
Relacionamento entre classes

José Augusto Fabri

Relacionamento entre Classes (Introdução)

- Objetivo: Permitir a interação entre objetos para realização dos casos de uso.
- Grande parte dos relacionamentos são obtidos pelo diagrama de colaboração.
- Classes + Relacionamentos = Diagrama de Classes.
- Principal diagrama do projeto de software - descreve o esqueleto do produto.

Relacionamento entre Classes (Introdução)

- A partir do diagrama de classes, geração parcial do código fonte.
- Tipos de relacionamentos entre as classes:
 - Associação;
 - Agregação, e;
 - Generalização.

Rel. Classes (Associação)

- Tipo de relacionamento mais comum presente no diagrama de classe.
- Todos software OO. terá uma ou mais associação.
- A associação é um relacionamento que permite um objeto da classe A possa se comunicar com o objeto da classe B.
- Objetivo: Possibilitar a comunicação entre os objetos de duas classes.

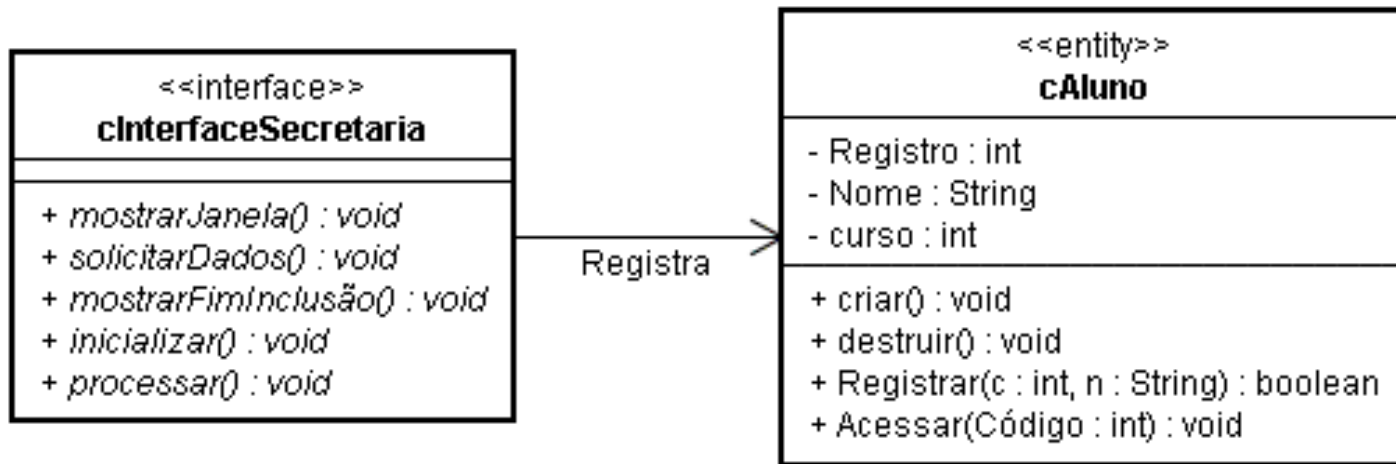
Rel. Classes (Associação)

- A associação aplica-se a classes independentes que precisam comunicar-se de forma uni ou bidirecional.
- Uma classe pode associar-se a mais de uma classe para fins de comunicação.
- Não existe um limite máximo para associações que uma classe pode possuir.
 - Isto é: uma classe pode se comunicar com se associar com várias classes ou X classes podem associar-se a um classe.

Rel. Classes (Associação - Notação UML)

- Segmento de reta ligando as duas classes.
- Associação unidirecional - inclui uma seta na extremidade de destino.
- Opcionalmente, mas fortemente sugerido, pode-se incluir um nome na associação.
 - Nome colocado sobre o segmento de reta e tem como objetivo indicar a natureza da associação.

Rel. Classes (Associação - Notação UML)

