

# Herança

Prof. Edwar Saliba Júnior  
Fevereiro de 2011

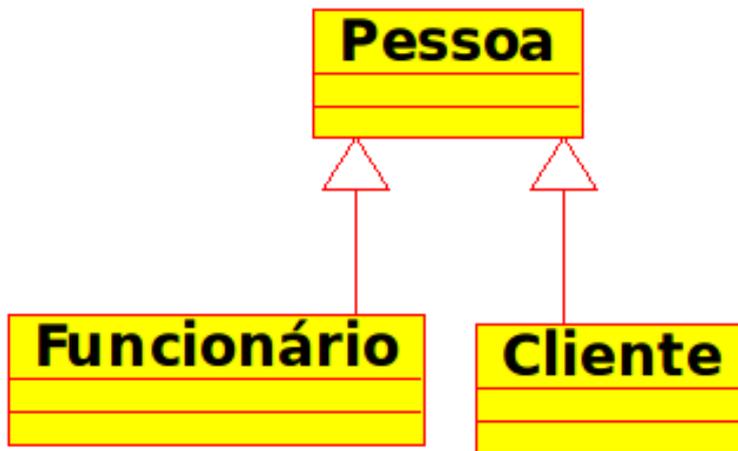
# Reutilização / Reuso

- 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;
  - Diminui-se o custo para produção do *software*.

# Herança

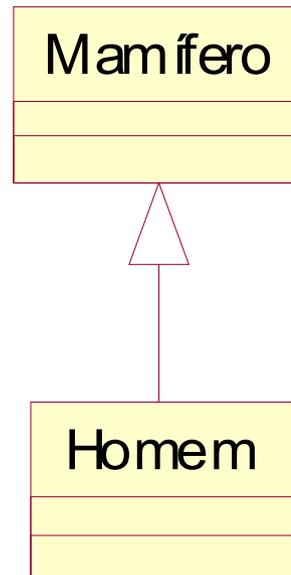
- É um recurso que permite que novas classes sejam definidas a partir de classes já existentes;
- Na UML representa o relacionamento do tipo “é um”;
- Na hierarquia de classes:
  - **Superclasse (ou ascendente)**: são as ascendentes de um classe;
  - **Subclasse (ou descendente)**: são as descendentes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

# Herança

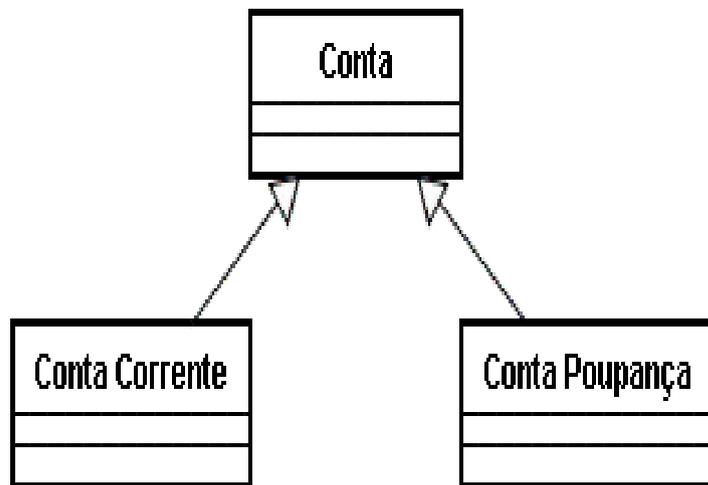


- Mamífero é superclasse de Homem.
- Homem é subclasse de Mamífero.
- Homem herda definições da classe Mamífero.

