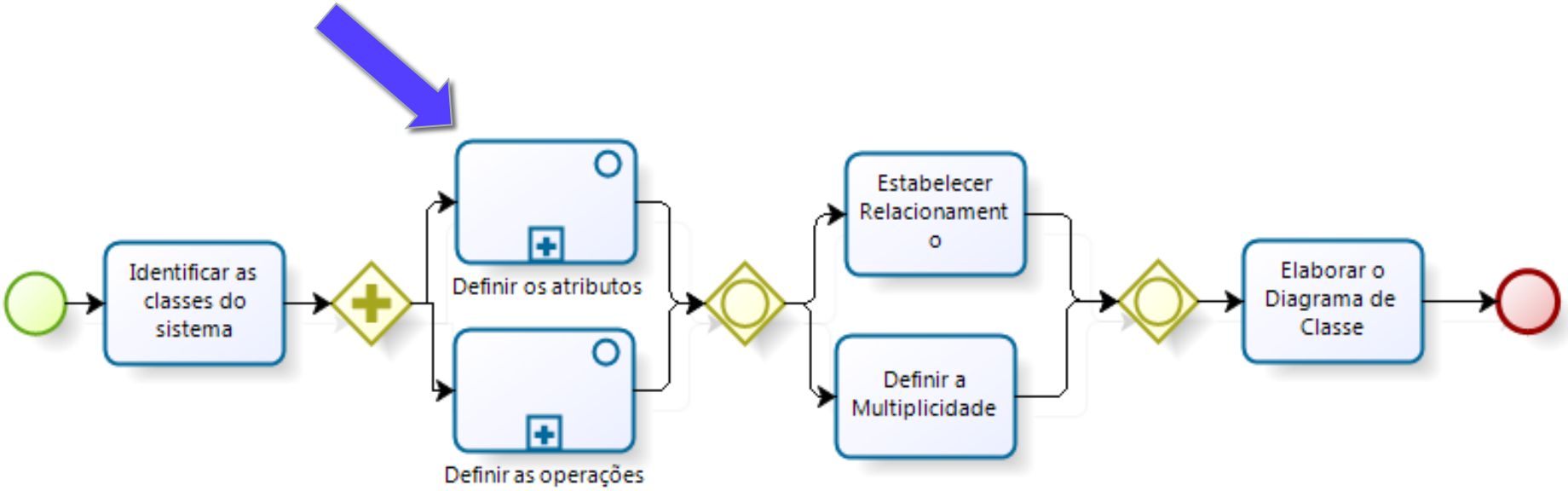
DIAGRAMA DE CLASSE

Prof. Sergio Akio Tanaka
sergio.tanaka@unifil.br

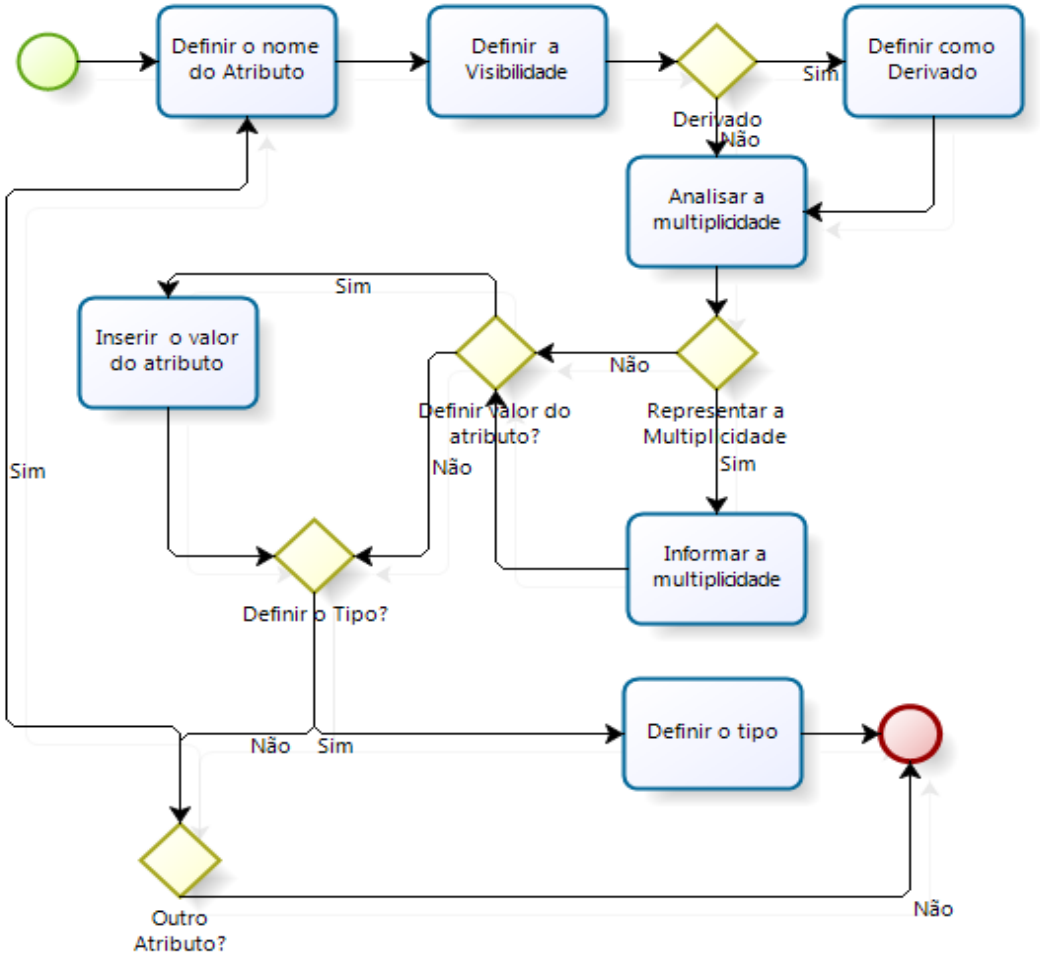
Diagrama de Classe

- Mostra um conjunto de classes, interfaces e colaborações e seus relacionamentos.
- Os diagramas de classes abrangem a visão estática do projeto de um sistema.