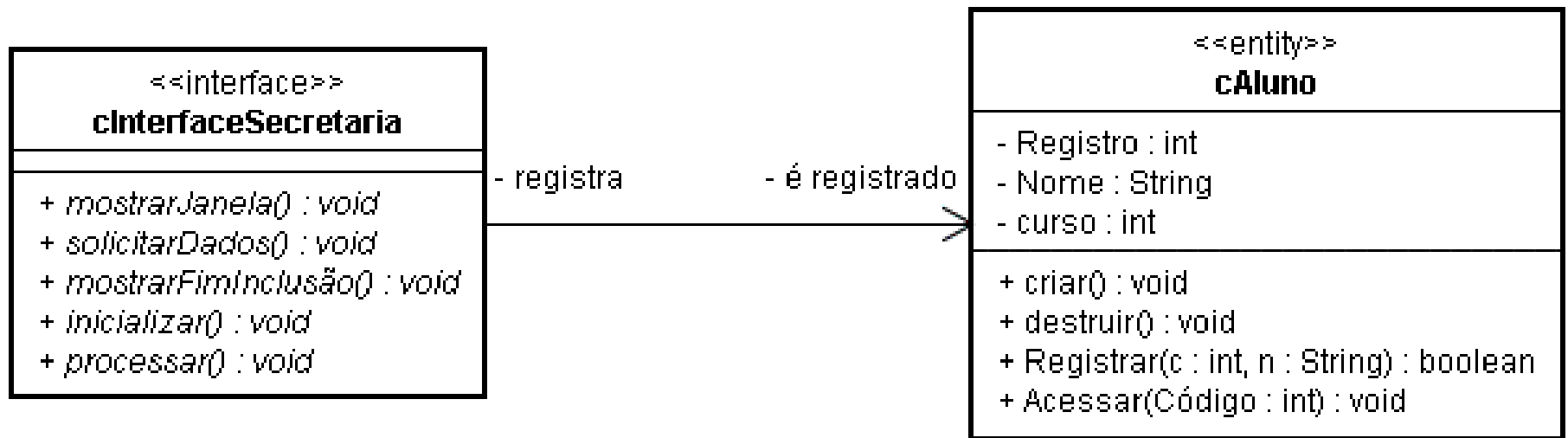


- Na figura acima, a associação Registra é unidirecional indicando que os objetos da classe cInterfaceSecretaria podem se comunicar com objetos da classe cAluno.
- Nesse caso a associação não pode ser navegada no sentido contrário.

Rel. Classes (Associação - Notação UML)

- Papeis das classes:
 - Pode-se incluir uma indicação dos papeis das classes nas associações.
 - Especificação de papel indica qual a participação ou atribuição de cada classe na associação.
- Indicação de papel:
 - Especificação do papel na forma de um texto nos extremos da associação.
 - Optar: Nome da associação ou papel das classes no relacionamento.

Rel. Classes (Associação - Notação UML)



Rel. Classes (Associação - Notação UML)

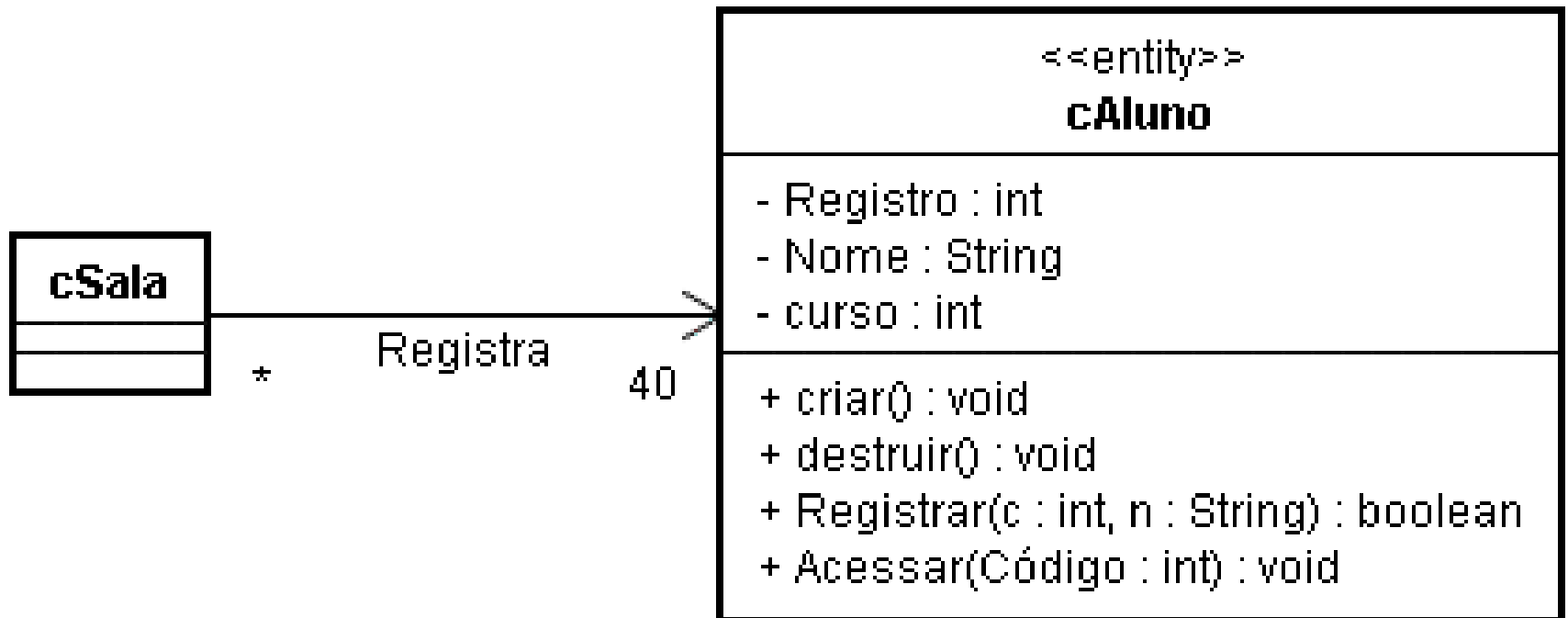
- Cardinalidade:
 - Especifica o número de objetos de cada classe envolvidos com a associação.
 - Quando não há uma especificação de cardinalidade, entende-se que a cardinalidade é 1.

