Relacionamentos entre classes

Relacionamentos entre classes

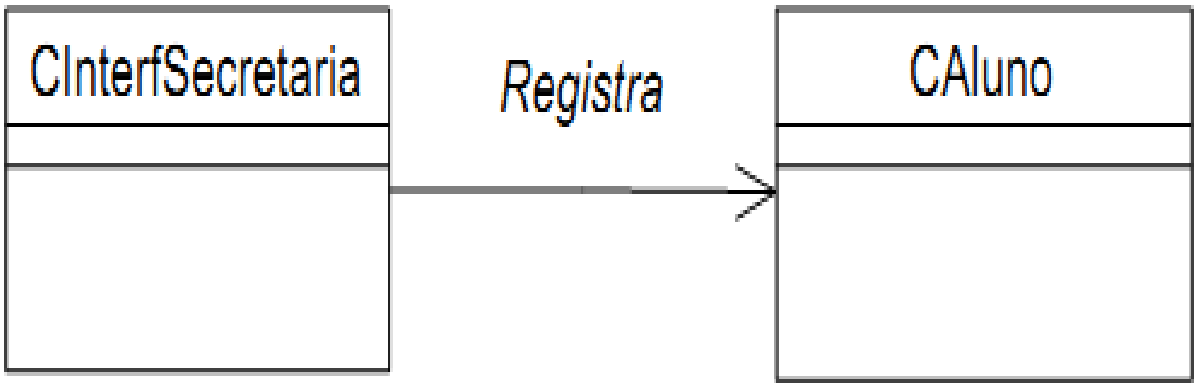
- Relacionamentos estruturais entre classes
- Precisam ser criteriosamente definidos durante o projeto do *software*
- São obtidos a partir da análise dos diagramas de colaboração
- É um dos principais diagramas da UML → define o esqueleto do sistema
- Há três tipos de relacionamentos entre classes: **associação, agregação e generalização**

Associação entre classes

- É o tipo de relacionamento mais comum e mais importante em um sistema orientado a objetos
- É um relacionamento ou ligação entre duas classes permitindo que objetos destas possam se comunicar
- Objetivo essencial da associação: possibilitar a comunicação entre objetos de duas classes
- A comunicação pode ser uni ou bidirecional

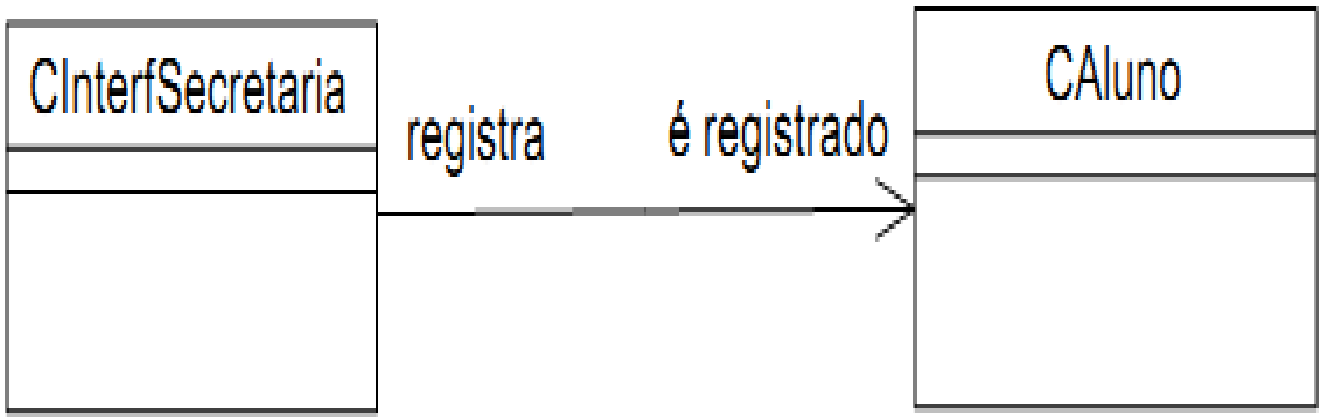
Notação UML para associações

- Segmento de reta ligando as duas classes
- Associação unidirecional: inclui-se uma seta na extremidade de destino
- Pode-se incluir um nome na associação
  - Indica a natureza ou finalidade da comunicação



