



## Atributos

- ▶ **Atributo representa alguma propriedade do item que está sendo modelado. Um atributo, portanto, é uma abstração do tipo de dados ou estados que os objetos da classe podem abranger.**



Prof. Dr. Alexandre Cardoso



## Atributos

- ▶ **Compostos por 3 porções:**

Visibilidade NomeDoAtributo : TipoDeExpressao =  
ValorInicial {Propriedade}

•origin	Name only
•+ origin	Visibility and name
•origin : Point	Name and type
•head : *Item	Name and complex type
•name [0..1] : String	Name, multiplicity, and type
•origin : Point = (0,0)	Name, type, and initial value
•id : Integer {frozen}	Name and property



Prof. Dr. Alexandre Cardoso



## Formas das Propriedades

1. Changeable: não há restrições na modificação de valores do atributo
2. addOnly: para atributos com multiplicidade maior que 1, valores adicionais podem ser adicionados, mas, uma vez criados não podem ser removidos ou alterados
2. frozen: os valores dos atributos não podem ser modificados após o objeto ser inicializado.



Prof. Dr. Alexandre Cardoso



## Operações

- **Uma operação é a implementação de um serviço que pode ser solicitado por algum objeto da classe para modificar o comportamento. Ele é compartilhado por todos os outros objetos da classe.**



Prof. Dr. Alexandre Cardoso



# Operações

## ▸ Sintaxe em UML

[visibility] name [(parameter-list)] [: return-type] [{property-string}]

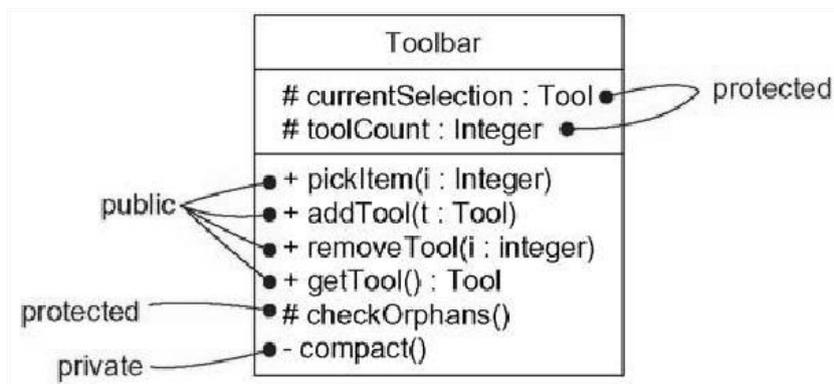
•display	Name only
•+ display	Visibility and name
•set (n : Name, s : String)	Name and parameters
•getID() : Integer	Name and return type
•restart() {guarded}	Name and property



Prof. Dr. Alexandre Cardoso



# Visibilidades - UML



Prof. Dr. Alexandre Cardoso



## Visibilidades - *private*

- ▶ Um método *private* não pode ser chamado a partir de outro módulo (classe).
- ▶ Um atributo ou propriedade *private* não pode ser lido ou escrito a partir de outro módulo.
- ▶ A declaração *private*, torna uma propriedade ou um método exclusivo à classe, os quais pertencem.



Prof. Dr. Alexandre Cardoso



## Visibilidades - *protected*

- ▶ Referenciando um método e/ou atributo como *protected*: informa-se que, os métodos serão chamados e os atributos poderão ser lidos e/ou escritos na superclasse, quanto em suas subclasses, poderão fazer acesso direto aos métodos e atributos da superclasse.