Rel. Classes (Associação - Notação UML)

- A especificação da cardinalidade é feita em cada extremo da associação:
- Notação utilizada.

Notação	Significado	Exemplo
Constante numérica	Indica um número fixo de objetos	5
Faixa de valores	Indica um número variável de objetos dentro da faixa de valores especificada	1..4
Presença ou Ausência	Indica nenhum ou um (ou mais) objetos	0..1
Indefinido	Indica um número qualquer de objetos	*

Rel. Classes (Associação - Notação UML)

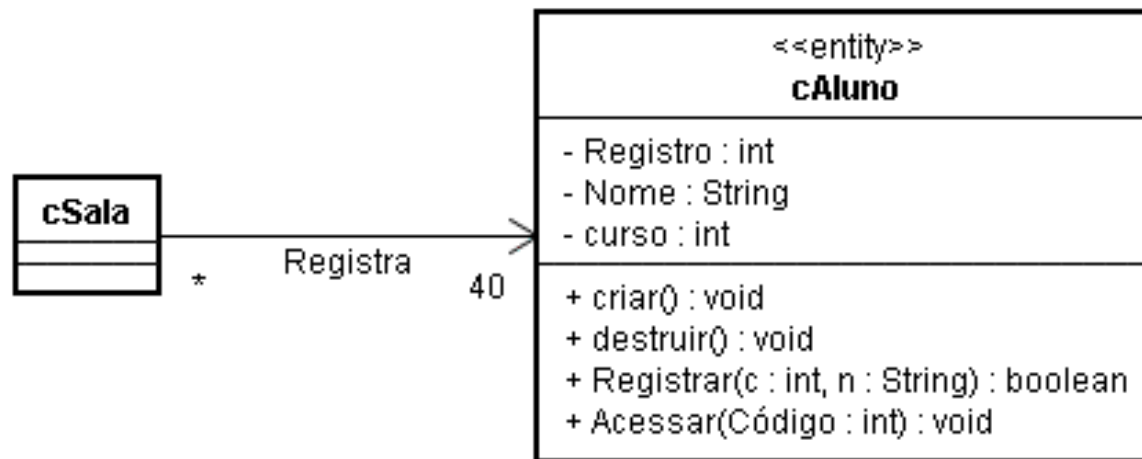


Rel. Classes (Associação - Notação UML)

- Como ler a cardinalidade?
 - Leitura de forma distinta para os dois sentidos da associação.
 - Para cada sentido, esquecer a cardinalidade no extremo de início da leitura e considerar que existe uma especificação de UM objeto da classe de início da leitura com tantos objetos quanto estiver especificado pela cardinalidade da classe da outra extremidade.

Rel. Classes (Associação - Notação UML)

- Exemplo de leitura:
 - Um objeto da classe cTurma se associa-se com 40 objetos da classe cAluno, e um objeto da classe cAluno se associa com um número indefinido (inclusive zero) de objetos da classe cTurma.



Rel. Classes (Levantamto das Associação)