Notação UML para associações

- Papéis das classes
  - Pode-se incluir uma indicação dos papéis das classes nas associações
  - Não é comum indicar o nome da associação e os papéis das classes para uma mesma associação → opta-se por uma ou outra



Notação UML para associações

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

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	*

Notação UML para associações

- A leitura da cardinalidade exige cuidados:
  - Deve-se fazer a leitura de forma distinta para os dois sentidos da associação
  - Para cada um dos sentidos:
    - Deve-se esquecer a cardinalidade no extremo de início
    - Deve-se considerar que existe a associação de um objeto da classe de início com vários objetos da classe de destino

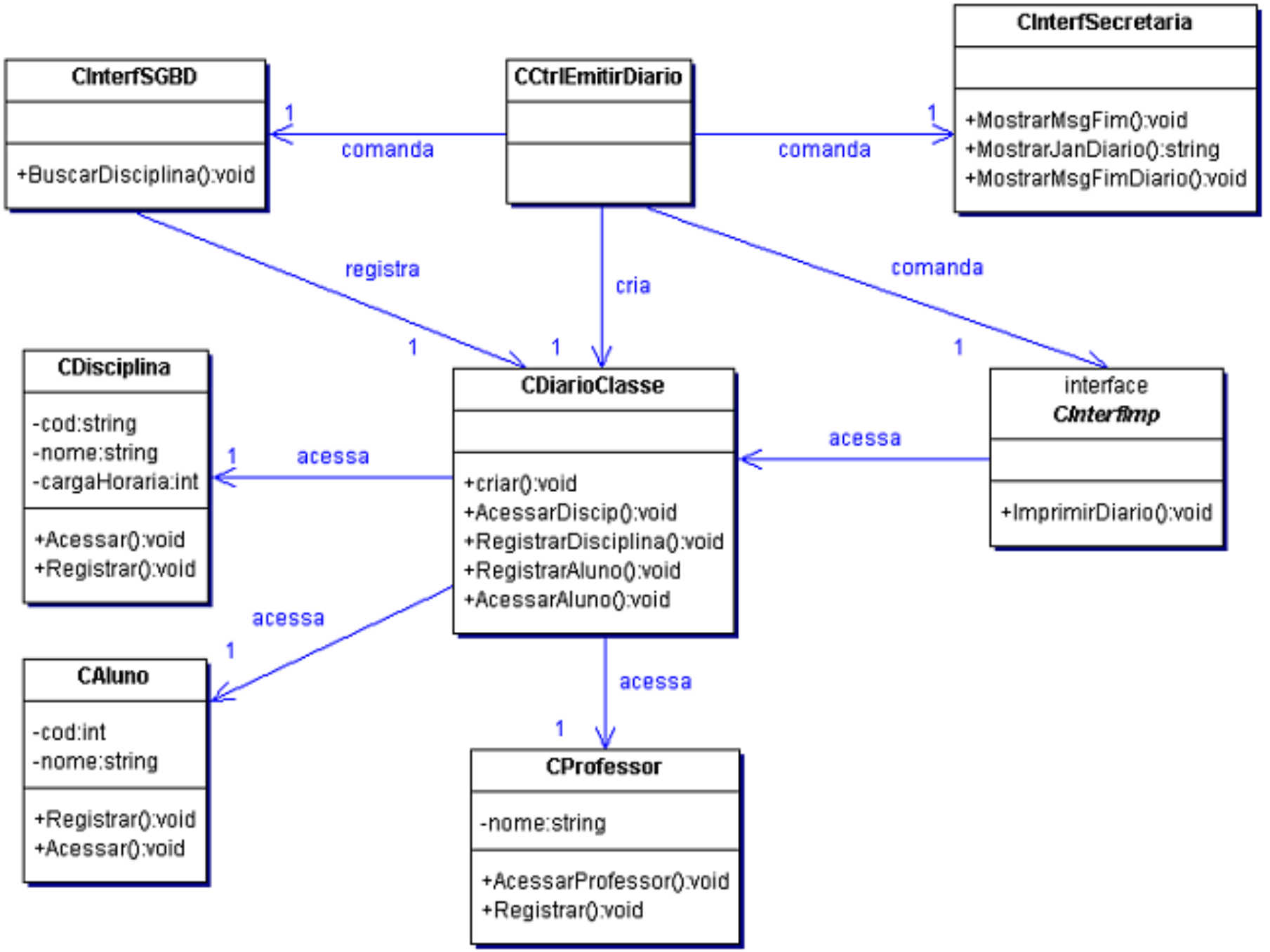
Notação UML para associações

- O exemplo abaixo é lido como:
  'Um objeto da classe CTurma se associa 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':