Prof. Dr. Alexandre Cardoso



## Visibilidade - *public*

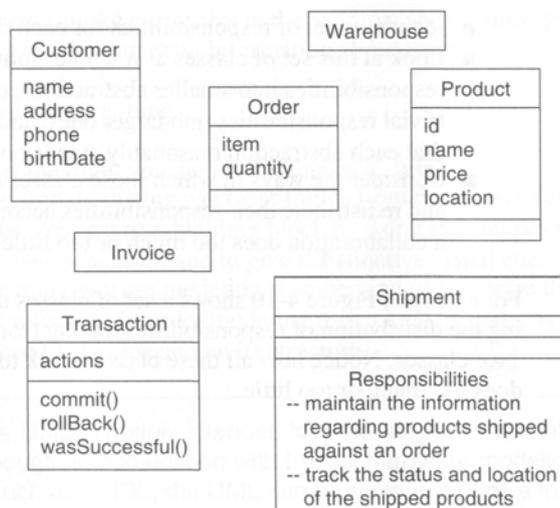
- ▶ Qualificar como *public*: estamos informando que tanto atributos, quanto métodos e propriedades estarão acessíveis em qualquer lugar onde houver uma referência à classe.



Prof. Dr. Alexandre Cardoso



## Diagramas sem Associações



Prof. Dr. Alexandre Cardoso



## D. Classes : Dicas

---

---

- ▶ Uma classe bem modelada deve :