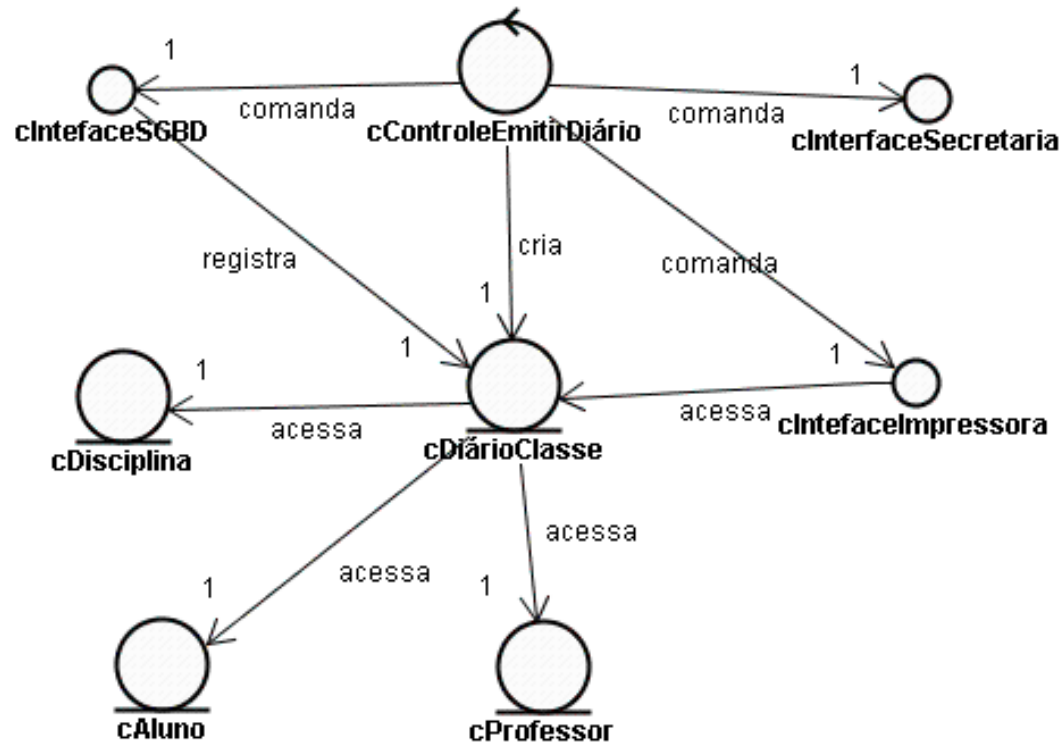
- Levantamento de associações:
 - Regra: Observar as necessidades de comunicação definidas nos diagramas de seqüência e de colaboração.
 - Para cada par de classes, verifica-se em todos os diagramas (seq. e colaboração) de todos os casos de uso, se existem mensagens trocadas entre objetos destas classes.

Rel. Classes (Levantando das Associação)

- Sempre que houverem mensagens trocadas, estabelece-se uma associação uni ou bidirecional conforme o sentido das mensagens.
- Embora este procedimento de levantamento permita uma definição bem fundamentada das associações, é ainda necessário um trabalho de complementação deste processo.
- Os DS e DC retratam cenário, um cenário pode não cobrir a totalidade de comunicações entre duas classes.
- Possibilidade de novas associações serem definidas.

Rel. Classes (Levantando das Associação)

- Exemplo de Associação baseada na figura III.16 (apostila).



Rel. Classes (Agregação entre Classes)

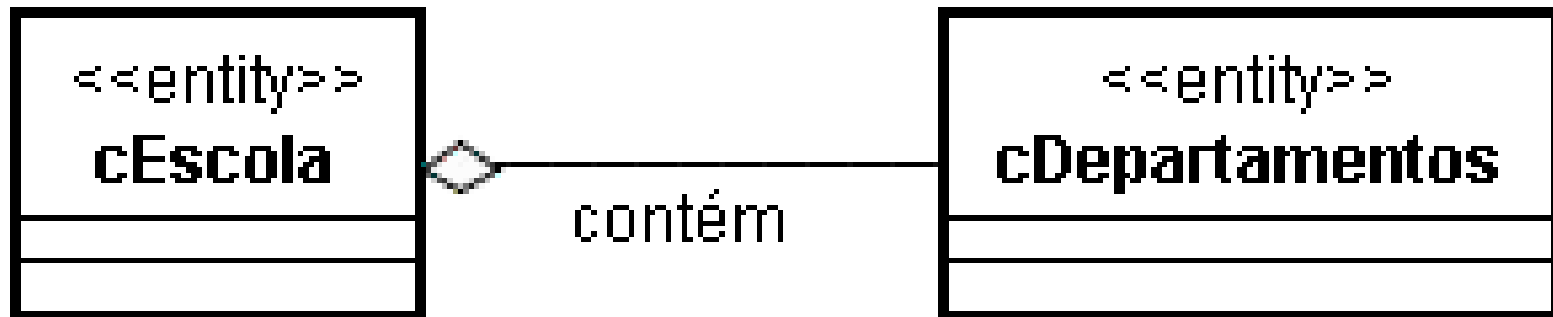
- A agregação é um relacionamento de pertinência entre classes.
- Objetivo: Estabelecer a inclusão de objetos de uma classe no interior de objetos de outra classe.
- Agregação: Uma relação “parte-de” já que o objeto agregado passa a constituir o fazer parte do objeto que agrega.
- Importante: Deve-se observar que não se trata de incluir (ou agregar) uma classe dentro da outra, mas objetos dentro de outros objetos.
- Agregação permite criar composições entre objetos – útil na definição de hierarquias de dados ou procedimentos.

