



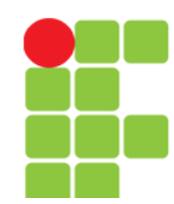
Herança

Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro
Prof. Edwar Saliba Júnior
Dezembro de 2019



Reutilização / Reuso

- São palavras-chave no conceito de herança;
- Reutilizar um código que já tenha sido escrito, e que já esteja sendo utilizado por um outro *software* é muito bom, pois:
 - evita-se “reinventar a roda”;
 - economiza-se tempo de desenvolvimento;
 - aumenta-se a qualidade do *software*, pois, partes reutilizadas já foram testadas e depuradas e
 - diminui-se o custo para produção do *software*.

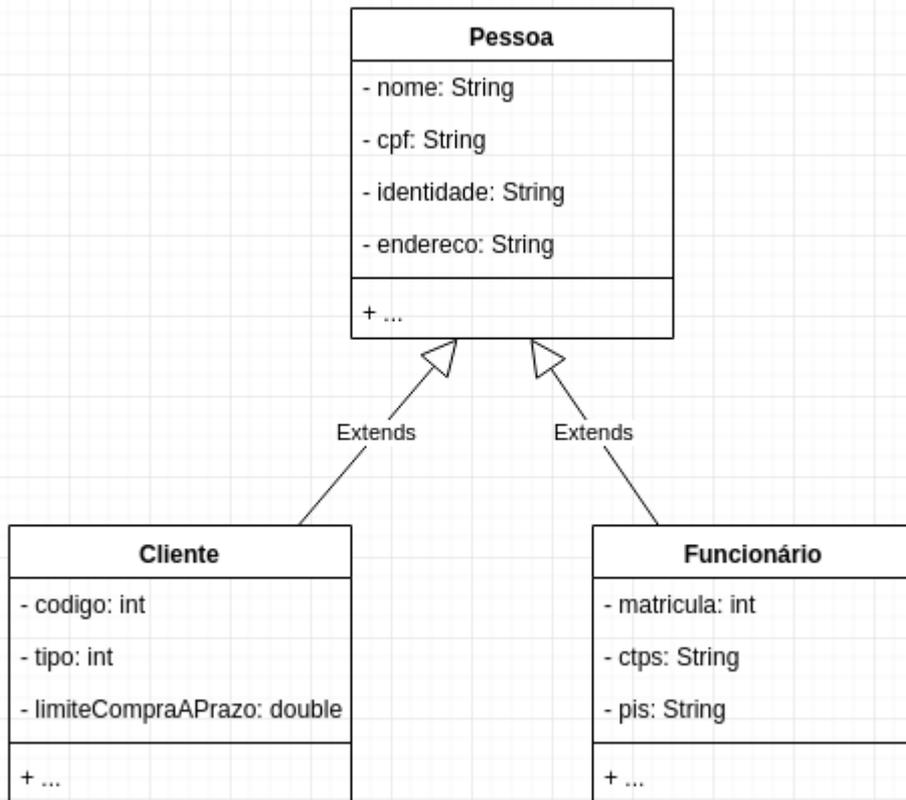


Herança

- É um recurso que permite que novas classes sejam definidas a partir de classes já existentes;
- Na UML representa o relacionamento do tipo “é um”;
- Na hierarquia de classes:
 - **Superclasse (ou ascendente)**: são as ascendentes de um classe;
 - **Subclasse (ou descendente)**: são as descendentes de uma classe;
 - **Classe Mãe**: é a ascendente direta de uma classe e
 - **Classe Filha**: é a descendente direta de uma classe.



Herança



- **Pessoa** é superclasse de **Funcionário** e de **Cliente**;
- **Funcionário** e **Cliente** são subclasses de **Pessoa**;
- **Funcionário** e **Cliente** herdam as definições da classe **Pessoa**. Ou seja, **herdam atributos e métodos da superclasse.**