  - Prover uma abstração clara de algo desenhado do vocabulário do sistema ou do domínio da solução.
  - Incorporar um pequeno e bem definido conjunto de responsabilidades e implementá-las muito bem.
  - Prover uma distinção clara entre especificação da abstração e implementação.
  - Ser entendível, simples e ainda assim extensível e adaptável



Prof. Dr. Alexandre Cardoso



## D. Classes: Dicas

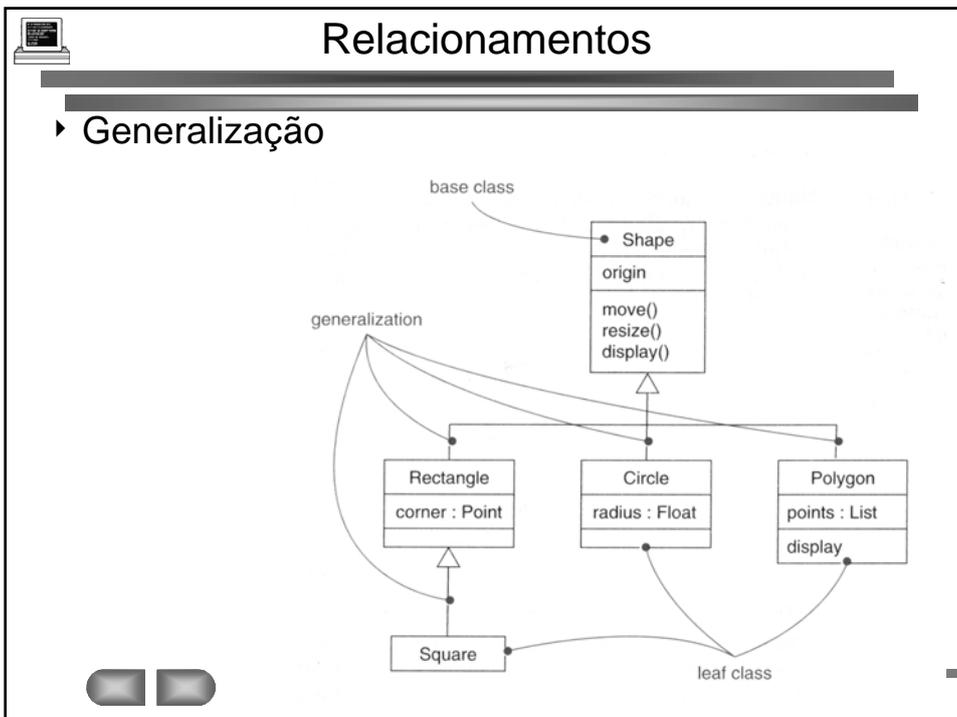
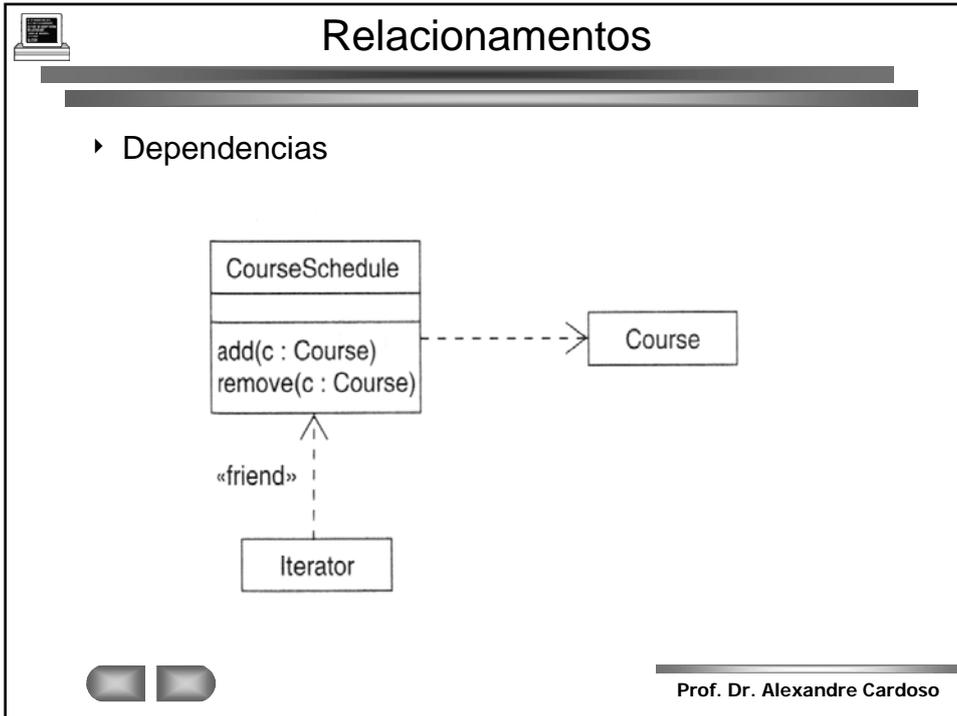
---