Rel. Classes (Agregação entre Classes)

- A agregação fornece um canal de comunicação entre o objeto que contém e o objeto contido.
- Comunicação unidirecional do objeto que agrega para o objeto agregado.
- Objeto que é agregado não conhece, a princípio, o objeto que agrega. Desta forma ele não pode comunicar-se com o objeto que agrega.

Rel. Classes (Agregação entre Classes – Notação UML)

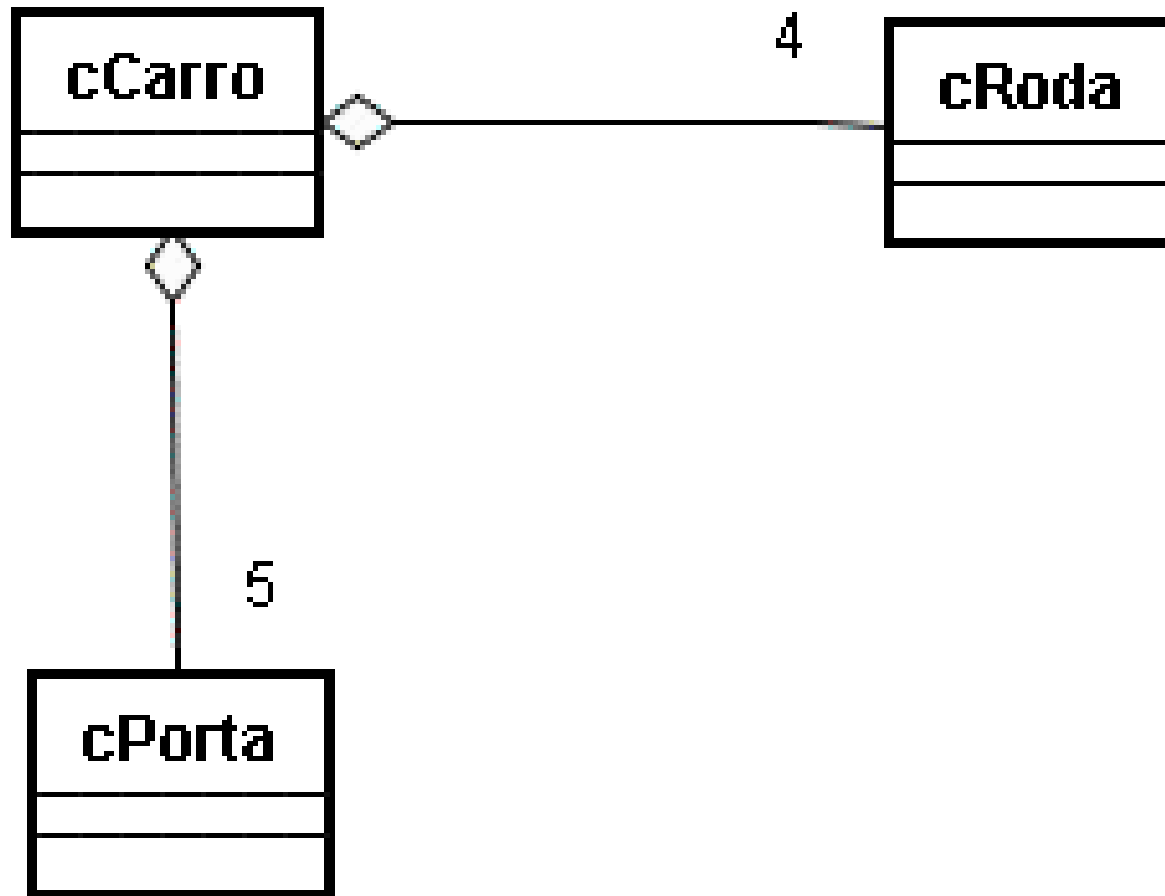
- Notação: Um segmento de reta ligando a classe dos objetos que agregam à classe dos objetos agregados.
- Inclusão de um losângulo.



Rel. Classes (Agregação entre Classes – Notação UML)

- Nome e papéis:
 - Inclusão do nome da agregação. Entretanto, como se trata sempre de uma relação de pertinência, a inclusão de nomes e papéis torna-se desnecessária, pois os nomes sempre seria “inclui” ou “contém”.
 - Cardinalidade: Mesma finalidade definida nas associações.
 - Importante: Um objeto não pode estar contido em mais de um objeto. Desta forma a cardinalidade no lado da classe dos objetos que agregam será sempre 1.