Workflow do Diagrama de Classe



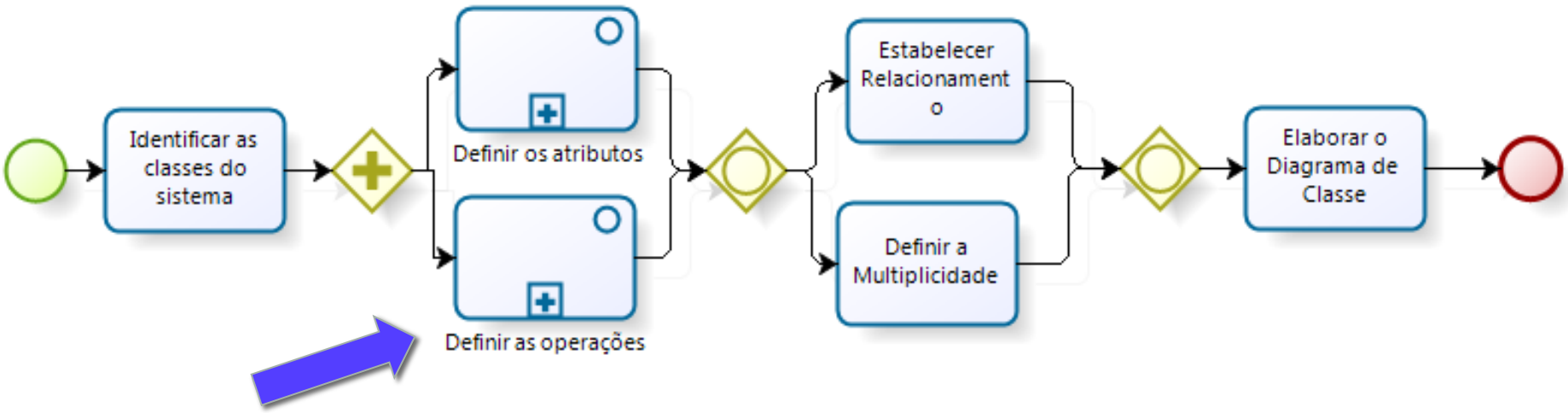
Passo a passo para Definir o Atributo



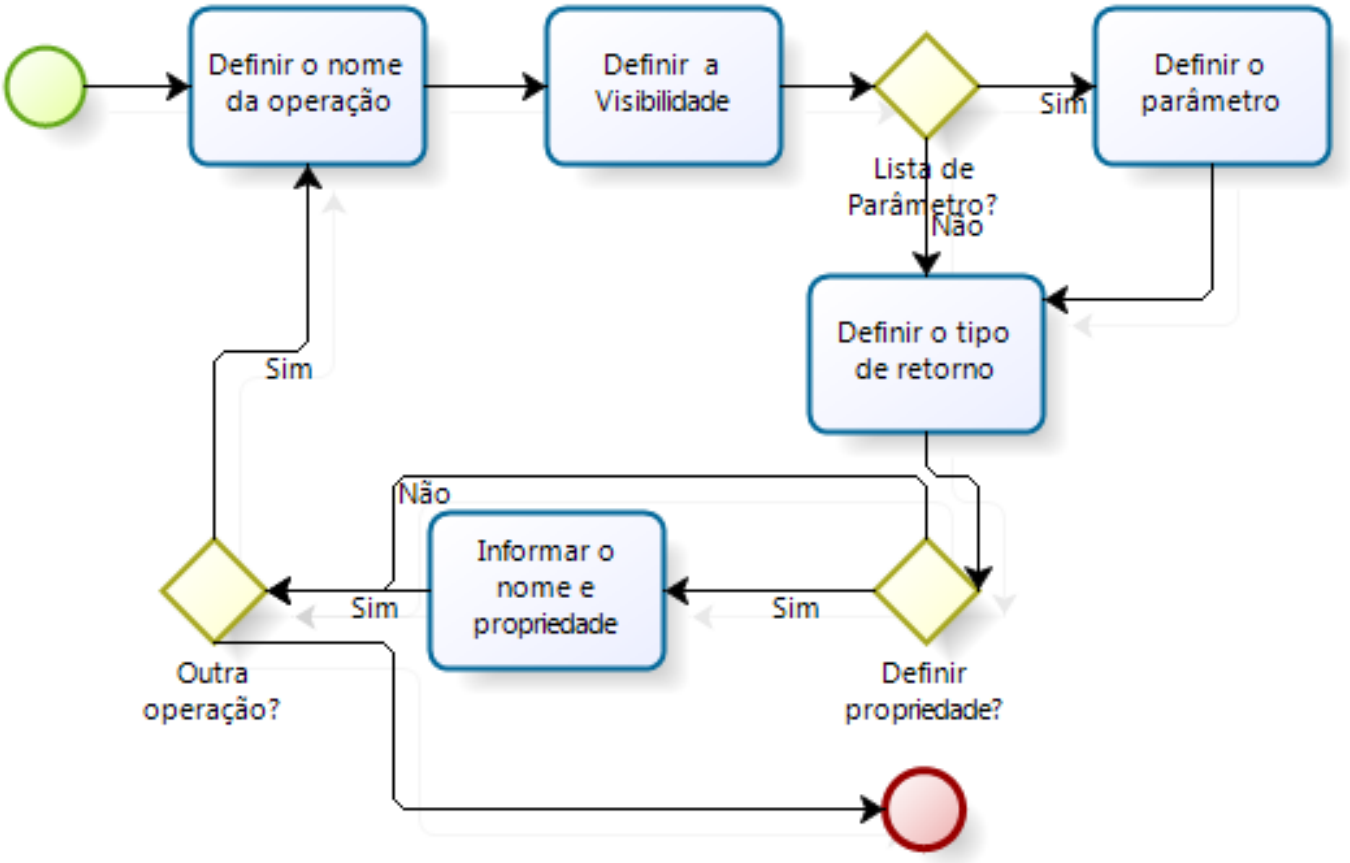
[visibilidade] [/] nome-do-atributo : [tipo] [multiplicidade] [=default] [{string_propriedade}]

-Nome: String=no default, {Required, 1..30 carac, espaços e pontuação permitidos}

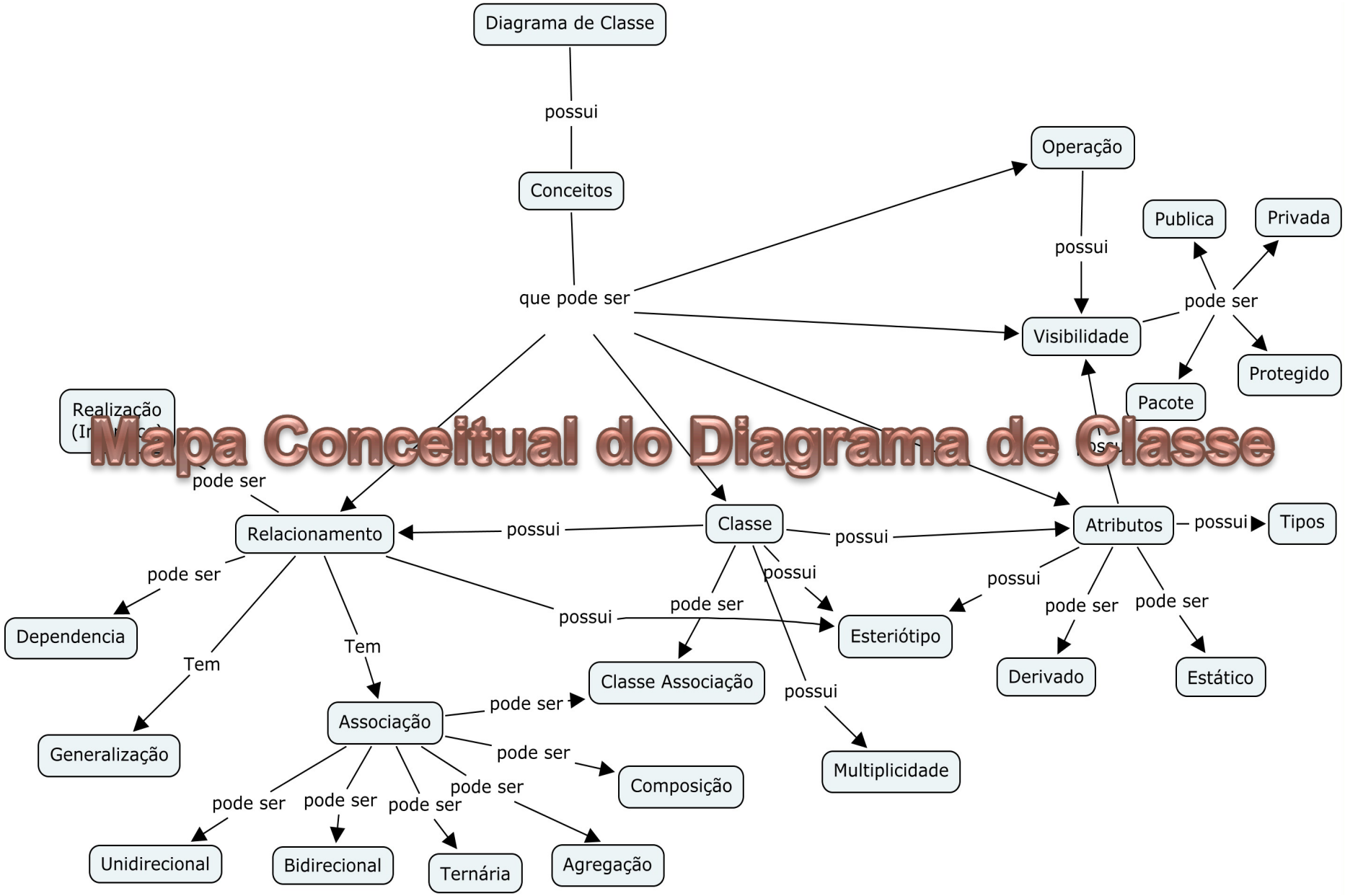
Definindo as Operações



Passo a Passo para Definir a Operação



[visibilidade] nome ([lista-parâmetro]) “:” [resultado-retorno] [(propriedades)]


Mapa Conceitual do Diagrama de Classe
Diagrama de Classe
possui
Conceitos
que pode ser
Operação
possui
Visibilidade
pode ser
Publica
Privada
Protegido
Pacote
Realização
pode ser
Relacionamento
possui
Classe
possui
Atributos
possui
Tipos
pode ser
Dependencia
possui
possui
pode ser
possui
Esteriótipo
pode ser
pode ser
Derivado
Estático
Tem
Tem
Classe Associação
possui
Generalização
Associação
pode ser
pode ser
Composição
Multiplicidade
pode ser
pode ser
pode ser
pode ser
Unidirecional
Bidirecional
Ternária
Agregação

