

# Programação Orientada a Objetos

Thiago Nunes de Sousa  
thiagonunes.tns@gmail.com



# Atributos de Classe

- **Atributos estáticos**
  - static
- **Não pertencem aos objetos, mas sim à classe**
  - Pode ser acessados por todos os objetos
- **Não podem ser chamados com this**
  - Pois não pertencem ao objeto!



# Métodos de Classe

- **Métodos estáticos**
  - static
- **Não pertencem aos objetos, mas sim à classe**
  - Pode ser acessados por todos os objetos
- **Não podem ser chamados com this**
  - Pois não pertencem ao objeto!
- **Não manipulam atributos de objeto**



# Static

- **Exemplo Eclipse**

- Contar quantas contas foram criadas
- Criar um atributo estático, incrementá-lo automaticamente e criar um método para recuperá-lo (get)
- Criar uma classe de validação e um método estático para validação de CPF
- Alterar a classe cliente para validar o CPF com este método



# Herança

- **Problema**

- Nosso sistema gerenciador de contas precisa manter agora contas correntes e contas poupanças
- Precisamos agora de uma classe para cada tipo de conta



# Herança

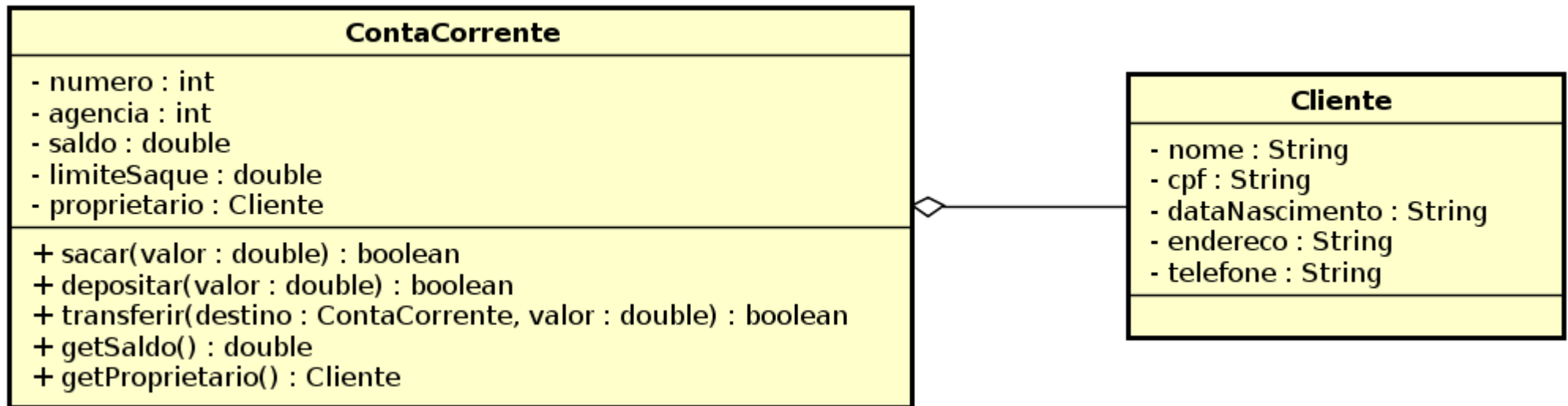
- **Conta Corrente**

- O que se faz com uma conta corrente?
  - Sacar dinheiro
  - Depositar dinheiro
  - Imprimir Proprietário
  - Consultar Saldo
  - Transferir dinheiro para outra conta



# Herança

## • Classe Conta



# Herança

- **Conta Poupança**

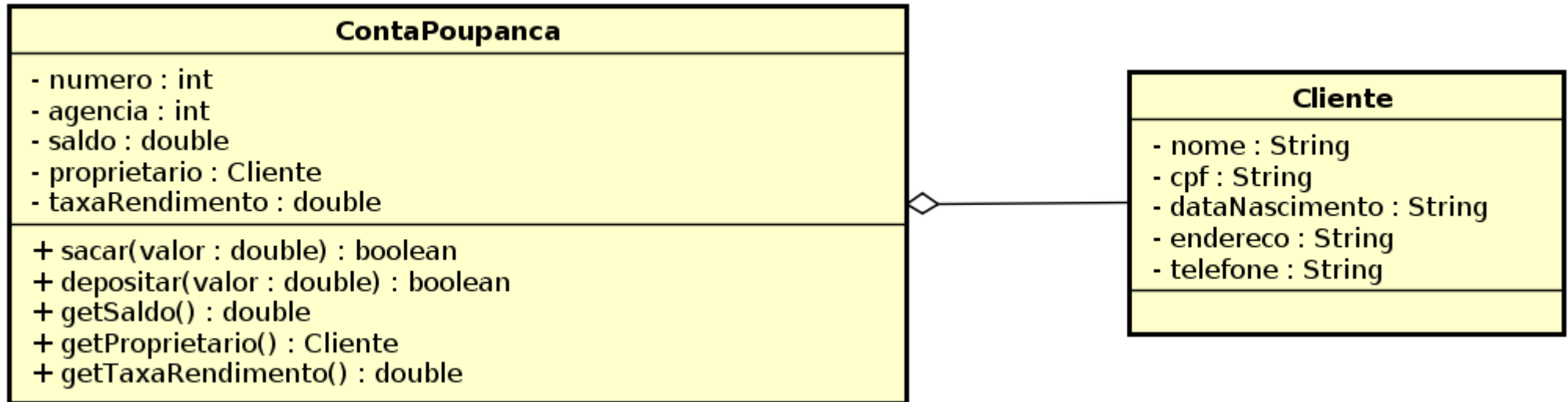
- O que se faz com uma conta corrente?
  - Saca dinheiro (não tem limite para saque)
  - Deposita dinheiro
  - Imprime Proprietário
  - Consulta Saldo
  - Tem Rendimento mensal