Rel. Classes (Agregação entre Classes – Notação UML)



Rel. Classes (Definição de Agregações)

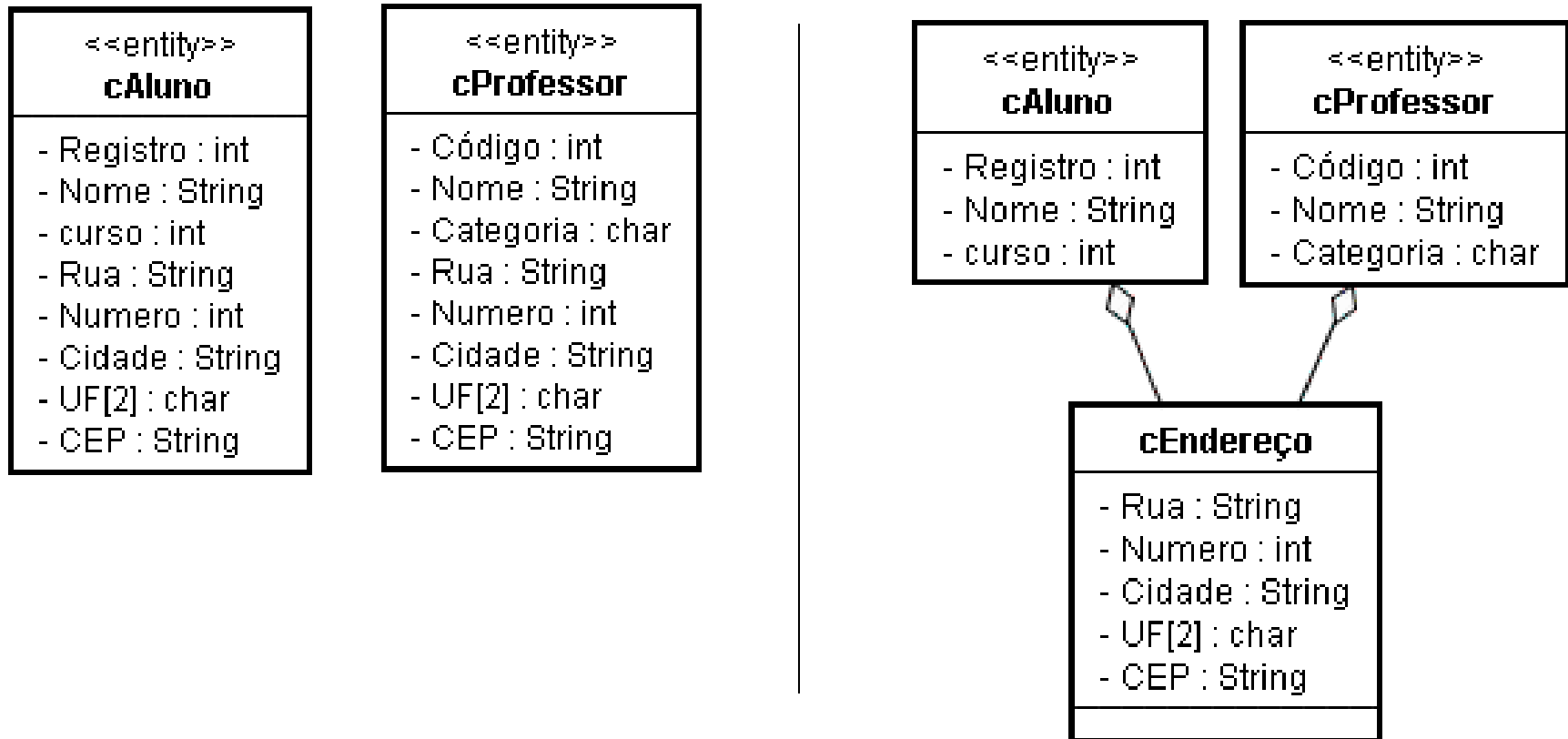
- Não existe uma técnica precisa para definição das agregações. Utilizar diretrizes gerais:
- Definição de agregação a partir de decomposições:
 - Classe com um conjunto extenso de responsabilidade.
 - Desmembrar a classe em classes menores.
 - Comunicação entre as classes para atender os serviços originais.
 - Inconveniente: Perda da identidade do classe.
 - Classe: cCarro – Desmembrada em cRodas, cMotor, cCarroceria (a identidade da classe se cCarro desapareceria)

Rel. Classes (Definição de Agregações)

- Definição de agregações a partir de composições:
 - Identificar conjunto de objetos que juntos compõem objetos maiores (objetos agregados que possuem uma identidade).
 - Possibilidade de criar novas classes definidas sobre relações de agregação com classes que representam suas partes.
 - Necessidade da utilização dos casos de uso do sistema.