# Herança

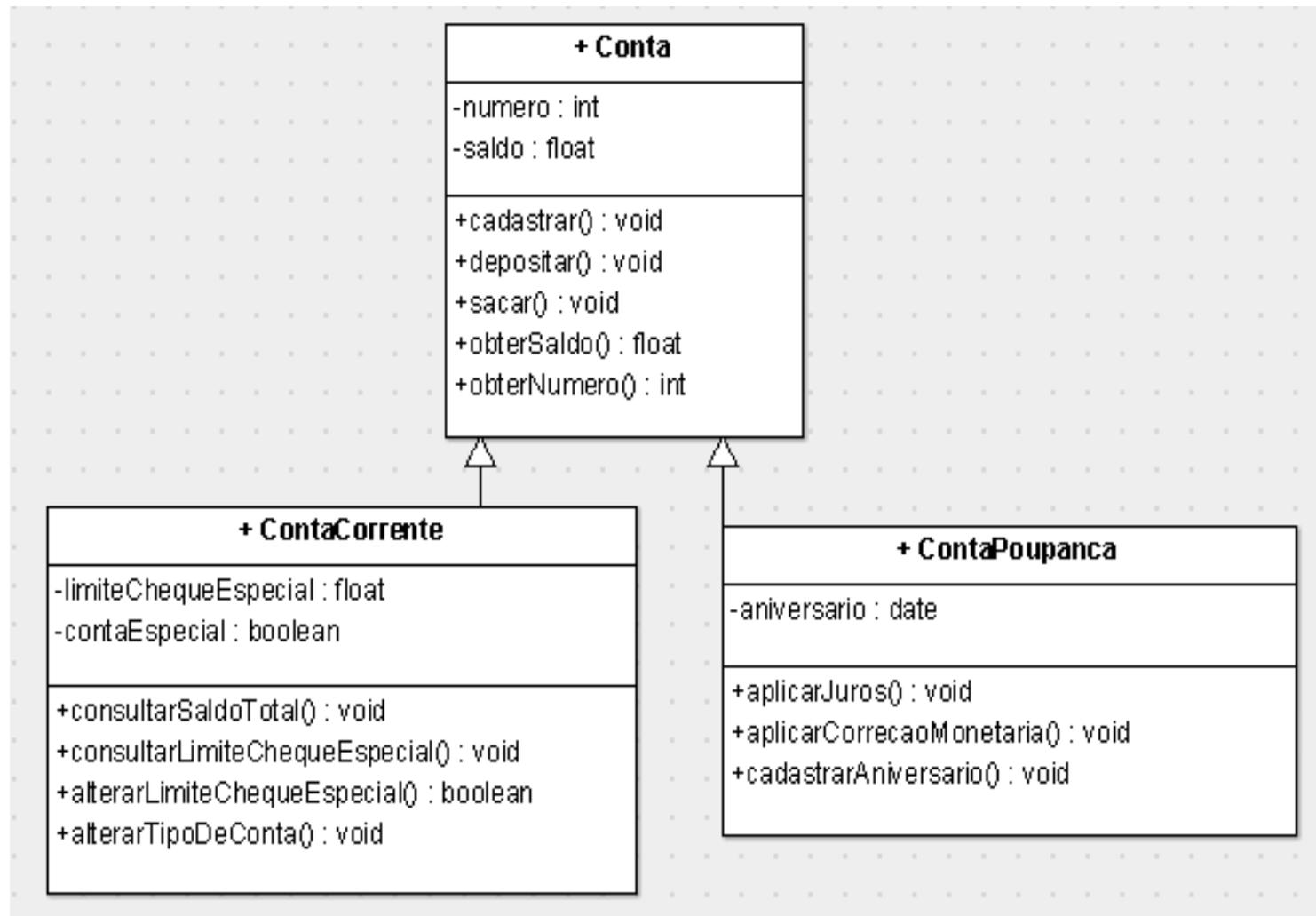
- Pensemos em outros exemplos de herança!

# Veja o Exemplo



- Aqui:
  - Conta é superclasse de Conta Corrente e de Conta Poupança;
  - Conta Corrente e Conta Poupança são subclasses de Conta;
- Significa que:
  - Conta Corrente herda características (atributos) e comportamentos (métodos) de Conta;
  - Conta Poupança que também herda características (atributos) e comportamentos (métodos) de Conta.

# Herança



# Herança

- *Conta Corrente* possui como atributos: *limiteChequeEspecial* e *contaEspecial*. Possui também: *número* e *saldo*, pois, herda estes dois atributos da classe *Conta*;
- *Conta Corrente* possui como métodos: *consultarSaldoTotal*, *consultarLimiteChequeEspecial*, *alterarLimiteChequeEspecial* e *alterarTipoDeConta*. Possui também: *cadastrar*, *depositar*, *sacar*, *obterSaldo* e *obterNumero*, sendo que estes cinco últimos são herdados da classe *Conta*.

# Herança

- *Conta Poupança* possui como atributos: *aniversário*, *número* e *saldo*, pois, herda estes dois últimos da classe *Conta*;
- *Conta Poupança* possui como métodos: *aplicarJuros*, *aplicarCorrecaoMonetaria* e *cadastrarAniversario*. Possui também: *cadastrar*, *depositar*, *sacar*, *obterSaldo* e *obterNumero*, sendo que estes cinco últimos são herdados da classe *Conta*.

# Como Criar Herança em Java

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

```
public class ContaCorrente extends Conta
```

- A linha acima indica que a classe `ContaCorrente` 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;
  - ***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, 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 ([odt](#) || [pdf](#))
- Exemplo 02 ([odt](#) || [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 10 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.

Fim