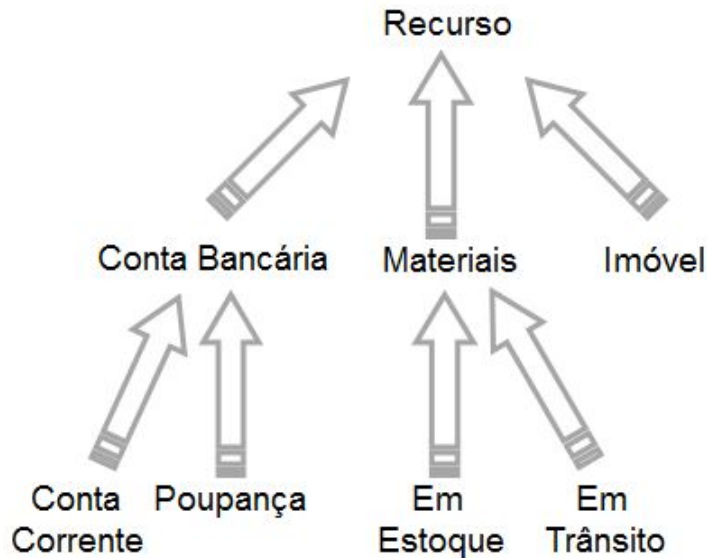
Fonte: [Tehmaster](#)

Orientação a Objetos

Abstração

- Nível de hierarquia no mesmo nível de abstração.

Imagem - Abstração Níveis de abstração



Fonte: Acervo próprio



Orientação a Objetos

Herança

- Classe permite herdar os recursos (estrutura e comportamento/método) de outra classe.
 - **Superclasse:** Classe cujos recursos são herdados (pai/mãe).
 - **Subclasse:** Classe que herda (derivada/filha). Pode ter estrutura própria e distinta.

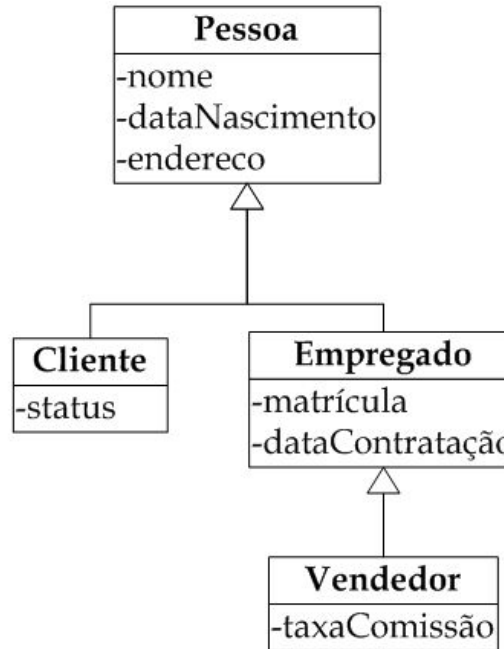


Orientação a Objetos

Herança

- Subclasses podem herdar (atributos e métodos) de todas as suas superclasses.
- Uma classe pode ter mais de uma superclasse;

Imagem - Abstração Pessoas



Fonte: [Tehmaster](#)



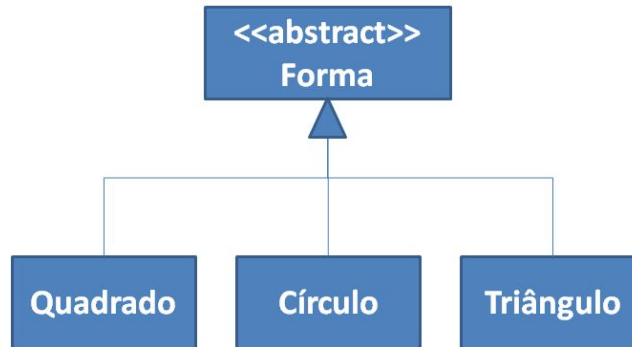
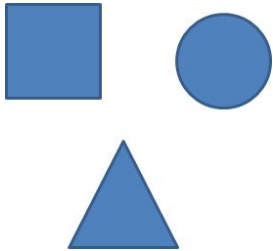
Orientação a Objetos

Herança

- Reaproveitar propriedades e comportamentos.