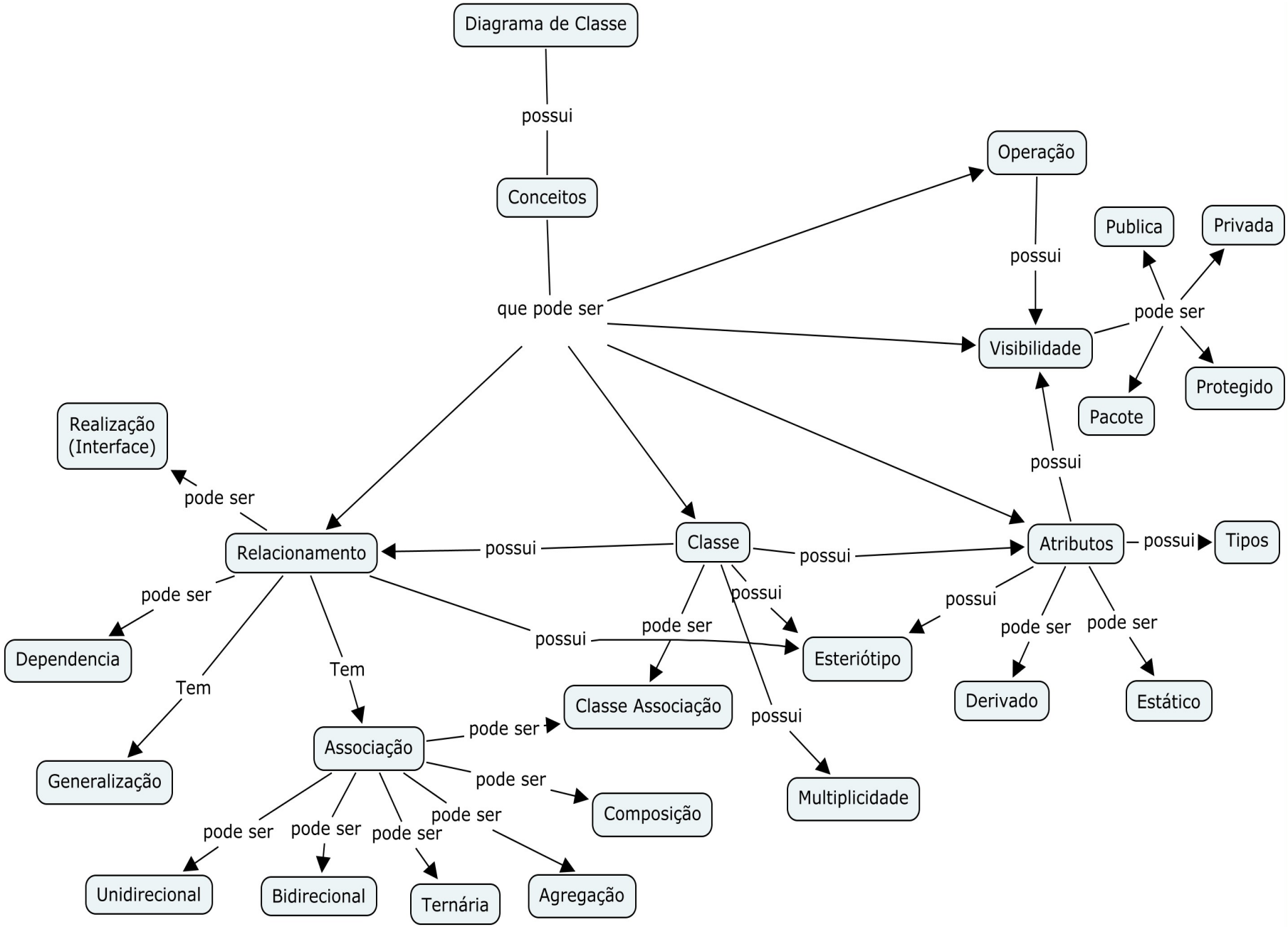


Diagrama de Classe
possui
Conceitos
que pode ser
Operação
possui
Visibilidade
pode ser
Publica
Privada
Protegido
Pacote
possui
Atributos
possui
Tipos
Classe
possui
Relacionamento
possui
Realização (Interface)
pode ser
Dependencia
pode ser
Tem
Generalização
Tem
Associação
pode ser
Classe Associação
pode ser
Composição
pode ser
Agregação
pode ser
Ternária
pode ser
Bidirecional
pode ser
Unidirecional
possui
Esteriótipo
pode ser
possui
Multiplicidade
possui
pode ser
Derivado
pode ser
Estático

Classe

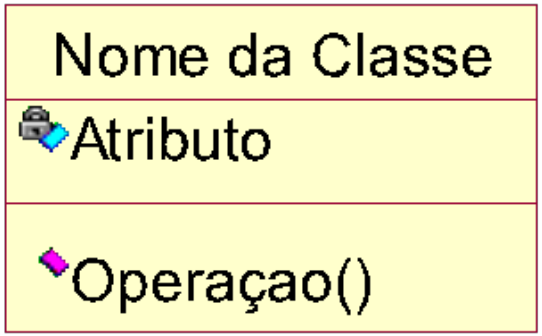
- As **classes** são os blocos de construção mais importante de qualquer sistema orientado a objetos.
- É uma descrição de um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos.

Classe

- Representa a abstração de um conjunto de OBJETOS do Mundo Real que possuem tipos de características e de comportamento em comum.

Classe Veículo

Notação da Classe



- A notação da classe é representada por um retângulo subdividido em 3 partes.
- Nome da Classe
- Atributo
- Operação

Atributos

- Representam as características de uma classe, ou seja, as peculiaridades que costumam variar de um objeto para outro

  - **Derivado:** seus valores são representados por algum tipo de calculo. Nestes casos o atributo é representado por uma / na frente do atributo

    /atributo

  - **Estático:** são atributos cujos valores são idênticos para todos os objetos de uma classe, ou seja é um atributo pertencente a classe propriamente dita. Sua identificação se dá através do nome da variável sublinhado

    atributo

Atributos

- Sintaxe do Atributo

- [visibilidade] [/] nome-do-atributo : [tipo] [multiplicidade] [=default] [{string_propriedade}]

- Ex.:

-Nome: String=no default, {Required, 1..30 carac, espaços e pontuação permitidos}

O nome, é privado, do tipo String, sem valores default, precisa ter de 1 a 30 caracteres, pode incluir espaços e pontuação.

Atributos

- **visibilidade**, que irá adotar um dos seguintes valores:
  - **público**: o atributo é visível dentro e fora da pacote que contém a classe;
  - **protegido**: o atributo é visível somente para a própria classe, para suas subclasses ou para amigos da classe (dependente da linguagem);
  - **privado**: o atributo é visível somente para a própria classe e para amigos da classe;
  - **pacote**: o atributo é visível somente para a próprio pacote.

- **/ Derivado:** seus valores são representados por algum tipo de cálculo. Nestes casos o atributo é representado por uma / na frente do atributo

- **nome**, que deve obedecer às convenções de nomenclatura da linguagem de implementação e do projeto;

Atributos

Sintaxe do Atributo

- **multiplicidade** especifica a quantidade valores que podem estar associados a um elemento do modelo

- **tipo**, que será um tipo de dado elementar suportado pela linguagem de implementação;

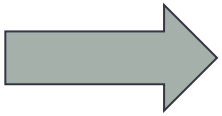
- **default - valor padrão ou inicial**, com o qual é inicializado quando novas instâncias da classe são criadas;

- **String_propriedade** O elemento propriedades permite acrescentar praticamente qualquer informação adicional sobre o atributo que não se encaixe em um dos elementos predefinidos

Operação

- Uma ação que o objeto executa, é uma ordem que faz o objeto a agir.
- A implementação de uma operação é chamada de **método**.

Operação



Operação

- Sintaxe da Operação
- [visibilidade] nome ([lista-parâmetro]) “:” [resultado-retorno] [(propriedades)]
- Visibilidade - Para cada operação, identifique a visibilidade da operação, podendo ser Publica, Protegida, Privada, Pacote.
- Nome – O nome da operação é Obrigatório

Operação - sintaxe

- Lista-parâmetro
- A lista-parâmetro é uma lista ordenada dos atributos, que, juntos, definem a entrada para uma operação.

- Resultado-retorno
- O resultado-retorno é a saída da operação.

- Propriedades
- O elemento propriedades permite acrescentar praticamente qualquer informação adicional sobre a operação que não se encaixe em um dos elementos predefinidos.