Levantamento de associações

- As associações são criadas a partir da necessidade de canais de comunicação entre objetos de duas classes
- As necessidades são obtidas a partir do diagrama de sequência e de colaboração
- Exemplo do levantamento de associações a partir de um diagrama de colaboração:

Levantamento de associações

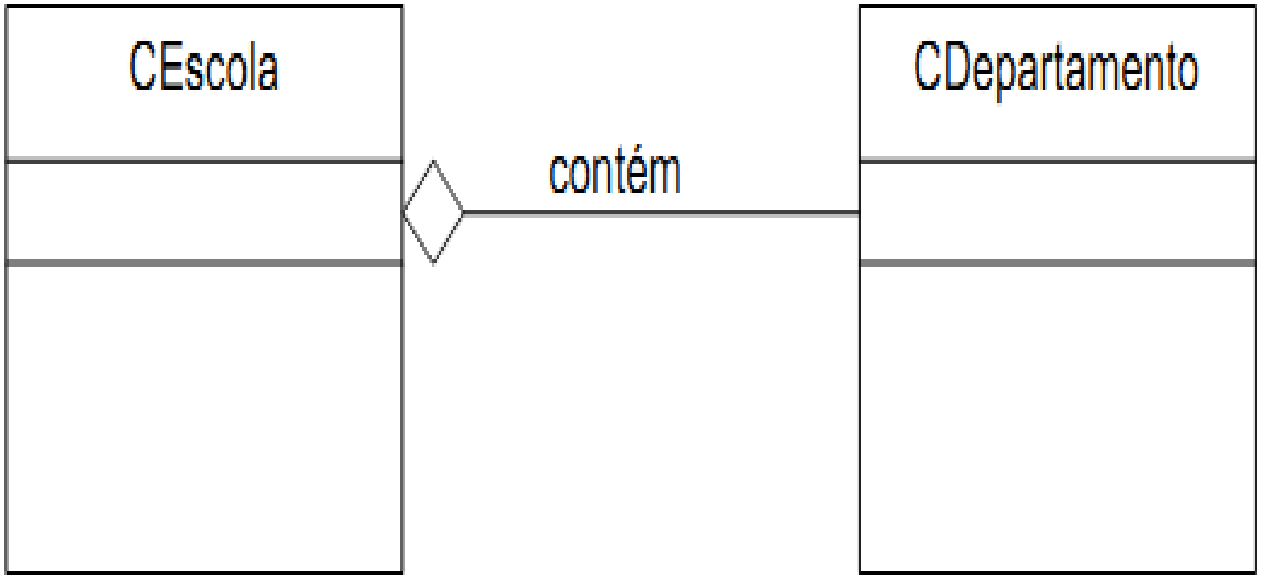


Agregação entre classes

- Relacionamento de pertinência entre classes
  - Permite a inclusão de objetos de uma classe no interior de objetos de outra classe
- Relação 'parte-de', 'tem-um', 'todo-parte'
- Objeto que agrega conhece o agregado mas este não conhece aquele → comunicação unidirecional
- Notação UML: segmento de reta ligando a classe dos objetos que agregam à classe dos objetos agregados.