Imagem - Herança Interface aparelho

3 tipos de **Forma**



Fonte: Acervo próprio



Orientação a Objetos

Polimorfismo

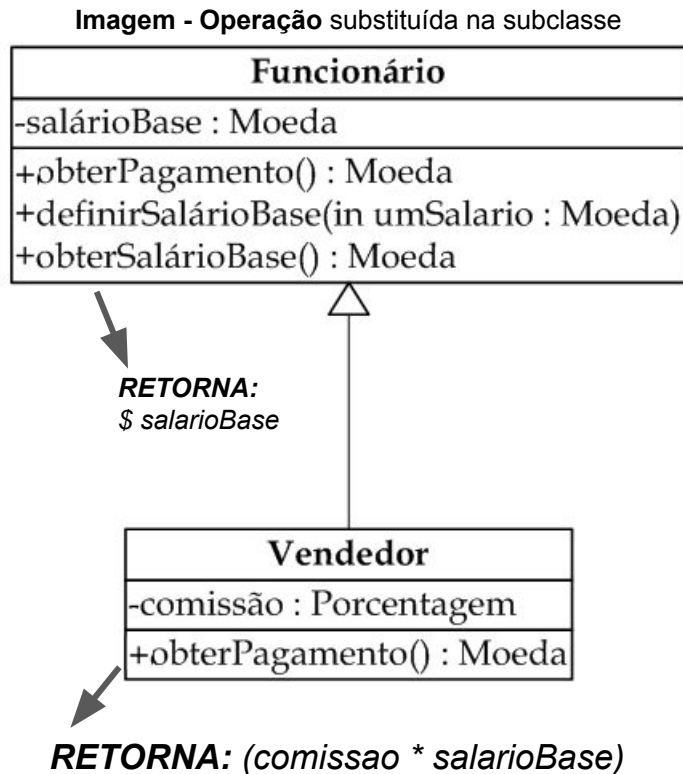
- Operações polimórficas: Operações com a mesma assinatura (método/operação);
 - Na subclasse redefine-se a implementação.
 - Classes diferentes respondendo a mesma operação (método), mas de formas diferentes.
-
- **Vantagens:**
 - Garantir que subclasses tenham interface (método) comum para todos.
 - O solicitante não precisa saber qual classe implementa, a mesma mensagem (método) é usada.



Orientação a Objetos

Polimorfismo

- Objetos de um mesmo tipo que possuem comportamentos diferentes, em situação distintas.
- Poder de Adaptar/Mudar;
- **Poli: Várias.**
Morfis: Formas;



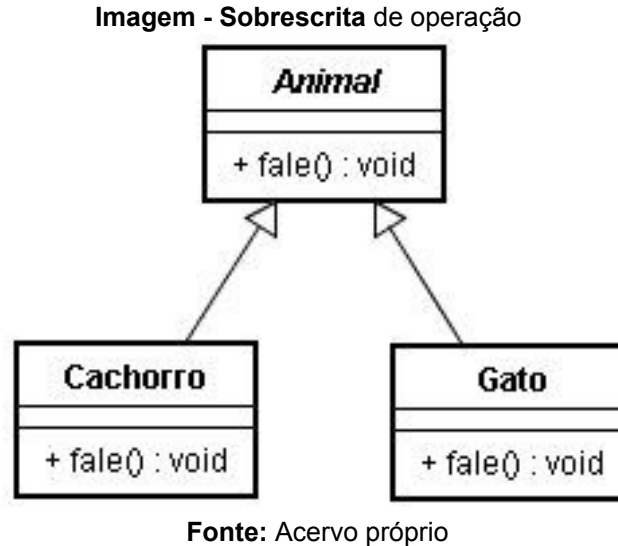
Fonte: Acervo próprio



Orientação a Objetos

Polimorfismo

- Sobrescrita de método/operação;
- A mesma operação realizada distintamente em várias classes;



Orientação a Objetos

Encapsulamento

- Esconde e protege propriedades da classe.
- Disponibiliza **O quê** se faz, e não **Como** se faz.
- Técnica de proteger sua estrutura das outras classes, apenas internamente acessíveis e implementadas.



Imagem - Encapsulamento



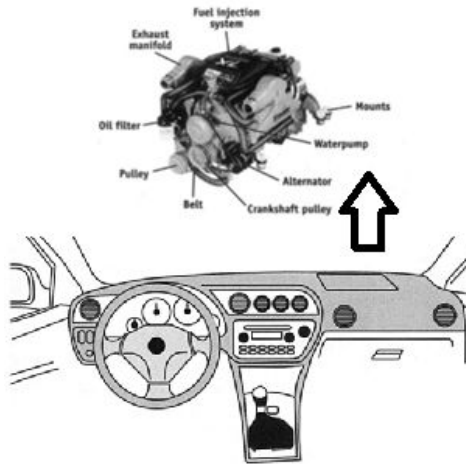
Fonte: Google Imagens

Orientação a Objetos

Encapsulamento

- As Partes do objeto são controladas apenas pelo próprio Objeto.
- “*Só o carro pode ligar o motor*”.

Imagem - Encapsulamento painel liga o motor



Fonte: Acervo próprio



Orientação a Objetos

Encapsulamento é proteção

- Visibilidade das propriedades da classe.

“-” = **privado**: Métodos e Atributos acessíveis somente pelo próprio objeto;

“+” = **público**: Acessível por quaisquer objetos;

“#” = **protegido**: Acessível por objetos de classes herdeiras.

* Detalhes adiante.

Imagem - Visibilidade Classe

Professor
- nome: String - matricula: String - dataContrato: Date - situacao: Enum - disciplina: String - cargaHoraMax: Integer
+ setDisciplina() + setCargaMax() + tirarFerias() + ministrarAula()

Fonte: Acervo próprio

