

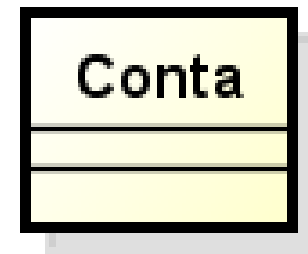


# Orientação a Objetos

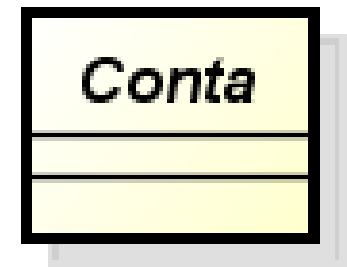
Diagrama de Classes

- Um diagrama de classes descreve:
  - Os tipos de objetos presentes no sistema;
  - Os vários tipos de relacionamentos estáticos existentes entre os objetos.
- Um diagrama de classes também mostra:
  - As propriedades de cada classe;
  - As operações de cada classe;
  - As restrições que se aplicam à maneira como os objetos estão conectados.

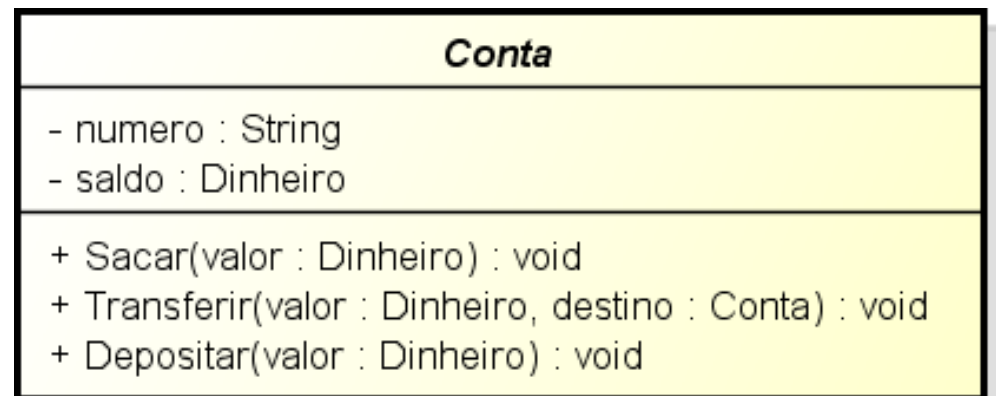
- Representam os tipos de objetos existentes no modelo.
- Descritas a partir de seus atributos, operações e restrições.
- Podem ser organizadas segundo uma estrutura de generalização/especialização.
- Classes podem ser identificadas a partir das especificações de casos de uso, sendo normalmente associadas a substantivos.



- São classes que não podem ser instanciadas diretamente.
- Possuem seu nome representado em itálico.

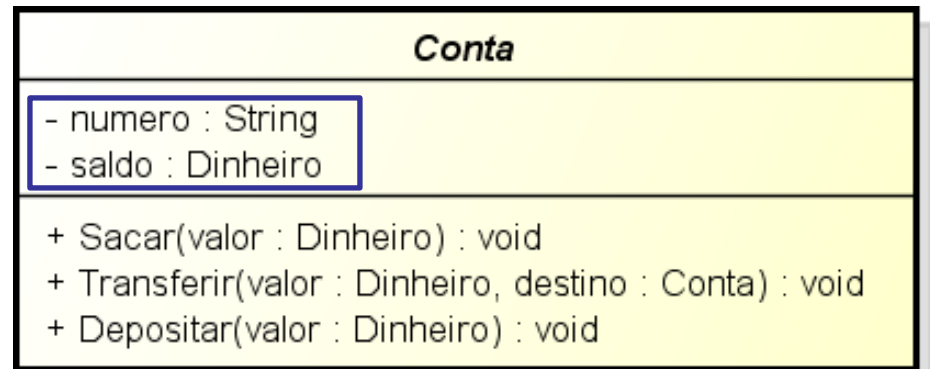


A UML utiliza a palavra característica como um termo geral que cobre as propriedades e operações de uma classe.



- As propriedades representam as características estruturais de uma classe.
- As propriedades são um conceito simples, mas elas aparecem em duas notações bastante distintas: atributos e associações.
  - Embora pareçam bastante diferentes em um diagrama, na realidade, tratam-se da mesma coisa.