Agregação entre classes

Na extremidade da classe dos objetos que agregam inclui-se um losango:

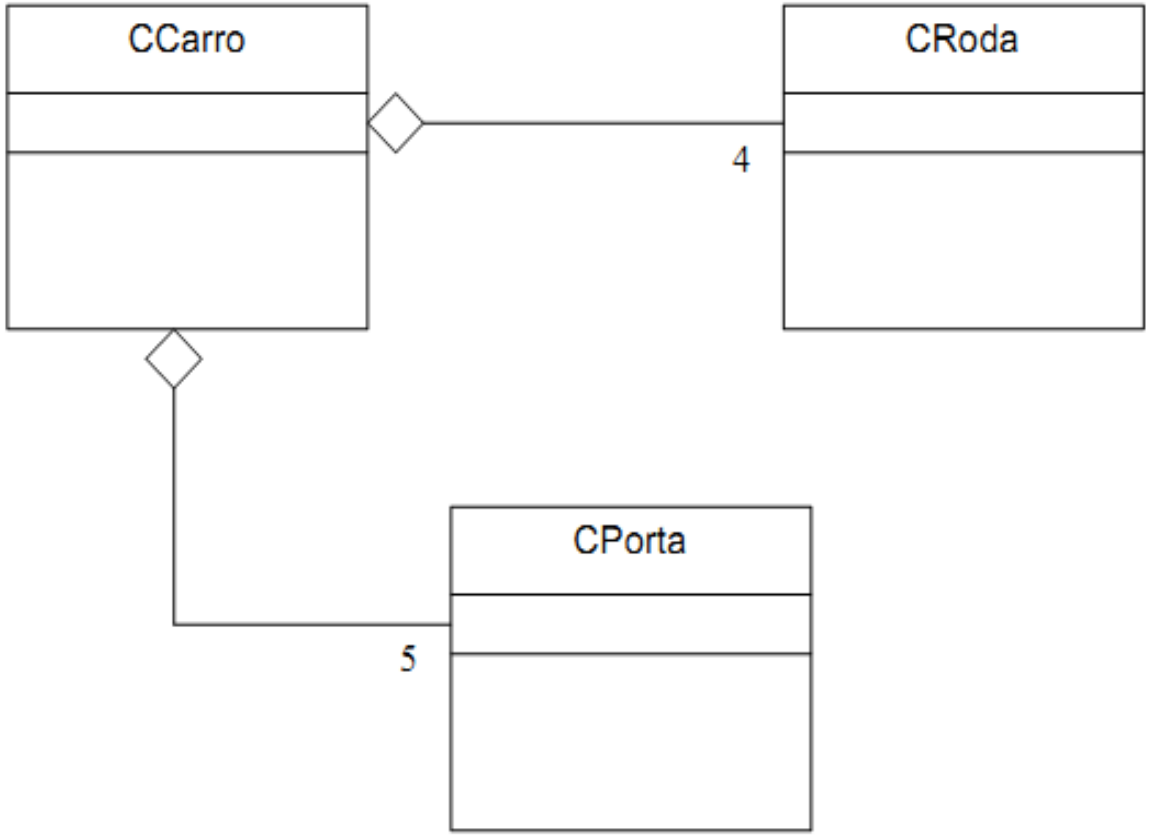


Agregação entre classes

- Cardinalidade
  - Número de objetos envolvidos na relação
  - Um objeto pode incluir vários outros mas não pode estar contido em mais de um objeto
    - i.e. cardinalidade no lado da classe dos objetos que agregam será sempre 1

Agregação entre classes

- Exemplo: um objeto da classe CCarro contém 4 objetos da classe CRoda e 5 objetos da classe CPorta:



Definição de agregações

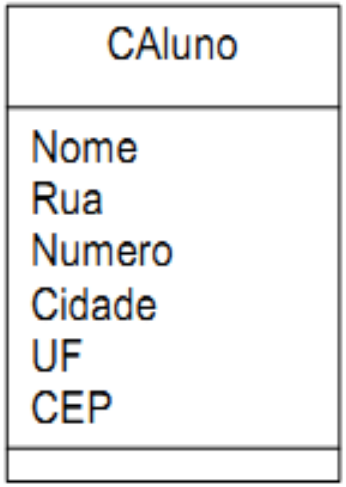
- Não existe uma técnica precisa para de se definir as agregações em um sistema
- Sugestões:
  - definição das agregações a partir de decomposições
    - quando uma classe tem muitas responsabilidades, tende-se a dividi-la. Tal divisão pode fazer com que a classe perca sua identidade
      - neste contexto, as agregações são muitos úteis porque dividem uma classe grande em outras menores, sem que a grande perca a identidade
  - definição das agregações a partir de composições
    - raciocínio é o inverso ao da decomposição
    - procura-se identificar conjuntos de objetos que juntos compõem objetos maiores