Relacionamento

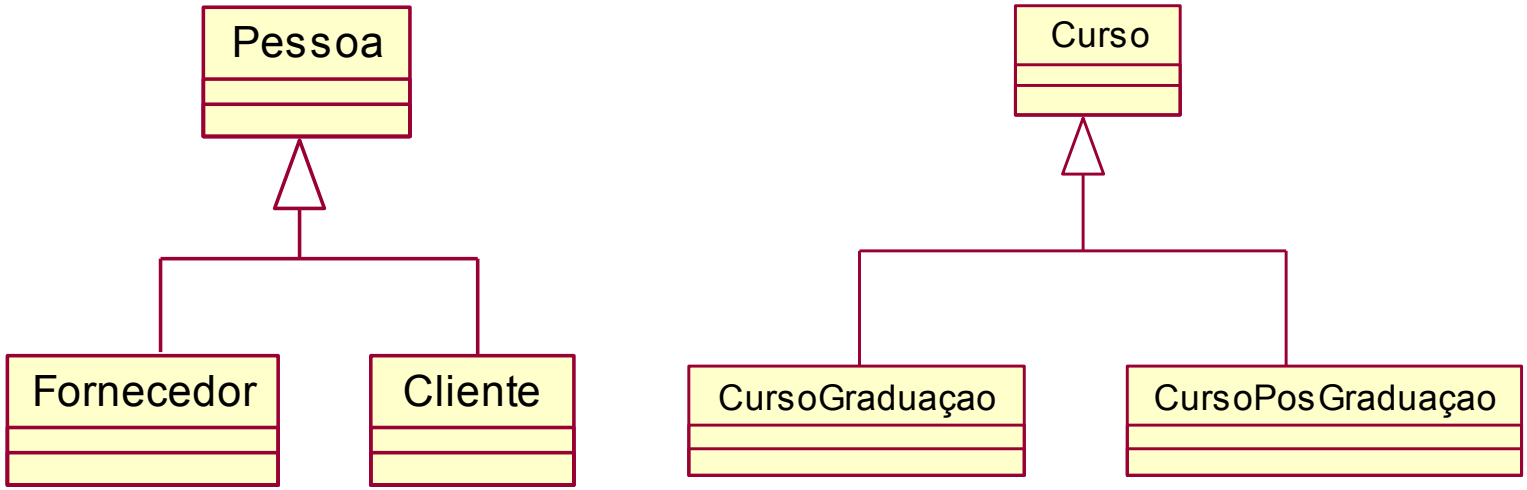
- Na UML, os modos pelos quais os itens podem estar conectados a outros, isto é, logicamente ou fisicamente, são modelados como relacionamentos, que permitem compartilhar informações e colaboram para a execução dos processos pelo sistema (GUEDES, 2005).

Associação Binária

- Ocorre quando são identificados relacionamentos entre duas classes.
- A associação é representada por uma linha reta entre as classes.

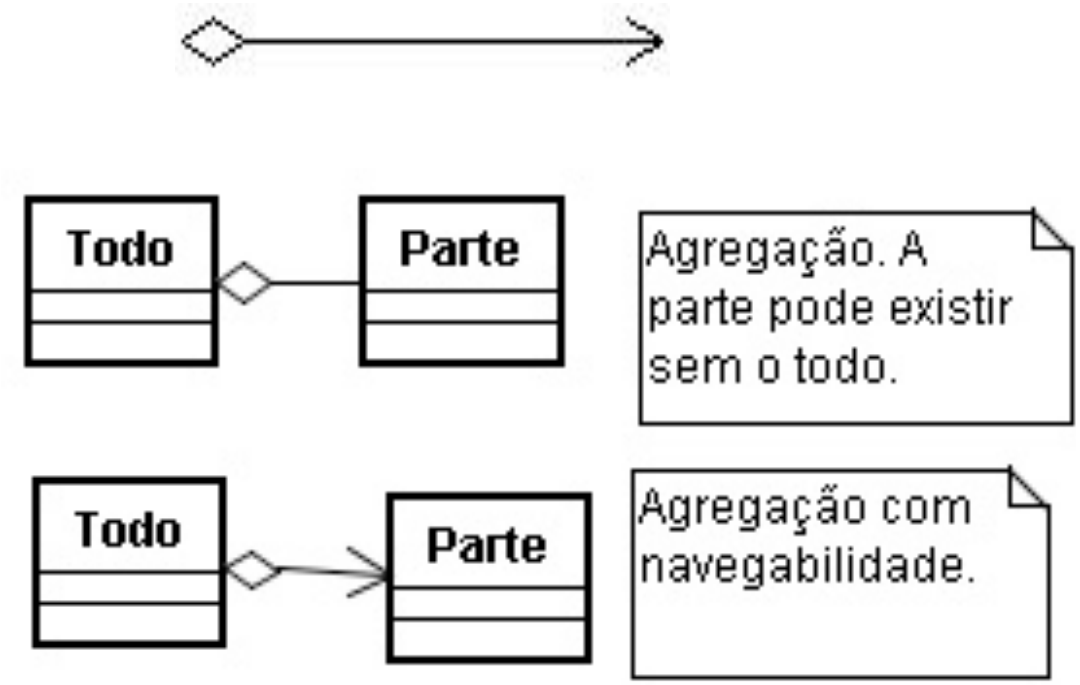
Generalização

- O objetivo dessa associação é representar a ocorrência de herança entre as classes, identificando as superclasses, chamadas gerais e subclasses, chamadas especializadas.
- Demonstrando a hierarquia entre as classes e possivelmente métodos polimórficos nas classes especializadas.



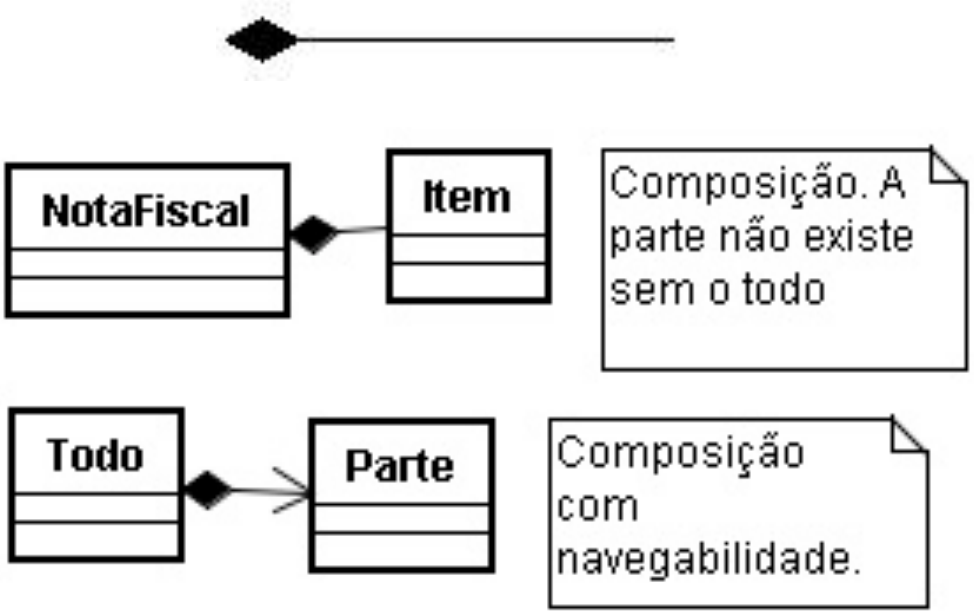
Agregação

- Esse tipo de associação tenta demonstrar uma relação todo/parte entre os objetos associados



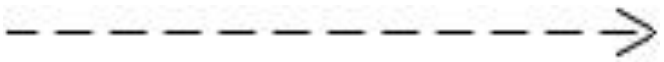
Composição

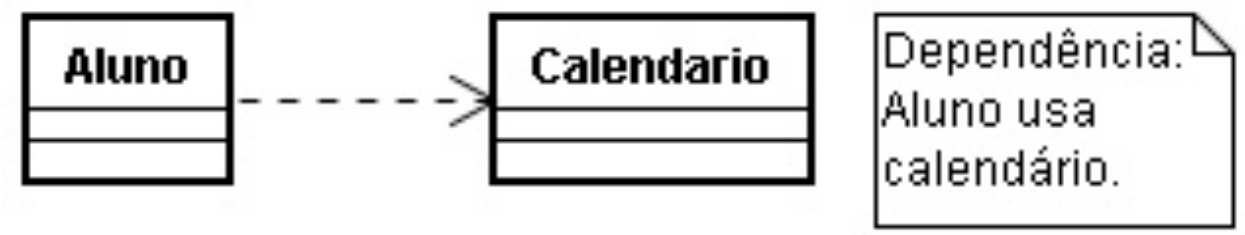
- constitui-se em uma variação da agregação, onde é apresentado um vínculo mais forte entre os objetos-todo e os objetos-parte, procurando demonstrar que os objetos-parte têm de estar associados a um único objeto-todo