- A notação de um atributo descreve uma propriedade como uma linha de texto dentro da caixa de classe em si.



powered by Astah

- Exemplo de notação de atributo:
  - nome: String [1] = “Sem nome” { readonly }

## ● Forma completa de um atributo:

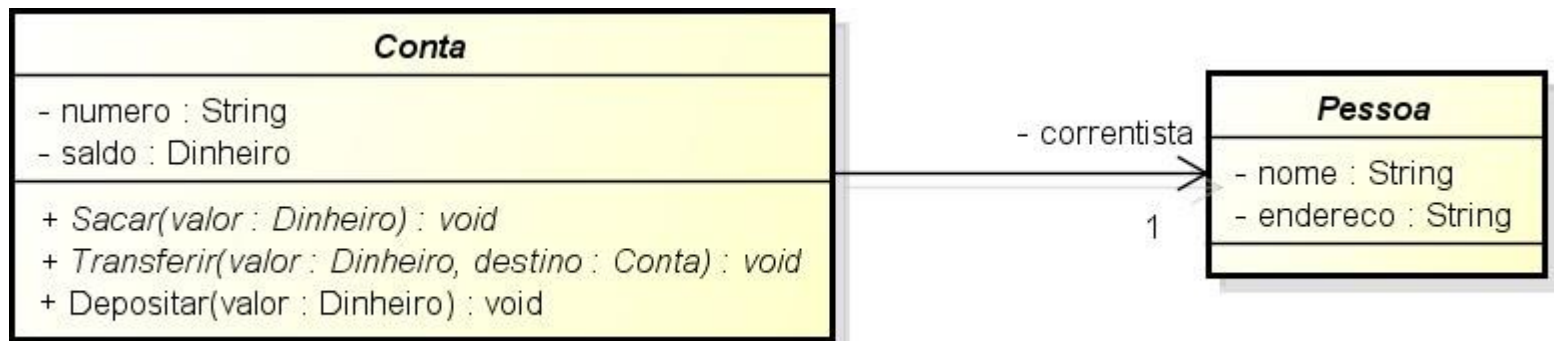
visibilidade nome: tipo multiplicidade = valor-por-omissão {lista de propriedades}

- Somente o nome é necessário.
- Visibilidade:
  - (+): público;
  - (-): privado;
  - (#): protegido.
- Tipo:
  - Restrição sobre o tipo de objeto que pode ser colocado no atributo.
- Multiplicidade:
  - Exemplos de multiplicidades: 1, 0..1, \*
- Valor-por-omissão:
  - Valor do atributo quando criado, que é mantido enquanto o atributo não recebe um novo valor.
- Lista de propriedades:
  - Permite indicar propriedades adicionais para o atributo.



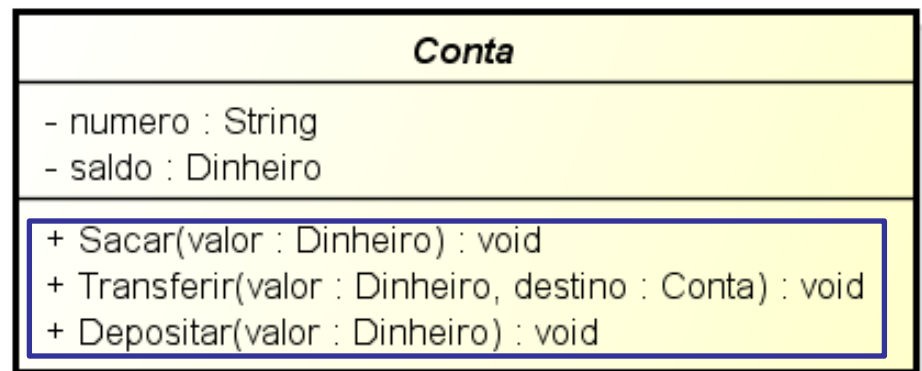
- Outra maneira de anotar uma propriedade é como uma associação.
- Uma associação é uma linha cheia entre duas classes, direcionada da classe de origem para a classe de destino.

- O nome da propriedade fica no destino final da associação, junto com a multiplicidade.
- O destino final da associação é a classe que é o tipo da propriedade.