Rel. Classes (Definição de Agregações)

- Definição de agregação a partir de partes comuns

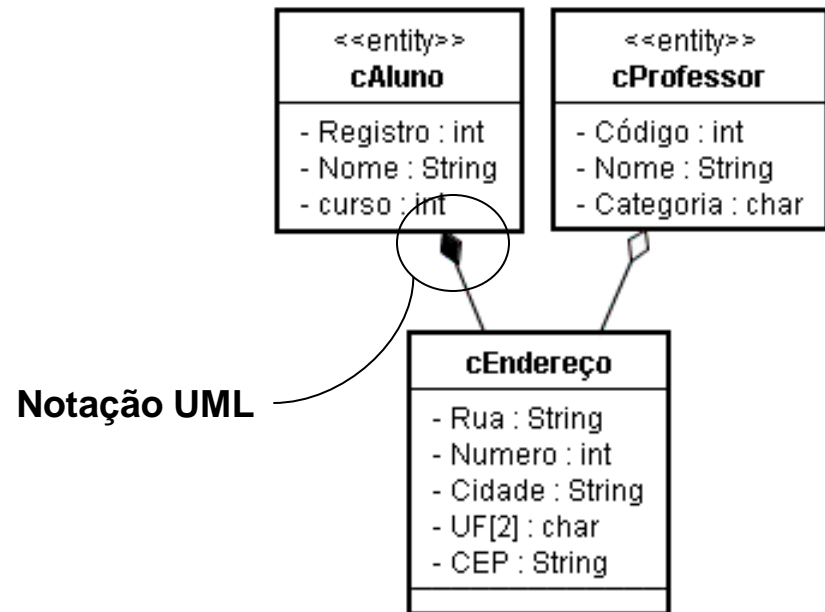


Rel. Classes (Tipos de Agregações)

- Tipos de Agregação:
 - Por Composição:
 - É uma agregação de fato. É feita a criação de um objeto dentro de outro objeto.

- Exemplo

```
class cAluno{  
    char nome[30];  
    cEndereço ender;  
    ....  
}
```



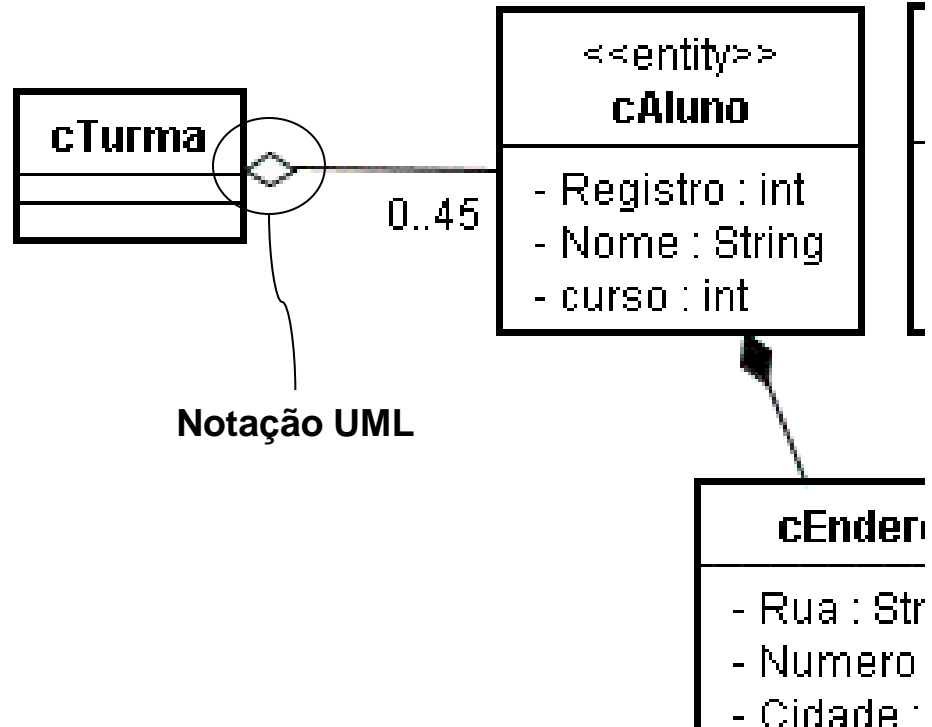
Rel. Classes (Tipos de Agregações)

- Agregação por Associação
 - Mesma interpretação da agregação por composição.
 - Diferença na forma de alocação do objeto – alocação dinâmica.
 - Objeto guarda apenas o endereço do objeto agregado.
 - Utilização: envolver um número variável de objeto.

Rel. Classes (Tipos de Agregações)

```
class cTurma
```

```
{  
    char codigo[8]  
    cAluno *aluno[45]  
    ....  
}
```



Rel. Classes (Generalização e Especialização)

- Envolvimento de duas classes:
 - Classe Base;
 - Classe Derivada.
- Leitura:
 - Da classe Base para Derivada = Especialização (a classe derivada é um caso especial).
 - Da classe Derivada para Base = Generalização (a classe base é caso geral).