Dependência

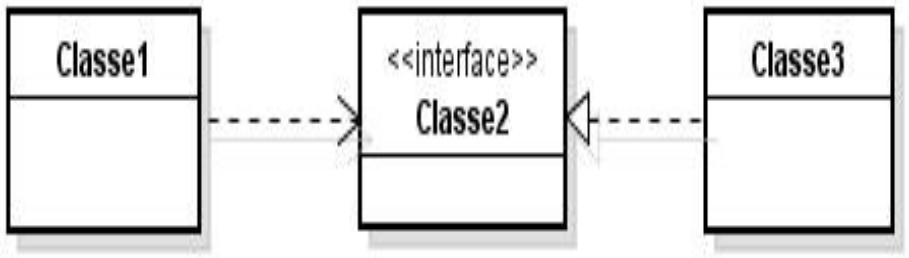
- É o relacionamento, como o próprio nome diz, identifica certo grau de dependência de uma classe em relação à outra





Realização (interface)

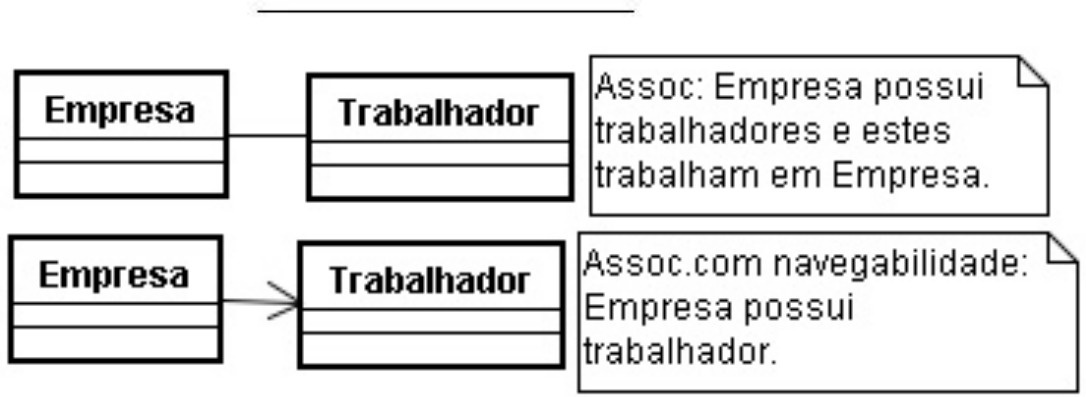
- É um tipo de relacionamento especial que mistura características dos relacionamentos de generalização e dependência, sendo usada para identificar classes responsáveis por executar funções para outras classes



Associação

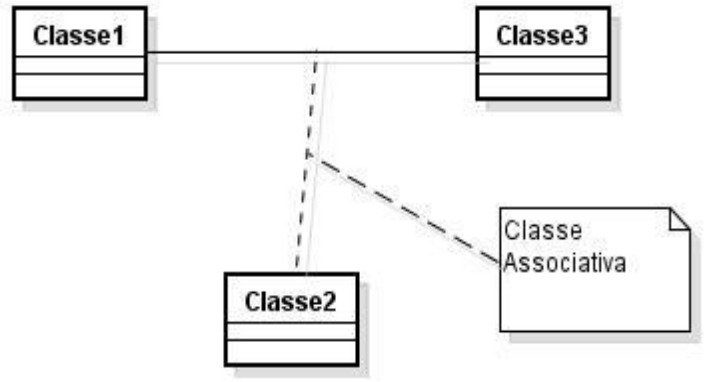
- Uma associação descreve um vínculo que ocorre normalmente entre os objetos de uma ou mais classes
  - **Unidirecional:** relacionamento de um objeto de uma classe com objetos da mesma classe

  - **Bidirecional:** relacionamento entre objetos de duas classes distintas



Classe Associativa

- São aquelas produzidas quando da ocorrência de associações que tenham multiplicidade muitos (*) em todas as suas extremidades

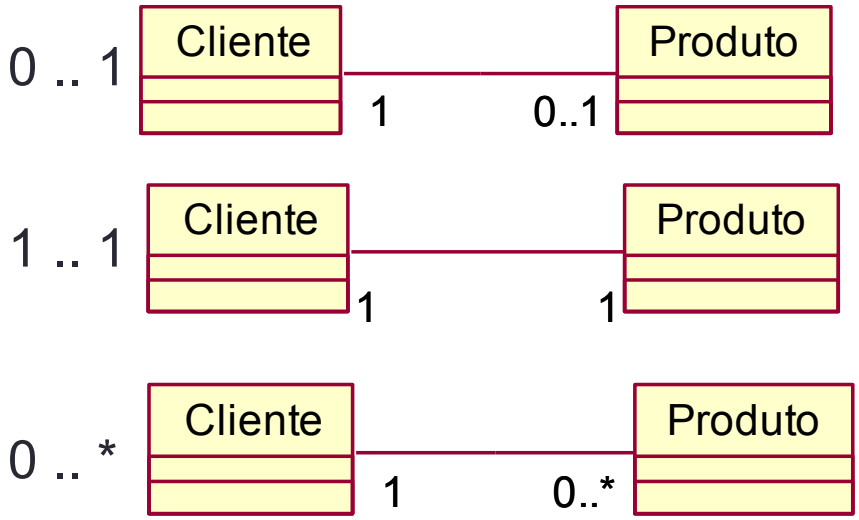


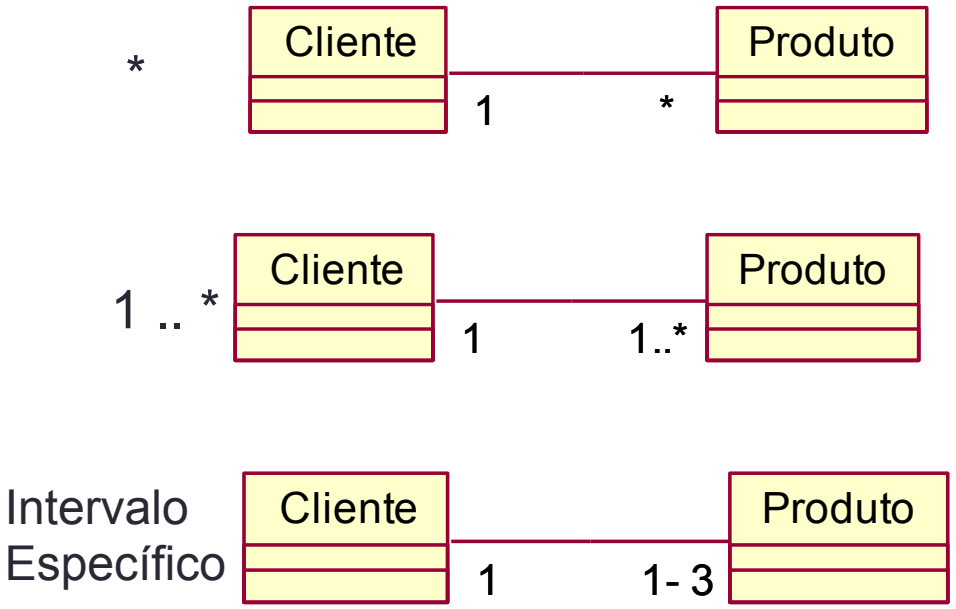
Multiplicidade (cardinalidade)

- **A multiplicidade especifica quantas INSTÂNCIAS de uma CLASSE relacionam-se a uma única INSTÂNCIA de uma CLASSE associada.**

- **A MULTIPLICIDADE depende de pressupostos e de como são definidas as fronteiras de um problema**

Exemplos de Multiplicidade





Tipo

• O tipo de atributo identifica um classificador que explica a espécie de informação que pode ser armazenada no atributo

Tipos → **int, boolean, varchar, double**

Visibilidade

- É utilizada para indicar o nível de acessibilidade de um determinado atributo ou método, sendo representada à esquerda destes.
- Existe basicamente quatro modos de visibilidade: público, protegido, privado e pacote

Privado : - atributo1
Protegido: # atributo2
Público : + atributo3
Pacote : ~ atributo4

Estereótipo

- permite a extensibilidade aos componentes ou associação da UML.