---

- ▶ Ao desenhar o diagrama de classes:
  - Mostre apenas as propriedades de classe que são importantes para a abstração e entendimento do contexto.
  - Organize listas de atributos e operações de acordo com agrupando por categoria.
  - Mostra classes relacionadas no mesmo diagrama.

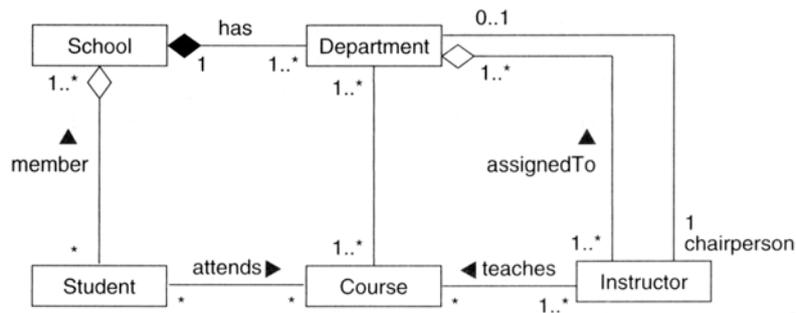


Prof. Dr. Alexandre Cardoso



## Relacionamentos

### ▸ Associações e agregações



Prof. Dr. Alexandre Cardoso

## Relacionamentos: Dicas

### ▸ Ao modelar relacionamentos:

- Use dependências apenas se o relacionamento não for estrutural.
- Generalização = “é tipo de”.
- Normalmente herança múltipla pode ser transformada em agregações.
- Cuidado com herança cíclica.
- Mantenha sua árvore de generalização balanceada.
- Associações indicam relacionamentos estruturais entre objetos.