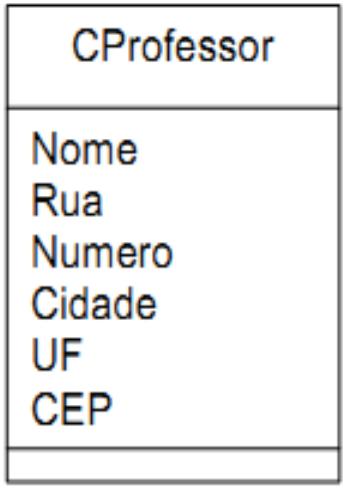
Definição de agregações

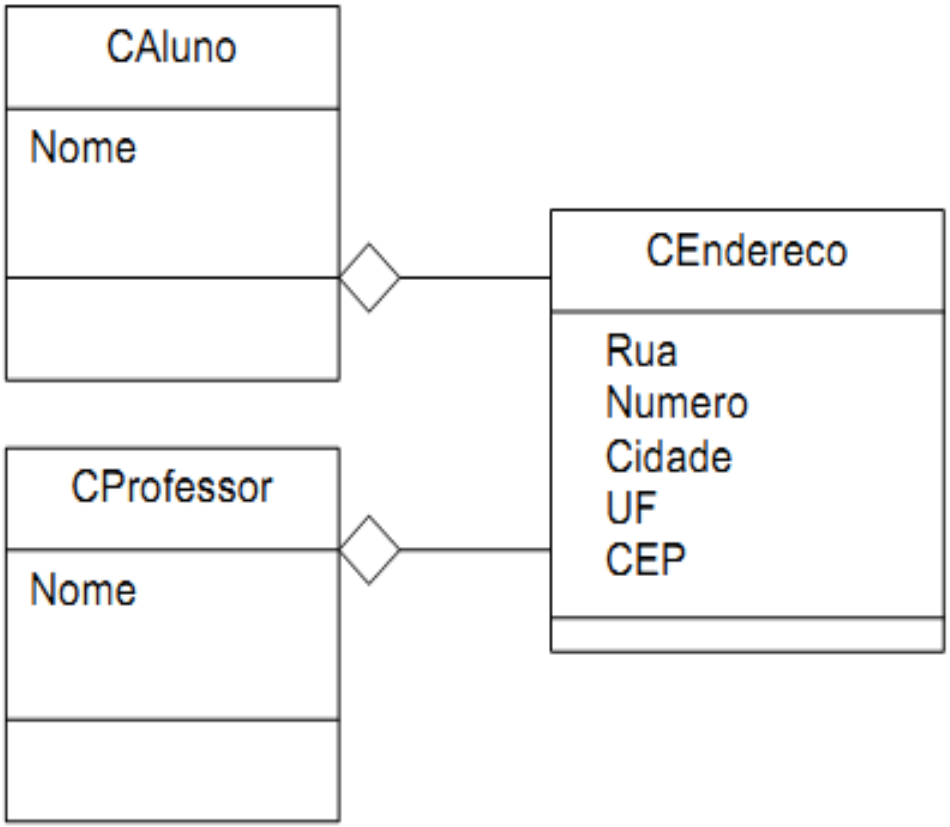
- definição de agregações a partir de partes comuns
  - quando se percebe dentro de duas ou mais classes um conjunto de atributos e/ou métodos semelhantes
    - se estes juntos possuem uma identidade, então poderia ser criada uma nova classe:

Definição de agregações

- Definição de agregações a partir de partes comuns





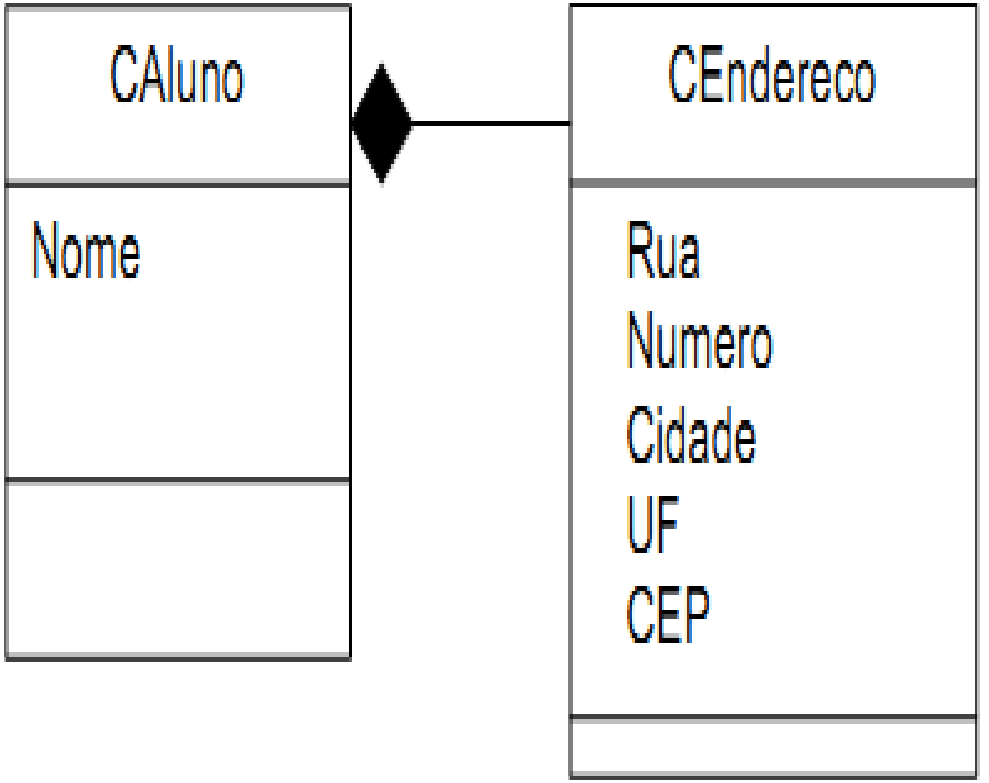


Tipos de agregação

- Agregação por composição
  - é uma agregação de fato
    - realmente é feita a criação ou alocação (estática) de um objeto dentro do outro
    - Exemplo: o objeto ender da classe CEndereco está sendo instanciado dentro da classe CAluno
    - O número de objetos agregados é fixo, uma vez que são alocados dinamicamente
    - A notação é um losango preenchido

Tipos de agregação

```
class CAluno
{
    char nome[30];
    CEndereco ender;

     ...
};
```