<<entity>>
<<boundary>>
<<control>>
<<enumeration>>

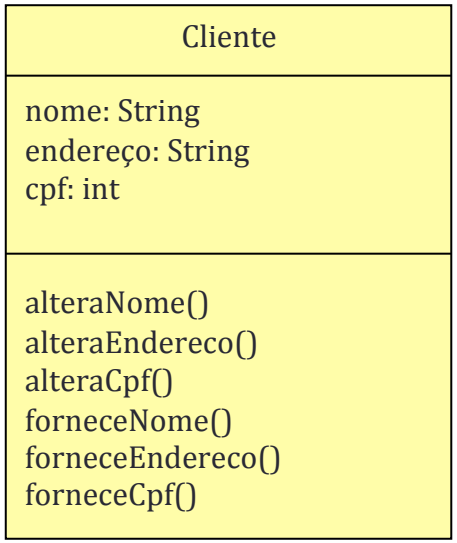
Classe x Objeto

- Uma **classe** é a descrição de atributos e comportamentos de um grupo de objetos com propriedades similares (atributos) e comportamento comum (operações ou métodos).

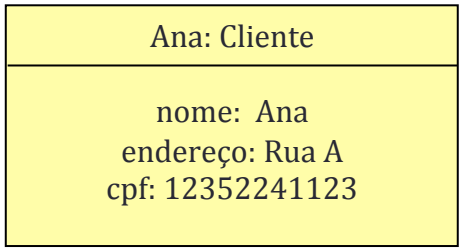
- Um **objeto** é uma instância de uma classe. Essa instância possui valores próprios em cada atributo definido na classe.