- Geralmente, atributos são utilizados para coisas pequenas, como datas, valores booleanos, números etc.
- Associações geralmente são usadas para classes mais significativas, como clientes e pedidos.
- Essa escolha está muito mais relacionada à ênfase do que a qualquer significado subjacente.

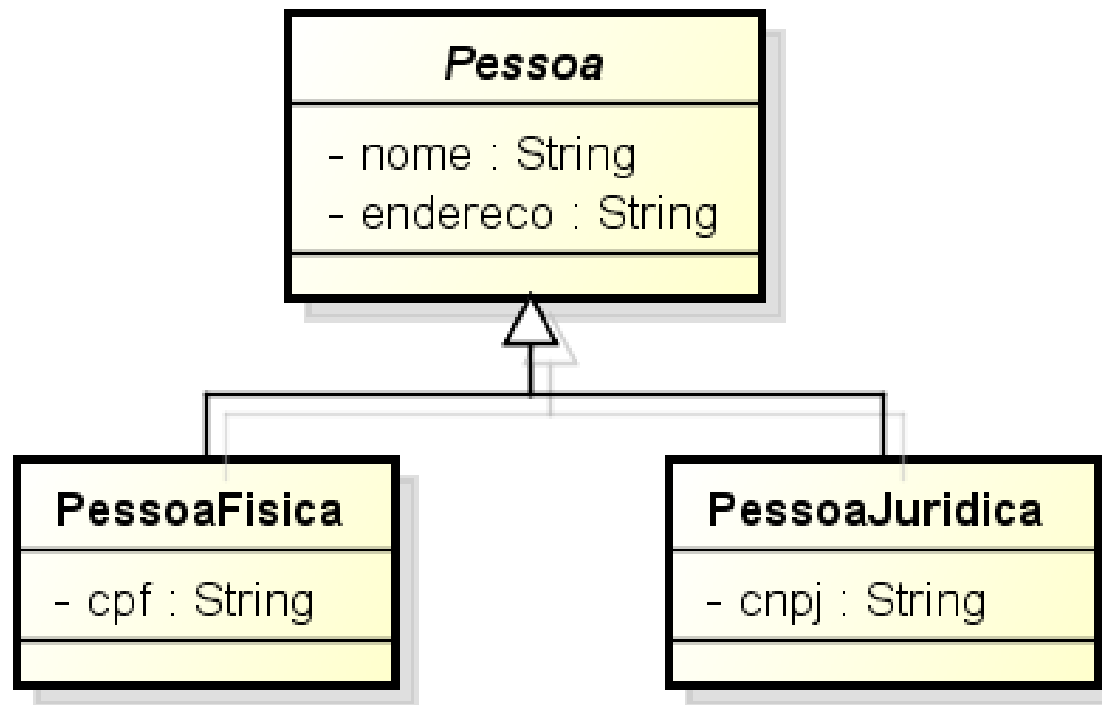
- Operações são as ações que uma classe sabe realizar.
- As operações correspondem aos métodos presentes em uma classe.
- Normalmente, as operações simples de manipulação de atributos (get e set) não são mostradas, pois essas operações, na maioria das vezes, podem ser inferidas.