Rel. Classes (Generalização e Especialização)

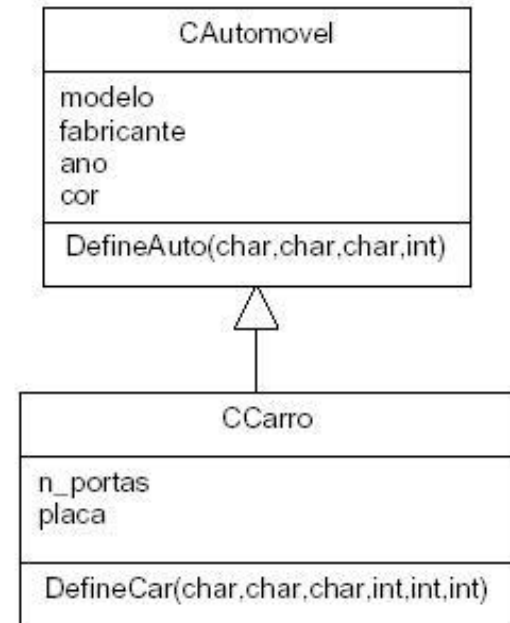
- Generalização e Especialização ocorrem sempre em uma forma de herança da classe base para a classe derivada.
- Classe derivada incorpora todos os métodos e atributos da classe base.
- Classe derivada também inclui métodos e atributos especiais.
- Importante: Nota-se que a herança é diferente da agregação no sentido em que na agregação incluía-se um objeto dentro do outro objeto perseverando-se o objeto que foi incluído. Na herança a classe base não mantém em seu interior o objeto da classe derivada.

Rel. Classes (Generalização e Especialização)

- Notação UML:
 - Segmento de reta ligando as duas classes com um triângulo ligado a classe base.

```
class CAutomovel
{
    char modelo[30];
    char fabricante[30];
    int ano;
    char cor;
public:
    DefineAuto(char, char, char, int);
};

class CCarro::public CAutomovel
{
    int n_portas;
    int placa;
public:
    DefineCar(char, char, char, int, int, int)
};
```



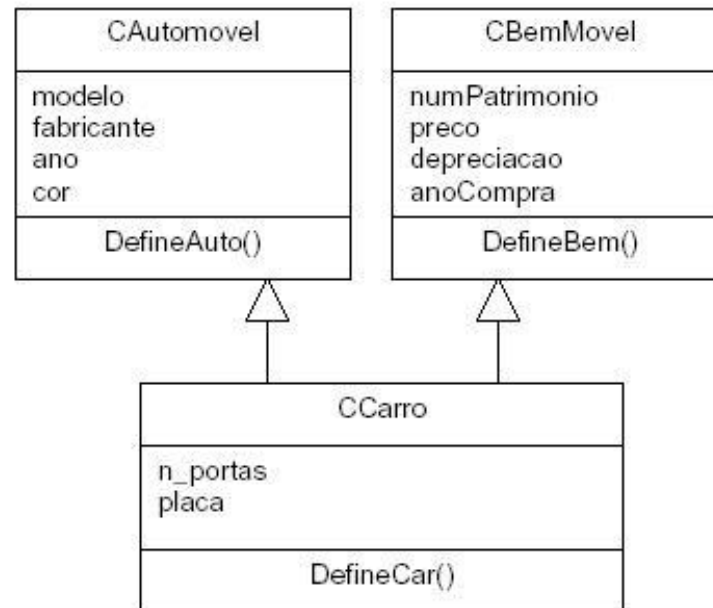
Rel. Classes (Generalização e Especialização)

- Herança múltipla:
 - Ocorre quando a classe base deriva mais de uma classe.

```
class CAutomovel
{
    char modelo[30];
    char fabricante[30];
    int ano;
    char cor;
public:
    DefineAuto(char, char, char, int);
};

class CBemMovel
{
    int numPatrimonio;
    float preco;
    int depreciacao;
    int anoCompra;
public:
    DefineBem(int, float, int, int);
};

class CCarro::public CAutomovel, public
CBemMovel
{
    int n_portas;
    int placa;
public:
    DefineCar(char, char, char, int, int, int)
};
```



Rel. Classes (Generalização e Especialização)

- Importante: Conceito de generalização parcial. A classe derivada é um caso especial de cAutomóvel e cBemMóvel.

Classes (Generalização e Especialização – Levantamento de generalização)

- Formas principais:
 - Identificação de partes comuns entre classes.
 - Comparar duas classes e verificar se as mesmas possuem parte comuns.
 - Se essas partes representarem mais da metade da classe e ao mesmo tempo as partes comum definem a essência da classe, pode-se criar uma classe base contendo as partes comuns e utilizar herança para integrá-la.
 - Síntese de classe base:
 - Concepção das classes a partir de classes abstratas.
 - Classes representando soluções gerais.
 - Gerar classes mais específicas seriam utilizadas através da herança.