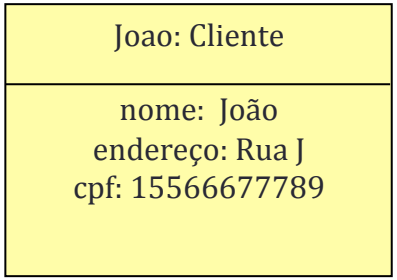
Exemplo de Classe e Objeto

Classe



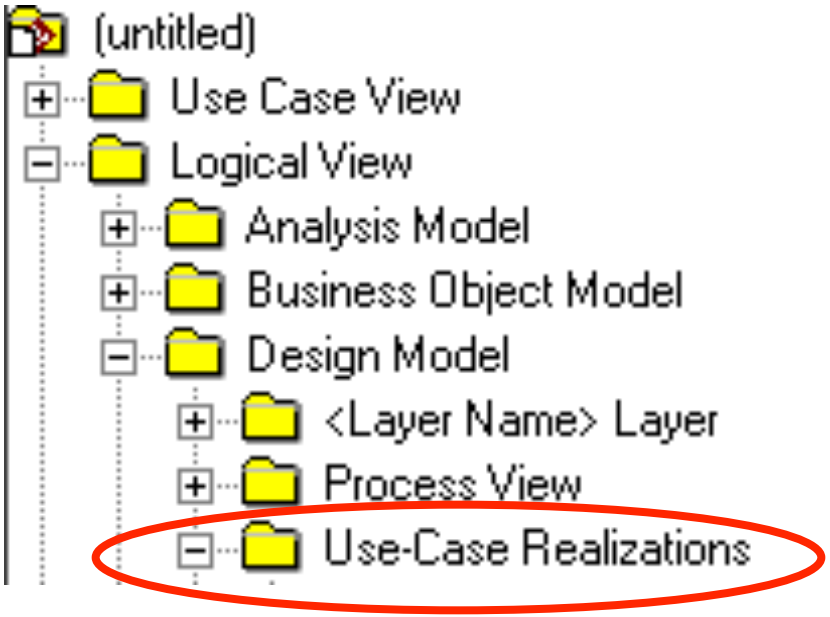
Objeto

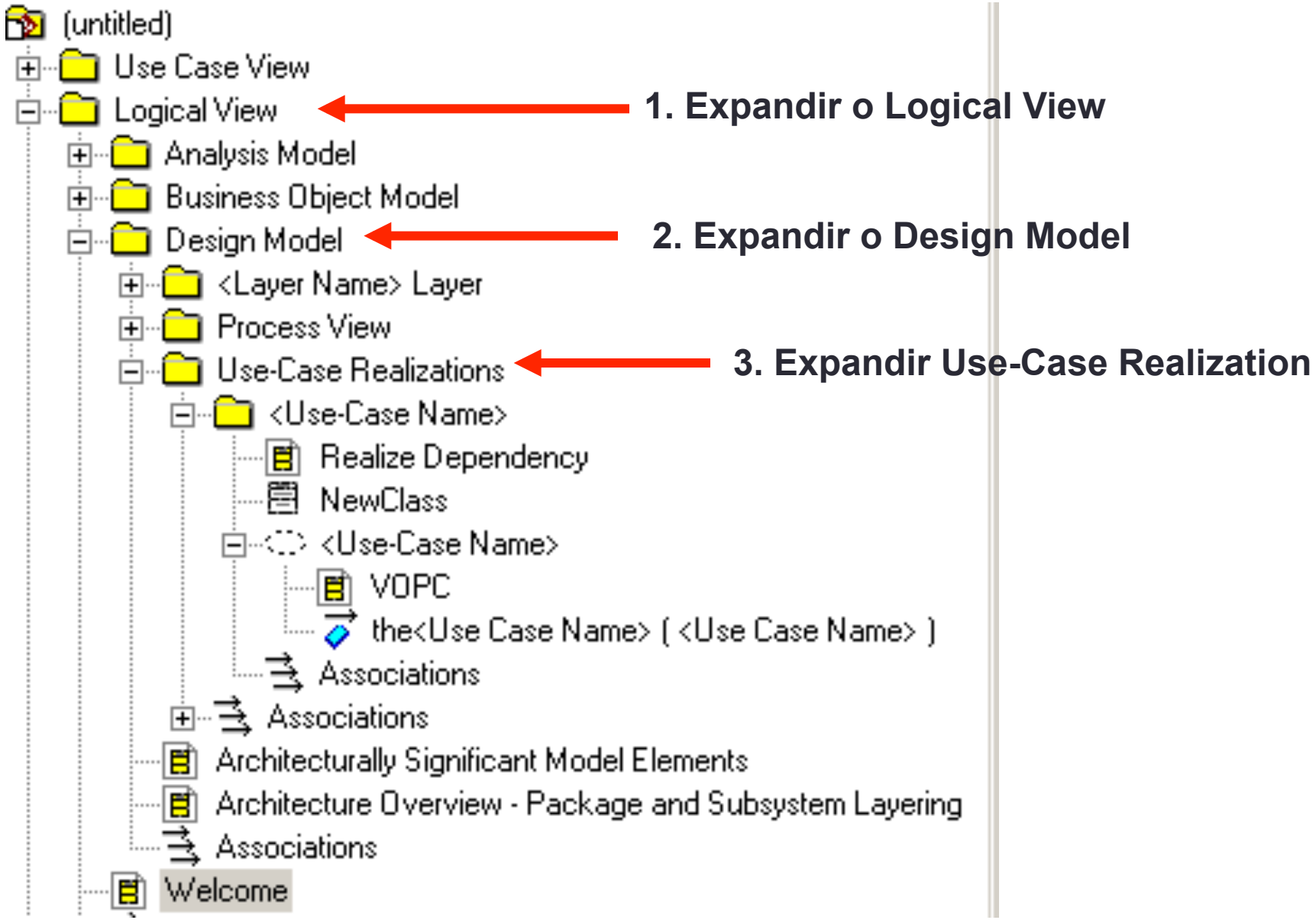




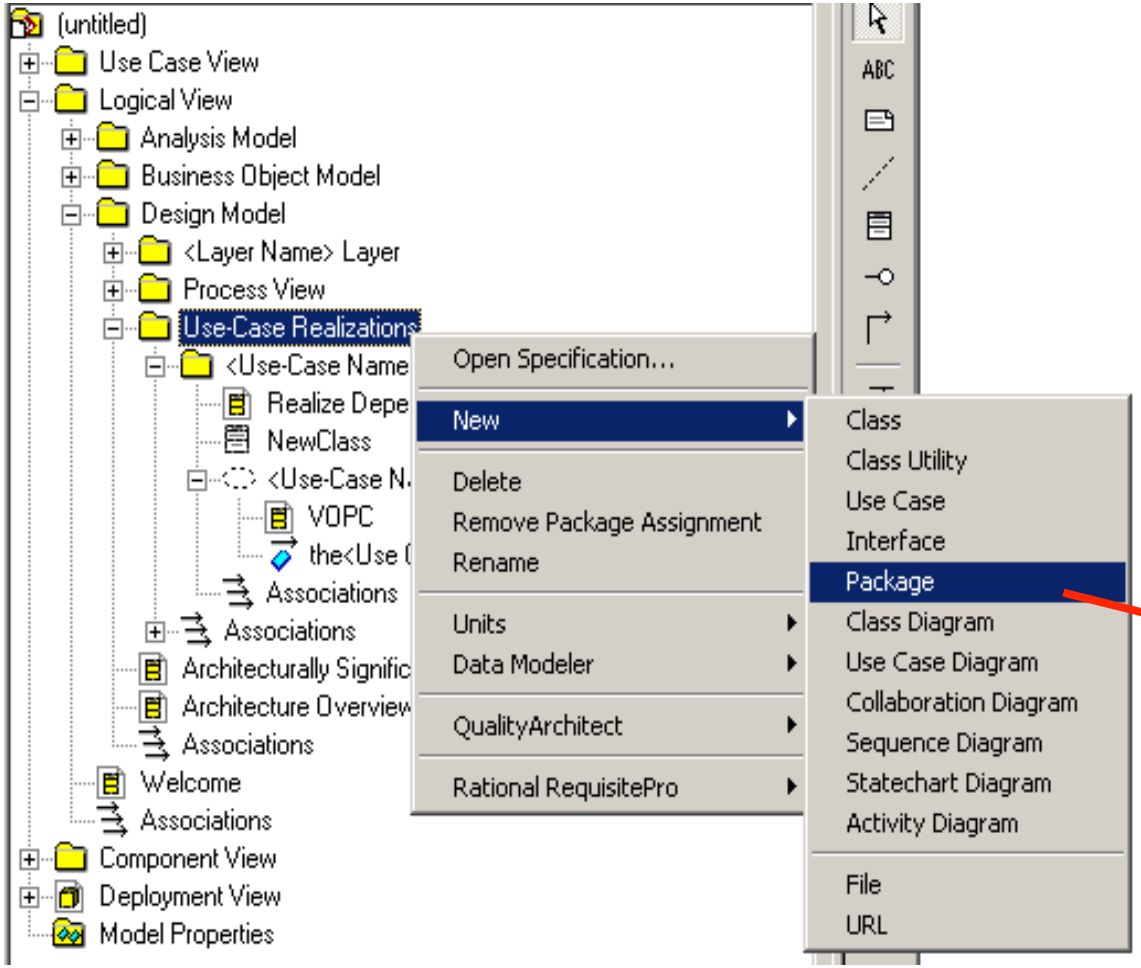
Onde fazer o Diagrama de Classe no Rational Rose?

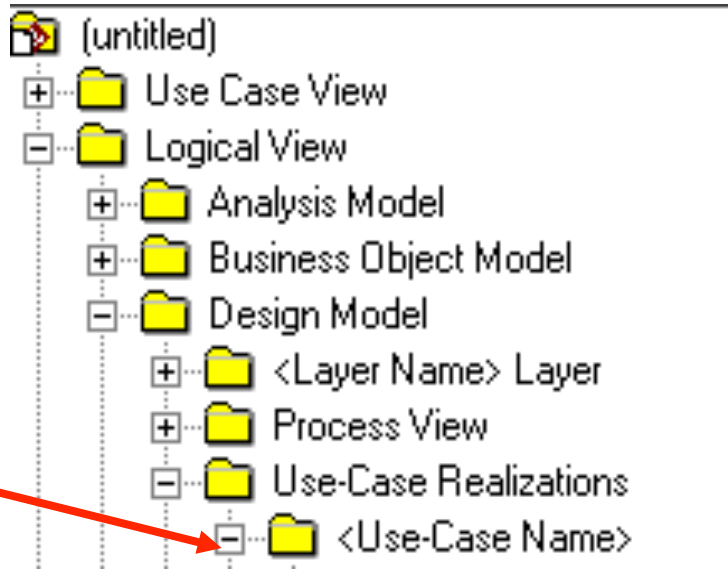
- O Diagrama de Classe deve ser feito no Logical View → Design Model
- Dentro do pacote Use-Case Realization



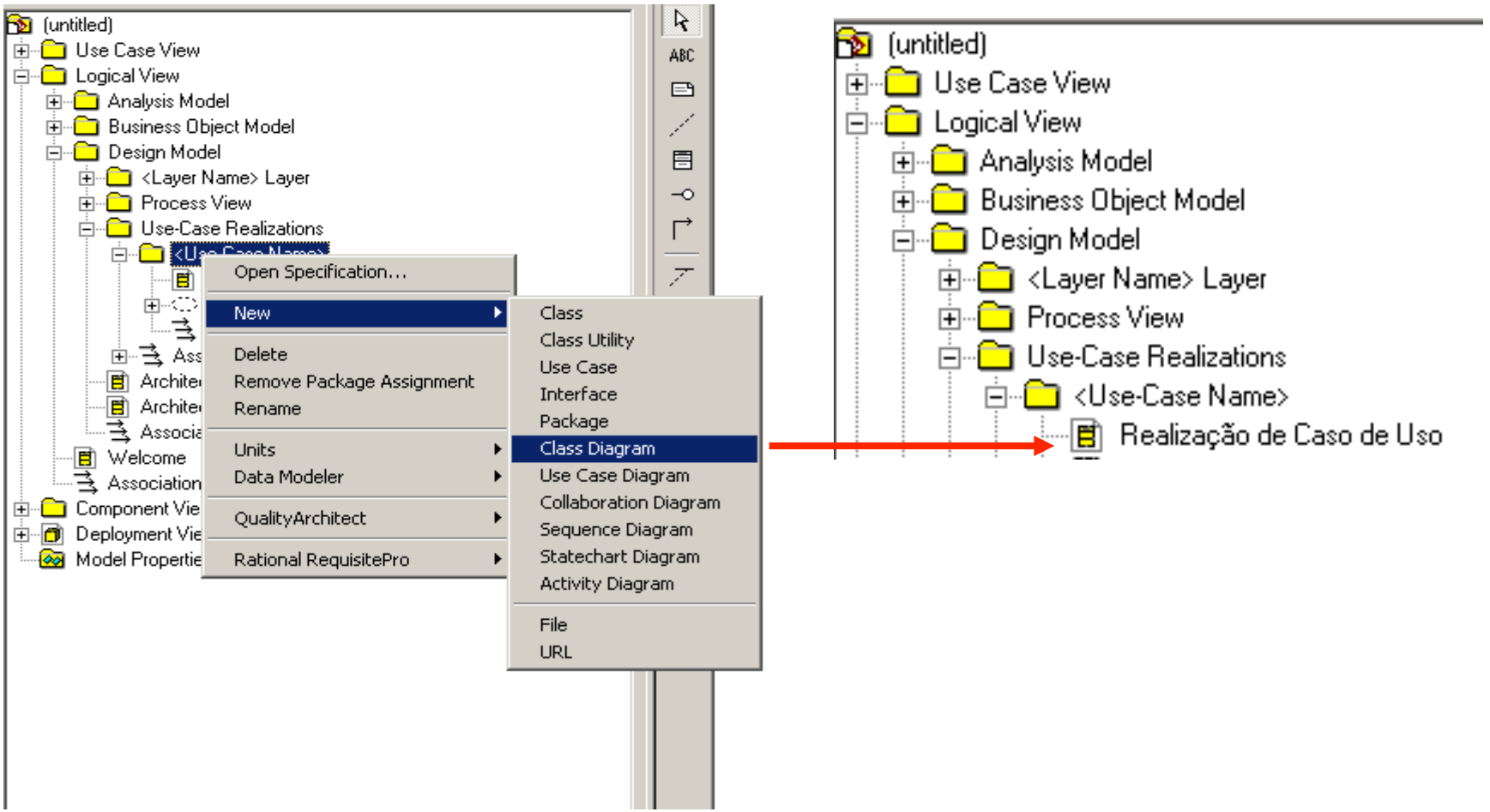

(untitled)
Use Case View
Logical View
Analysis Model
Business Object Model
Design Model
<Layer Name> Layer
Process View
Use-Case Realizations
<Use-Case Name>
Realize Dependency
NewClass
<Use-Case Name>
VOPC
the<Use Case Name> (<Use Case Name>)
Associations
Associations
Architecturally Significant Model Elements
Architecture Overview - Package and Subsystem Layering
Associations
Welcome
1. Expandir o Logical View
2. Expandir o Design Model
3. Expandir Use-Case Realization