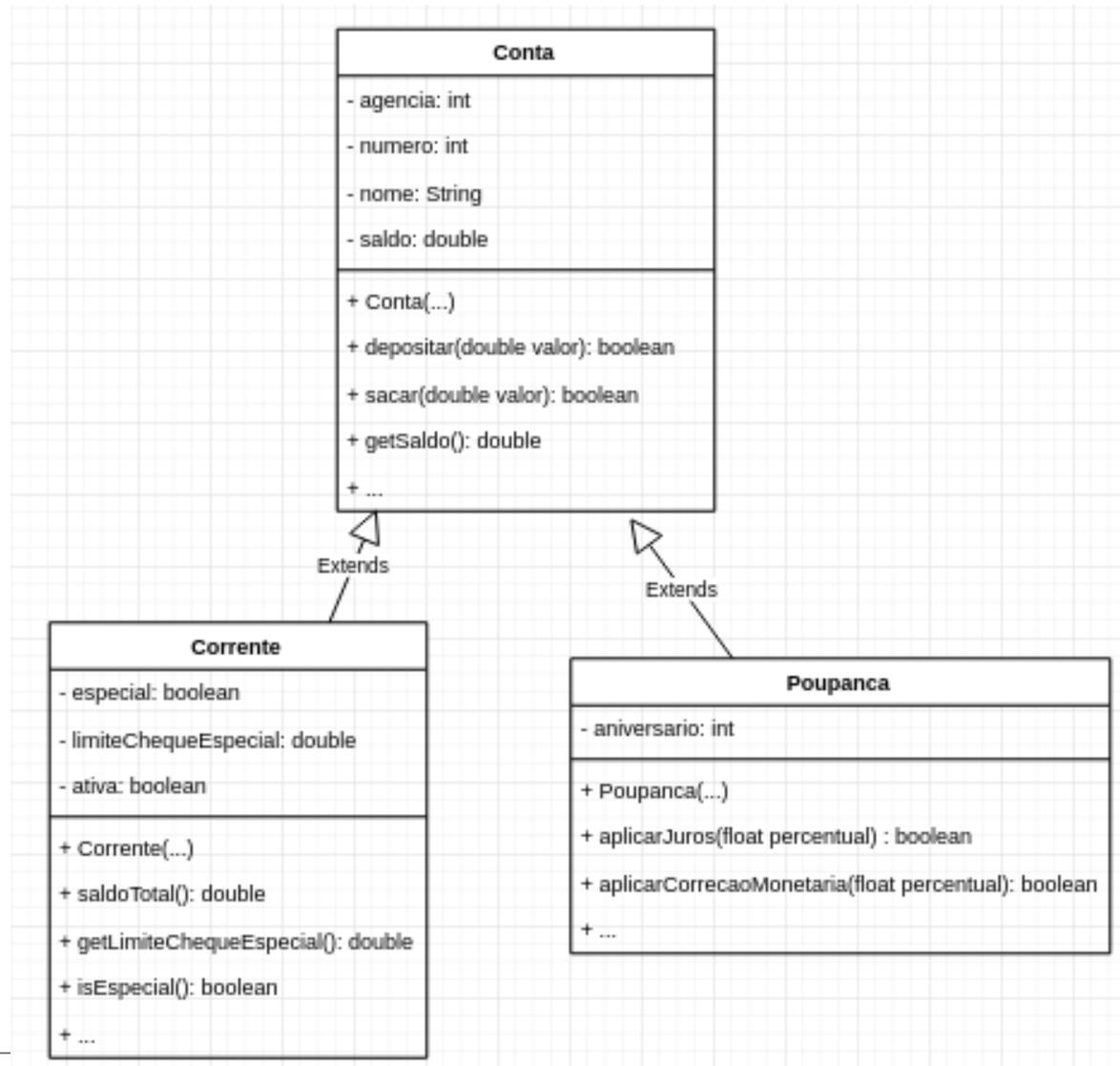


Herança

- Pensemos em outros exemplos de herança!



Herança





Herança

- **Corrente** possui como atributos:

`especial,`

`limiteChequeEspecial` e

`contaEspecial.`

possui também:

`agência,`

`número,`

`nome` e

`saldo,`

pois, herda estes últimos da classe **Conta**;

- **Corrente** possui os métodos: `saldoTotal`, `getLimiteChequeEspecial` e `isEspecial`. Possui também: `depositar`, `sacar` e `getSaldo`, sendo que estes três últimos são herdados da classe **Conta**.



Como Criar Herança em Java

- A palavra chave **extends** indica herança em Java;
- Exemplo:

```
public class Corrente extends Conta
```

- A linha acima indica que a classe **Corrente** herda os atributos e métodos da classe **Conta**.



Modificador de Acesso

- Nos relacionamentos de herança, o modificador de acesso:
 - ***public***: indica que o atributo ou método é visível nas subclasses;
 - ***private***: indica que o atributo ou método não é visível nas subclasses e
 - ***protected***: indica que o atributo ou método é visível e, no caso dos atributos, alteráveis pela subclasse.



Chamada de Métodos da Superclasse

- Para chamar o método construtor da superclasse, usa-se a seguinte linha de comando na subclasse:

```
super(lista de parâmetros);
```

- Redefinição de métodos: se um método é implementado na subclasse B com a mesma assinatura de um método existente na superclasse A, então, diz-se que o método foi redefinido. Neste caso, o método que será executado para um objeto da classe B será aquele definido na classe B.



Exemplo de Herança em Java

- Exemplo 01 ([pdf](#))
- Exemplo 02 ([pdf](#))



Exercício

- Escreva um *software* que possua as seguintes classes: Veículo, Carro, Caminhão e Moto (onde estas três últimas são subclasses de Veículo). Seu *software* deverá ser capaz de armazenar 100 veículos de cada um dos três subtipos especificados. Construa uma estrutura de *menus* onde o usuário do *software* possa facilmente manipular este cadastro.



Bibliografia

- DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**; tradução Edson Furmankiewicz; revisão técnica Fábio Lucchini. 6a. ed., São Paulo: Pearson, 2005.
- FERREIRA, Kecia Aline Marques. *Slides* da disciplina de Programação de Computadores II. CEFET-MG, 2009.