# Herança

## • Classe Conta Poupança



# Herança

- **Precisamos replicar as informações de número da conta, agência e proprietário nas duas classes ?**
- **Precisamos criar um método para depositar em cada um das classes ?**
- **Se precisarmos adicionar um atributo novo às nossas contas, precisaremos replicá-lo nas duas classes ?**



# Herança

- **Solução**

- Generalização da Classe Conta
  - Juntar todos os atributos e métodos repetidos em uma única classe
- Criar as classes especialistas ContaCorrente e ContaPoupança
  - Herdam todos os atributos e métodos da classe Conta
  - Podem ter seus atributos e métodos específicos



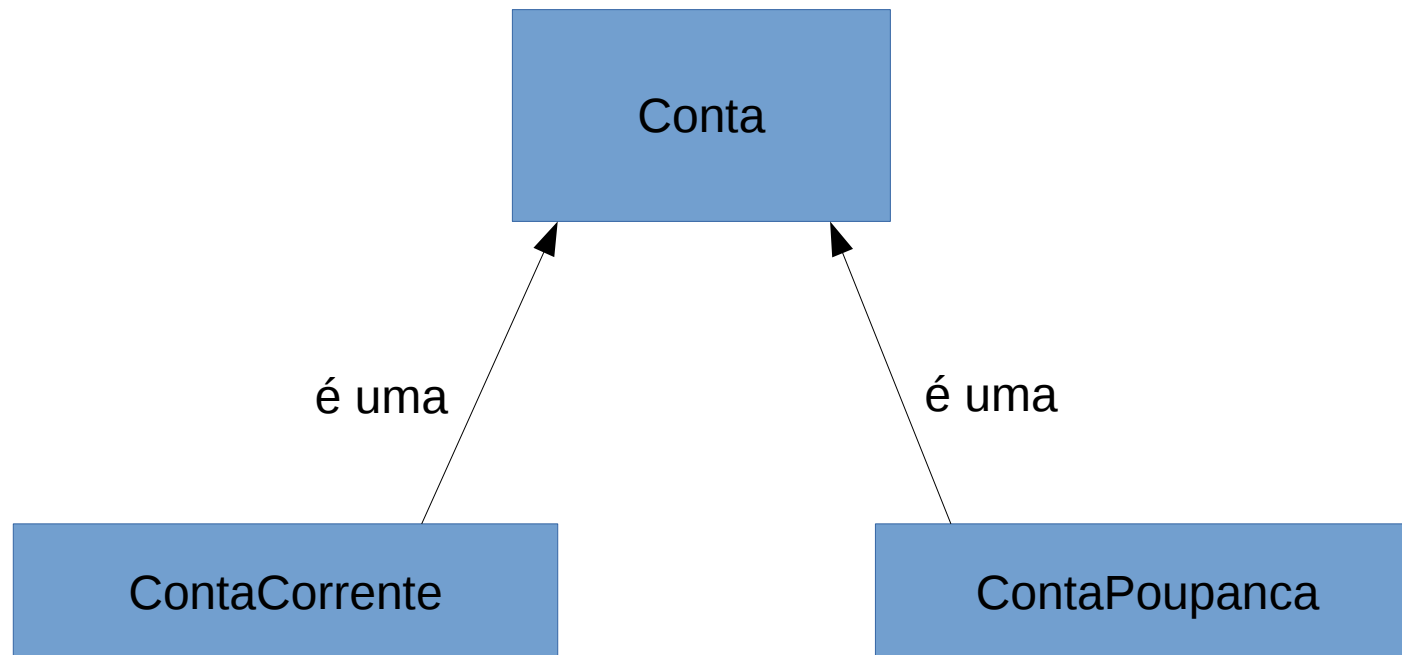
# Herança

- **Solução**

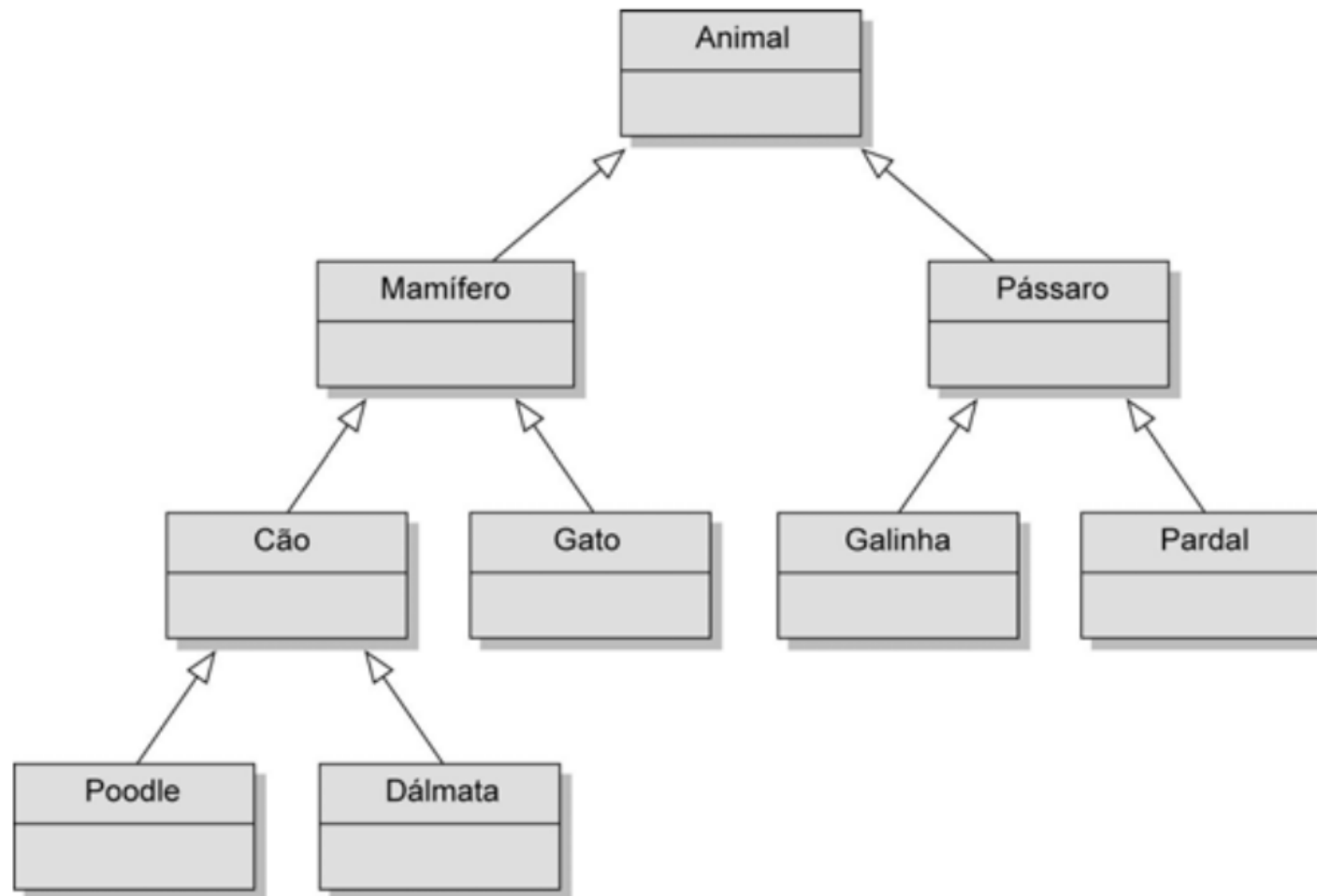
- Criação de uma classe Mãe
  - **Conta**
- Criação de duas classes filhas
  - ContaCorrente
  - ContaPoupanca



# Herança



# Herança



# Herança em Java

- **Class ContaCorrente extends Conta {...}**
- **Class ContaPoupanca extends Conta {...}**



# Herança

- **Eclipse ;)**

- Criar classes Conta, ContaCorrente e ContaPoupanca
- Implementar os métodos específicos
- Testar





# Sobrescrita de Método

- **Problema**

- Os saques da conta corrente têm um limite
- Os saques da conta poupança **não**.

- **Como resolver?**



# Sobrescrita de Método

- Criar um novo método na classe Conta Poupança ?
- Chamá-lo sacarPoupanca ?
- **NÃO!**



# Sobrescrita

- **Solução**

- Sobrescrever o método “sacar” na classe ContaPoupanca

- **Como?**

- Eclipse ;)
  - Implementar o método reescrito e testar