4. Criar o pacote com o nome do Caso de Uso





5. Fazer a realização de caso de uso

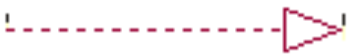


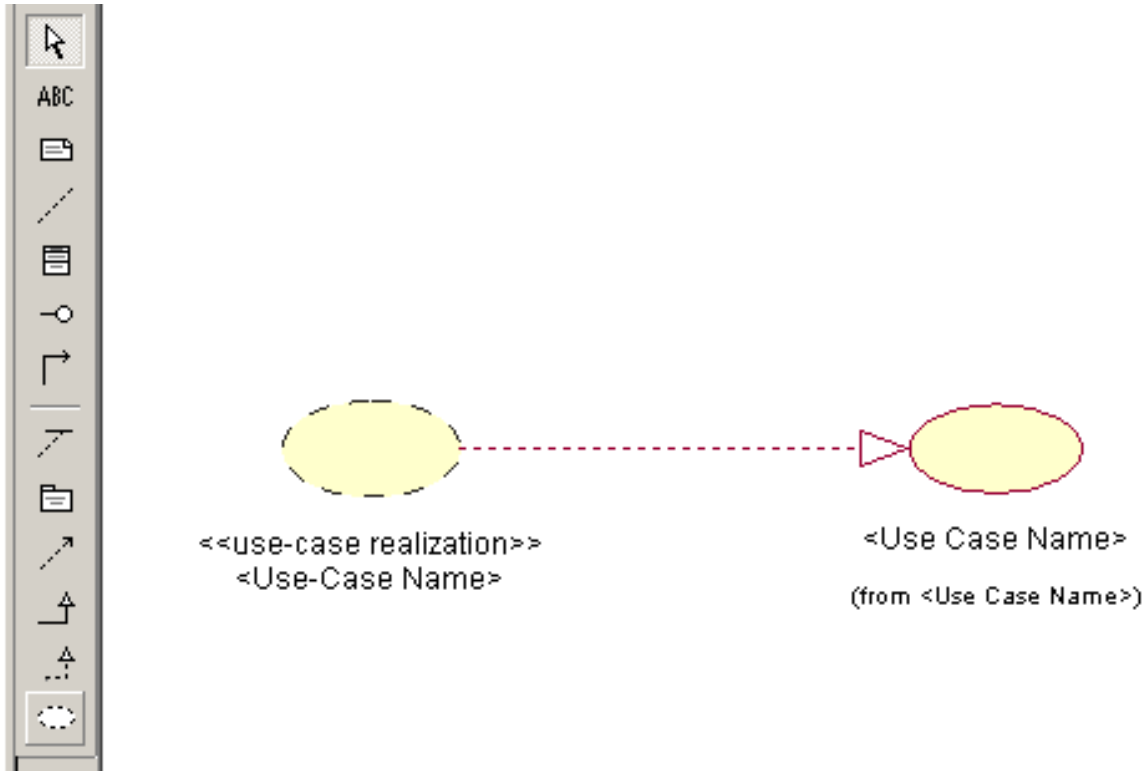
6. Clique duplo na Realização do Caso de uso, para fazer a realização

6.1. Arrastar o caso de uso do Use Case View

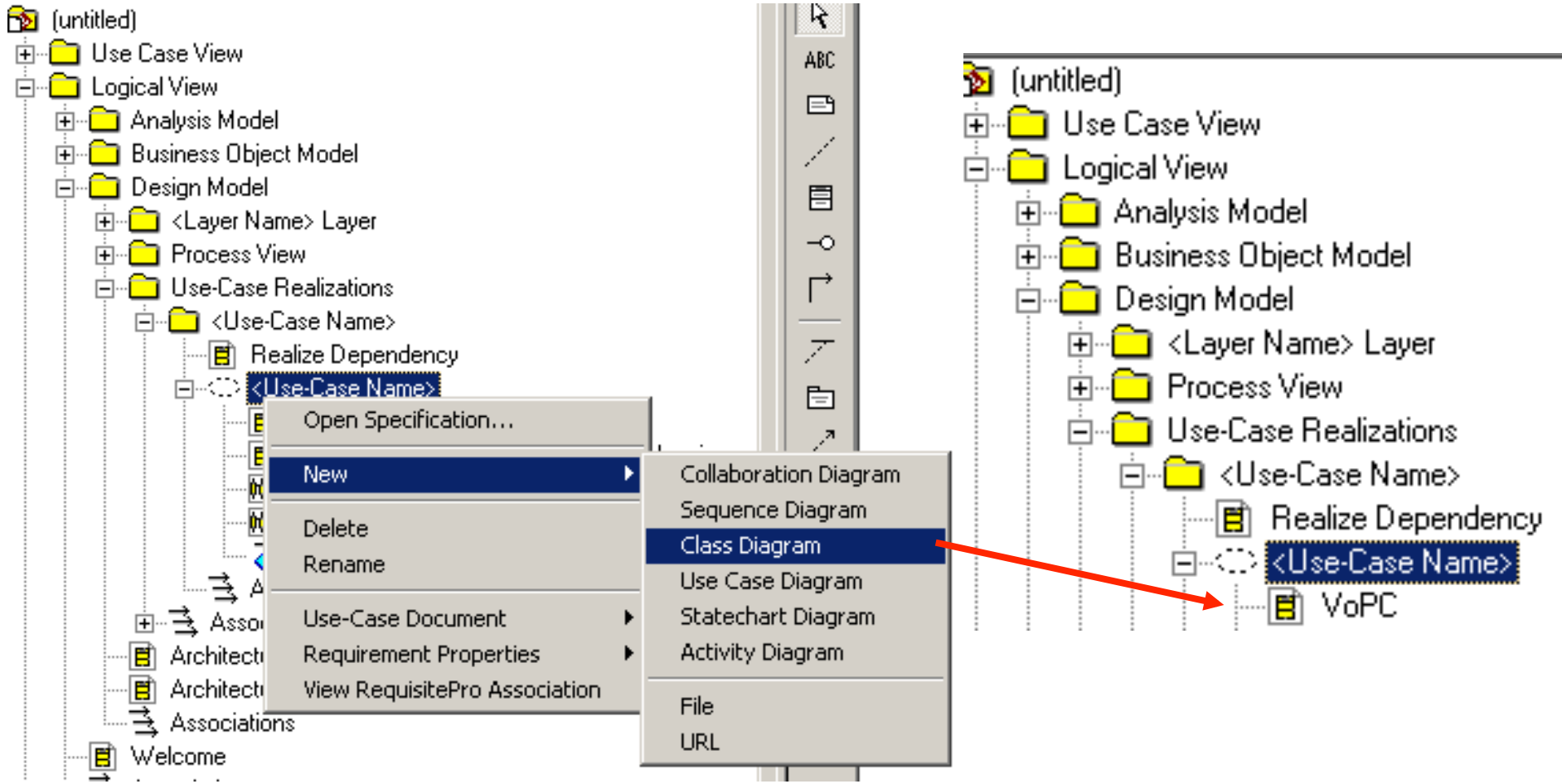
6.2. Arrastar o elemento do use case realization

6.3. Fazer a associação entre eles utilizando o Realize




<<use-case realization>>
<Use-Case Name>
<Use Case Name>
(from <Use Case Name>)

7. VoPC – View of Participating Class

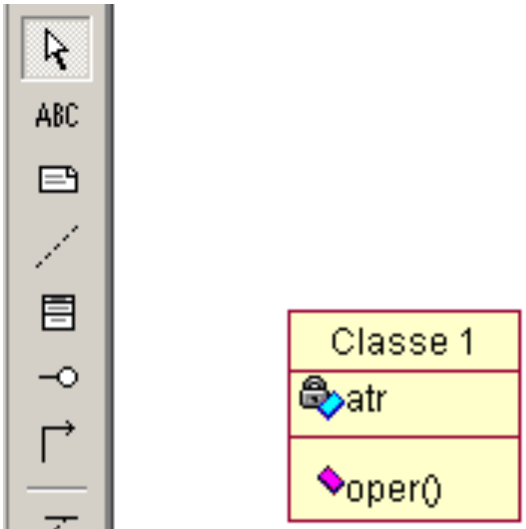


8. Clique duplo no VoPC, para a criação da classe

8.1. Arrastar o elemento que representa a classe

8.2. Inserir o nome da classe, atributos e operações

8.3. Caso necessário, efetuar a associação entre as classes. 


ABC
Classe 1
atr
oper()

- Após realizado todos os VoPC´s para as classes, deverá arrastá-las para o Welcome e efetuar as associações entre elas.