

AfroDev

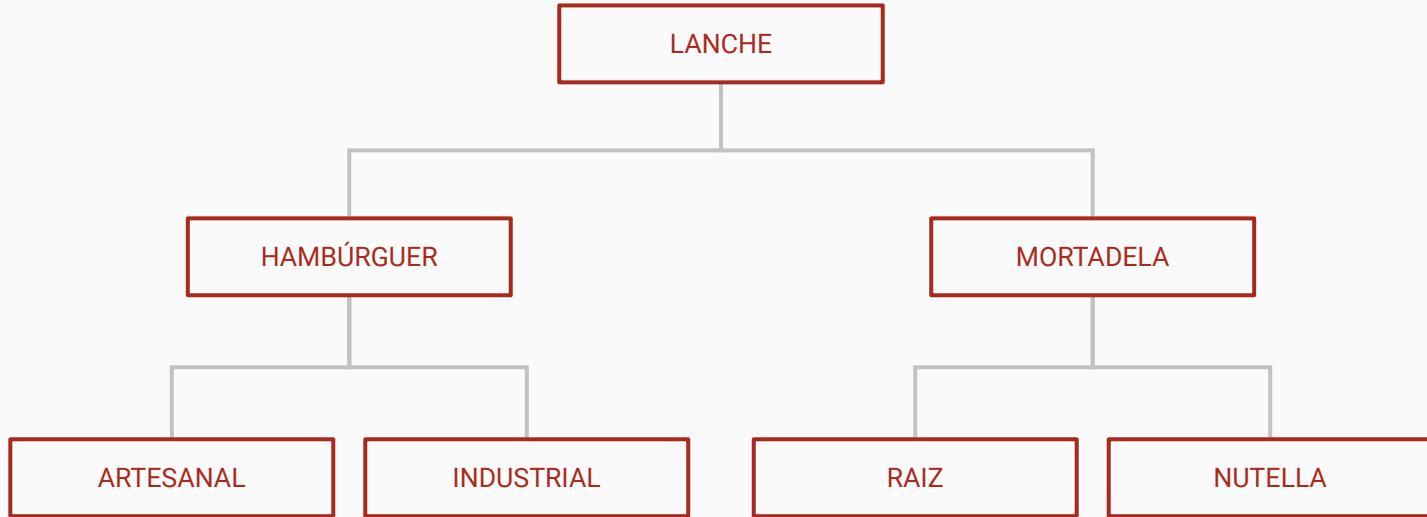
Java Básico - Aula 2 - Parte 2



Herança - Definição

- É um mecanismo natural para classes filhas herdarem atributos (membros) e comportamentos (métodos) da classe pai
- Um jeito de agrupar os atributos e comportamentos comuns e isso tudo ficar concentrado na classe pai (economia de código)
- Ainda que existam características comuns, a herança dá a possibilidade das classes filhas exporem suas particularidades em termos de atributos e comportamentos. Inclusive “mudar” o comportamento da classe pai

Herança - Exemplo



Herança - Dicas

- Regra de ouro: “É-UM(A)”
- Exemplo: HAMBÚRGUER “é um” LANCHE
- Conta Especial “é uma” Conta
- Carro “é um” Veículo
- Fusca “é um” Carro
- Nutella é uma mortadela?
- Caminho sem fim?
- Uso: Lanche lanche = new Hamburger();
- Quem vai usar, vai referenciar “Lanche” e não “Hamburger”. Vantagem para escrita de código!!!

Agregação - Definição/Dicas

- Regra de ouro: “CONTÉM”
- Carro “contém” portas
- Banco “contém” contas
- Escola “contém” alunos
- Pode ser avançado, mas mantenha esse bit ligado. Agregação deve ter preferência sobre Herança.
- Herança PODE se tornar tão complexa que fica difícil de ser alterar/corrigir.
- Artigo muito interessante:
<https://www.thoughtworks.com/pt/insights/blog/composition-vs-inheritance-how-choose>

Agregação - Exemplo

```
1 package org.basic.java;  
2  
3 public class Carro {  
4  
5     private Porta portaMotorista;  
6  
7     ...  
8 }
```

Herança/Agregação - Exercícios

- Codifique o exemplo de herança do “Lanche”, mas em linguagem JAVA
- Codifique um exemplo de agregação pensando num relógio de parede, como o da figura abaixo:



Revisão

- Aprendemos sobre um conceito fundamental de OO que é herança. Isso serve para compartilhar, de maneira eficiente, atributos e comportamentos, na relação “É-UM”
- Vimos que agregação serve para “compor” classes a fim de criar uma nova classe que é a foto da agregação/composição de partes menores