Prof. Dr. Alexandre Cardoso

Relacionamentos: Dicas

- ▶ Ao desenhar relacionamentos:
  - Linhas retas x Linhas obliquas
  - Evite cruzar linhas
  - Evite desenhos de relacionamentos supérfluos

Prof. Dr. Alexandre Cardoso

Mecanismos Comuns

The diagram illustrates three common UML mechanisms:

- stereotype:** A box representing a subsystem node with the text «subsystem» Billing {version = 3.2}. A dot on the left is labeled "stereotype" and a dot on the right is labeled "tagged value".
- constraint:** A line connecting a "Server" node and a "Building 1 Hub" node. The line is labeled "constraint" and has the text {> 10M/Sec line} written next to it.
- stereotyped node:** Both the "Server" and "Building 1 Hub" nodes are labeled as "stereotyped node".

Prof. Dr. Alexandre Cardoso

 **Mecanismos Comuns: Notas**

---

simple text

Publish this component  
in the project repository  
after the next design review.  
*egb 1/5/98*

embedded

See <http://www.gamelan.com>  
for an example of this applet.

link to document

 See encrypt.doc for  
details about this algorithm.

**Prof. Dr. Alexandre Cardoso**

 **Mecanismos Comuns: Outros Adornos**

---

Transaction

---

addAction()  
removeAction()  
perform()  
rollback()

---

Exceptions

emptyTransaction  
noSuchAction  
resourceLocked

Client

---

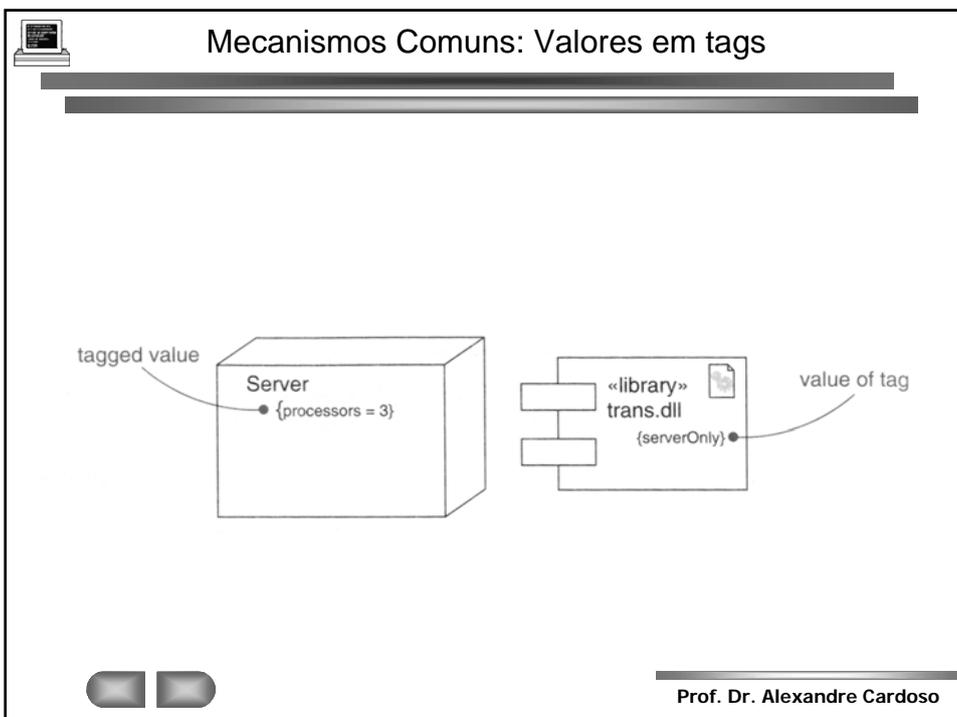
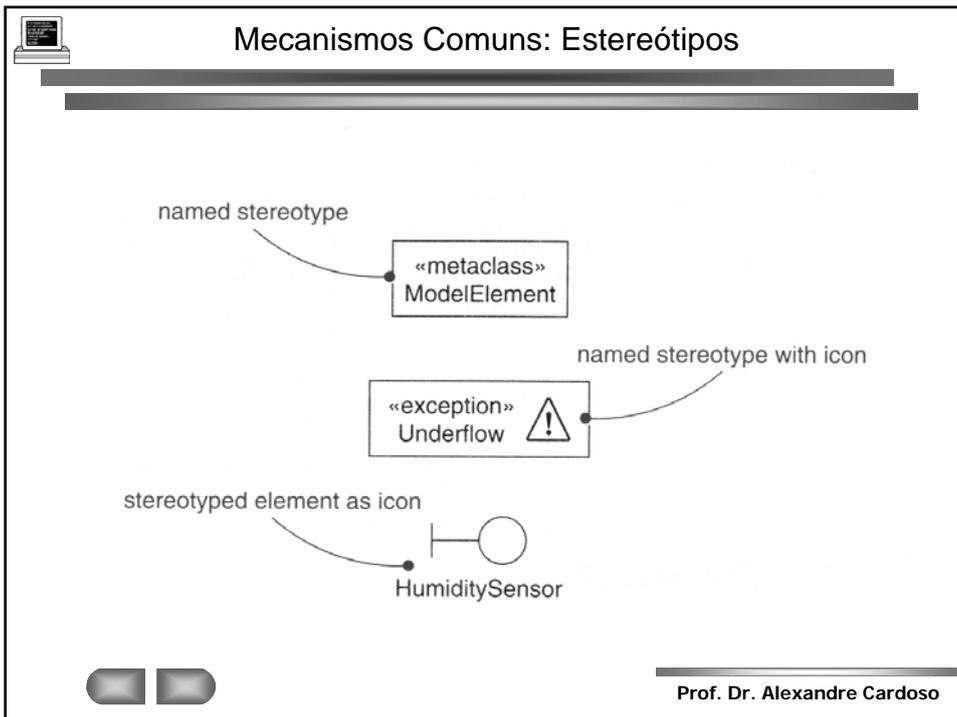
bill.exe  
report.exe  
contacts.exe