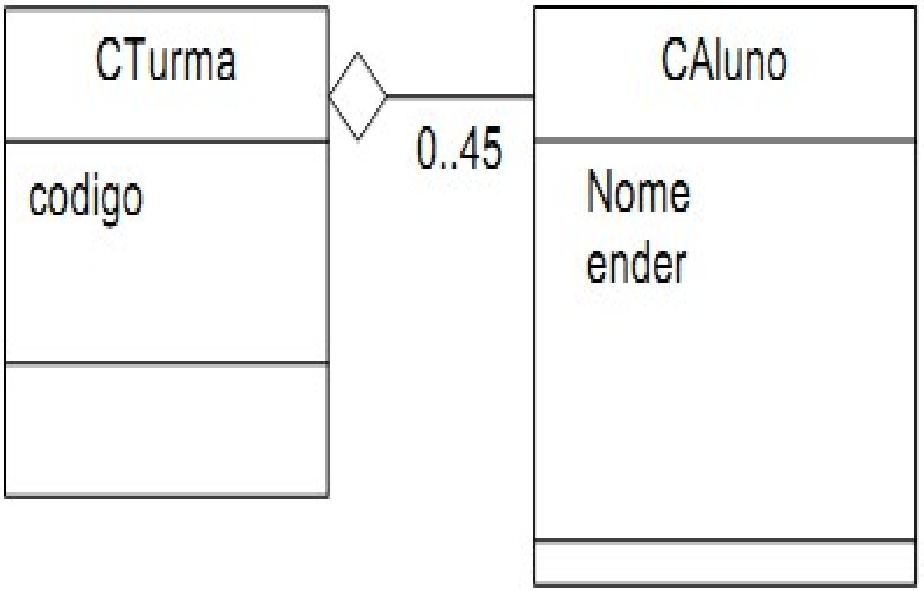


Tipos de agregação

- Agregação por associação
  - Tem a mesma interpretação do que a agregação por composição
    - Entende-se que o objeto agregado é um componente do objeto que agrega
  - A alocação do objeto agregado ocorre de forma estática fora do objeto que agrega ou de forma dinâmica em seu interior
  - Sua implementação ocorre através de associações
    - São consideradas agregações quando se realiza um controle de escopo para os objetos agregados
  - Utilizada quando se deseja estabelecer uma agregação envolvendo um número variável de objetos

Tipos de agregação

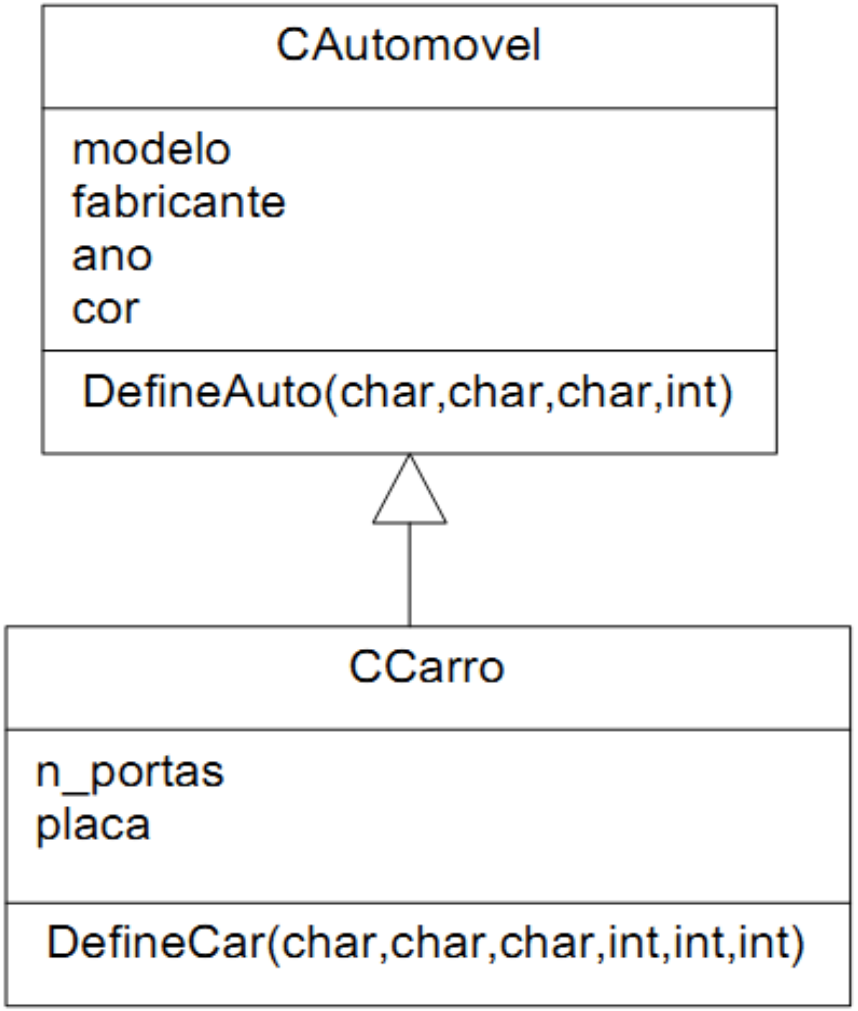
```
class CTurma
{
    char codigo[8];
    CAluno *aluno[45];

    ...
};
```



Generalização/Especialização de Classes

- Relacionamento estrutural entre duas classes
  - classe base (superclasse)
  - classe derivada (subclasse)
- Herança (relacionamento 'é um'
  - a derivada herda as propriedades da base