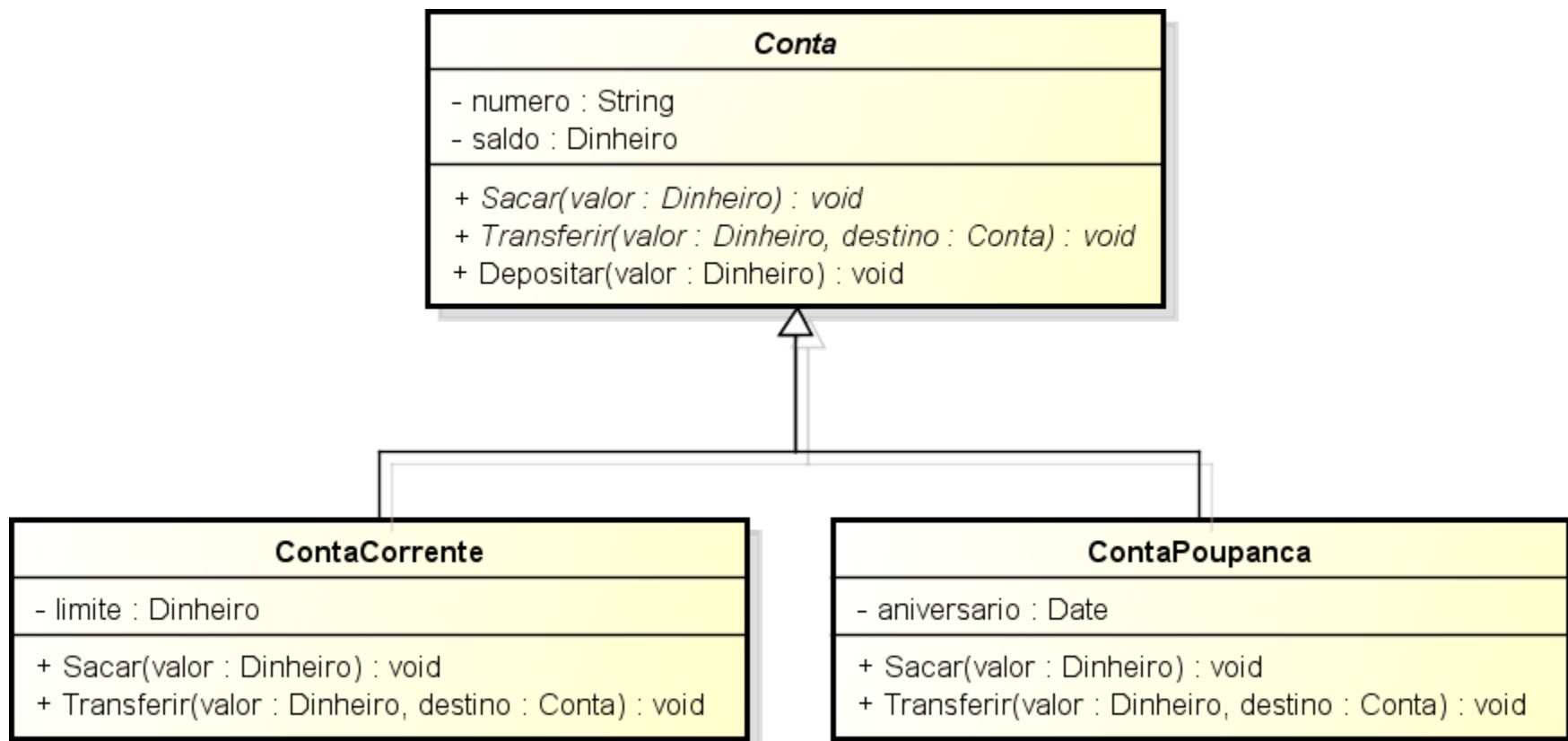
- A sintaxe completa da UML para operações é:  
visibilidade nome {lista-de-parâmetros}: tipo-de-retorno {lista-de-propriedades}
- Visibilidade:
  - (+): público;
  - (-): privado;
  - (#): protegido.
- Nome:
  - Sequência de caracteres que dá nome à operação.
- Lista-de-parâmetros:
  - É a lista de parâmetros da operação.
- Tipo-de-retorno:
  - É o tipo do valor retornado, se houver um.
- Lista-de-propriedades:
  - Permite indicar propriedades adicionais para o atributo.
- Exemplo:
  - + retornarSaldoEm (data: Date): Dinheiro

- Um exemplo típico de generalização é o que envolve pessoas físicas e jurídicas.
  - Elas têm diferenças mas também possuem muitas semelhanças.
  - As semelhanças podem ser colocadas em uma superclasse geral chamada Pessoa.
  - As diferenças de cada subclasse são colocadas em suas respectivas subclasses.

- Todos os atributos e operações da superclasse são herdados pela subclasse.
- Uma subclasse pode sobrepor nenhum, um ou todos os métodos de sua superclasse.







- Substituição: dentro de qualquer código que utilize um objeto cujo tipo é definido em uma superclasse, um objeto cujo tipo é definido por uma subclasse da superclasse supracitada pode ser utilizado em substituição ao objeto do tipo da superclasse.
- Exemplo: se a classe Conta possui uma associação com a classe Pessoa denominada “correntista”, então a propriedade “correntista” pode receber tanto objetos do tipo PessoaFisica quanto objetos do tipo PessoaJuridica.