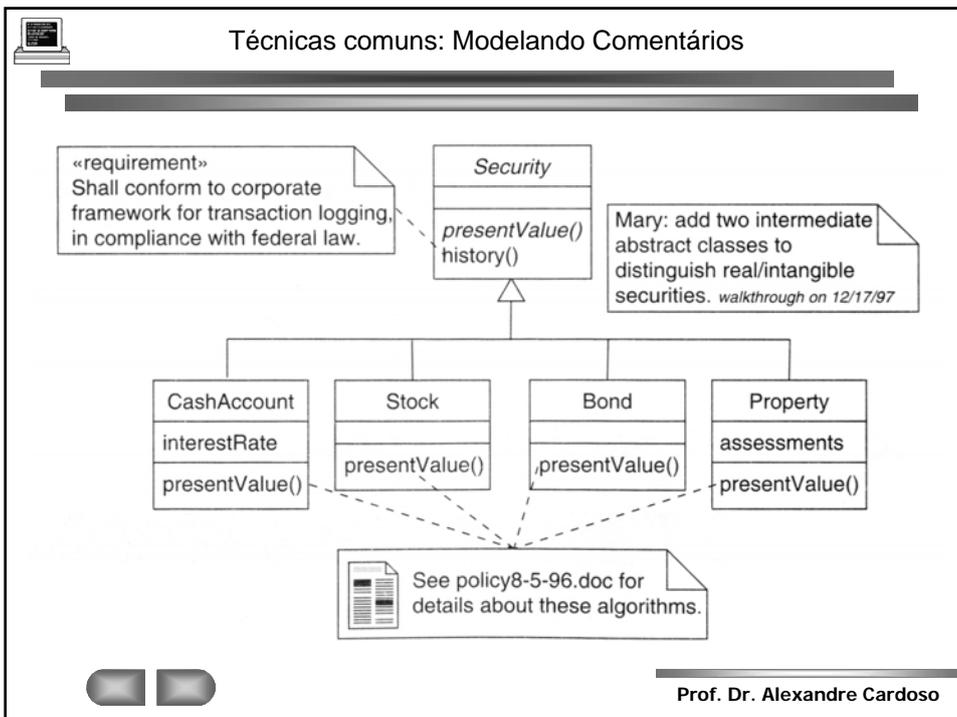
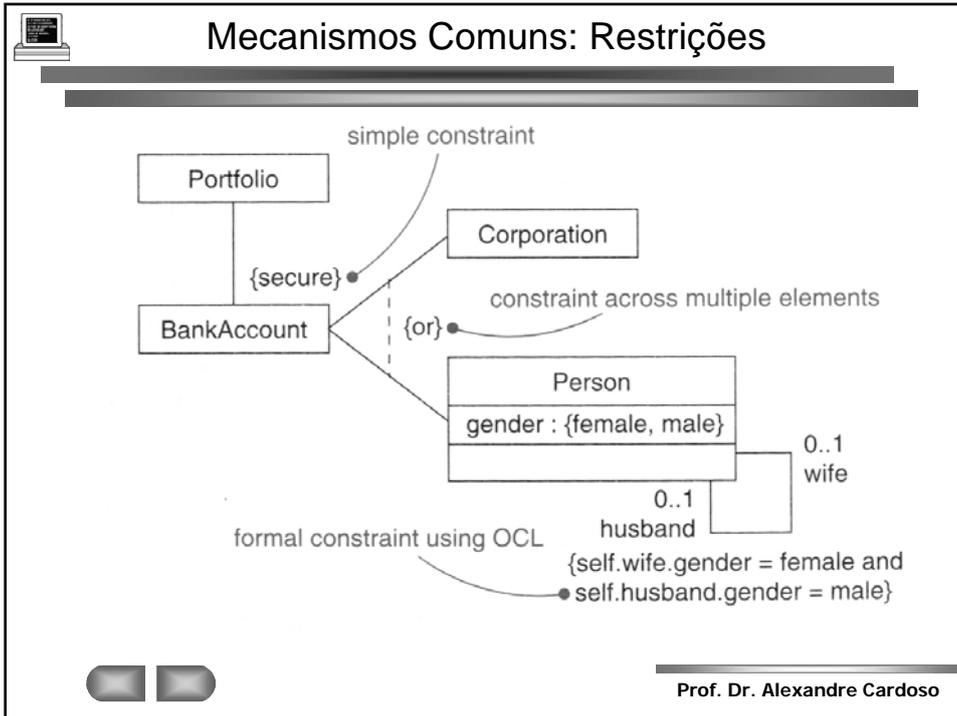
anonymous compartment

named compartment

**Prof. Dr. Alexandre Cardoso**







## Mecanismos comuns: Dicas

- ▶ Ao adornar o modelo com notas:
  - Use notas apenas para requisitos, observações, opções e explicações que não podem ser feitas usando outros recursos da UML.
  - Pensem em notas como “post-it”s eletrônicos.
- ▶ Ao desenhar notas:
  - Ao invés de lotar o modelo de comentários longos, crie um anexo e referencie-o na nota.



Prof. Dr. Alexandre Cardoso



## Mecanismos comuns: Dicas

- ▶ Ao melhorar um modelo com estereótipos, valores em tags ou restrições:
  - Padronize as extensões usadas no projeto.
  - Escolha nomes curtos e significativos.
  - Se a precisão for flexível, use restrições como texto livre. Senão use a OCL.
- ▶ Ao desenhar:
  - Use estereótipos gráficos apenas quando necessário.
  - Considere sobreposição e cores como estereótipos, em adição aos ícones.



Prof. Dr. Alexandre Cardoso