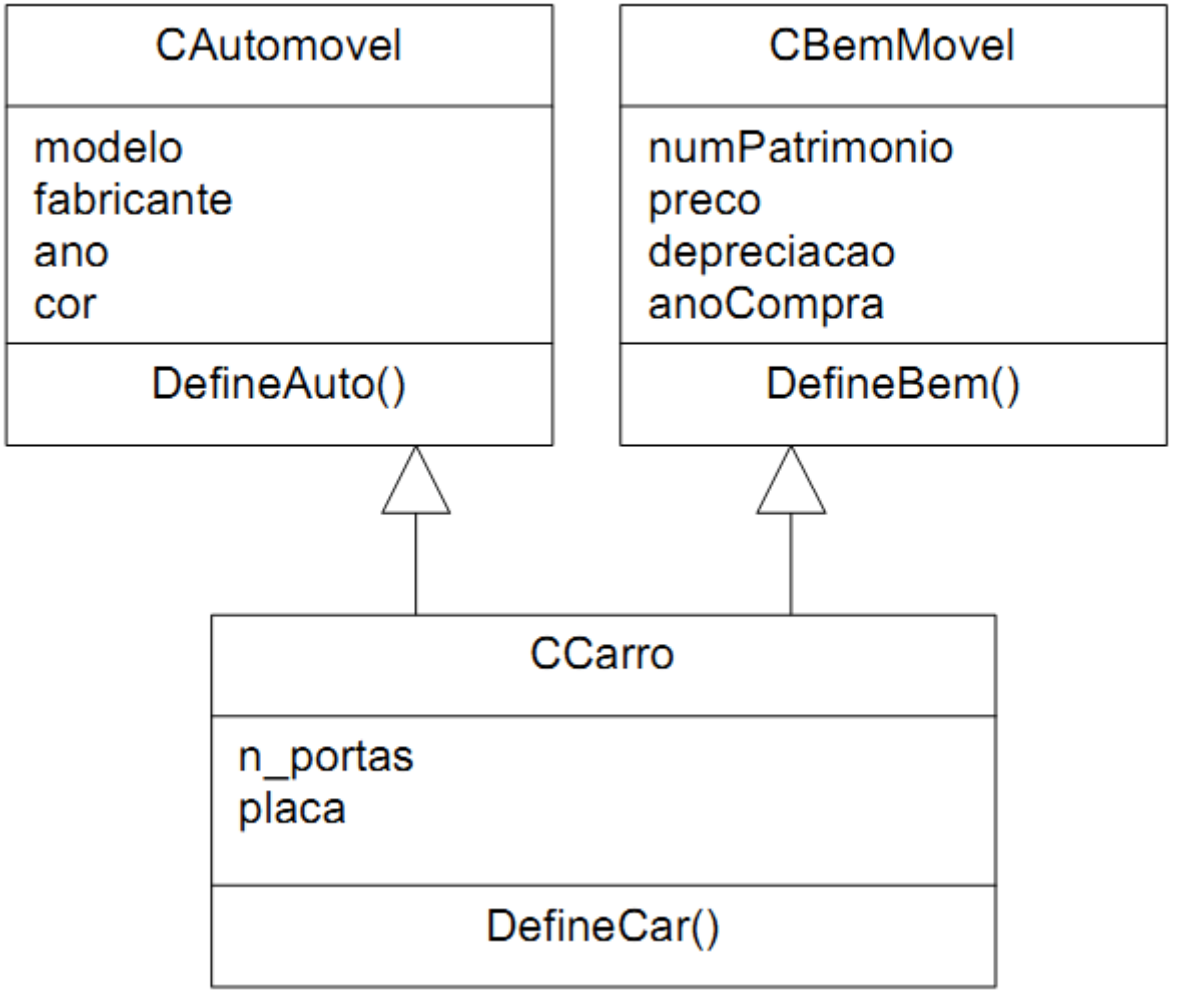
Generalização/Especialização de Classes




```
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)
};
```

Herança Múltipla

Ocorre quando uma classe deriva de mais de uma classe base



Herança Múltipla

Quando uma classe deriva de mais de uma classe base

```
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)
};
```

Exercício

4) Para a especificação apresentada em aula, crie a arquitetura do *software*: partindo das classes propostas no exercício 2, verifique o relacionamento estrutural entre elas (associações, herança) e desenhe o diagrama de classes.

Bibliografia

[1] STADZISZ, Paulo Cézar.
Projeto de software usando a UML. Versão 2002.
CEFET-PR