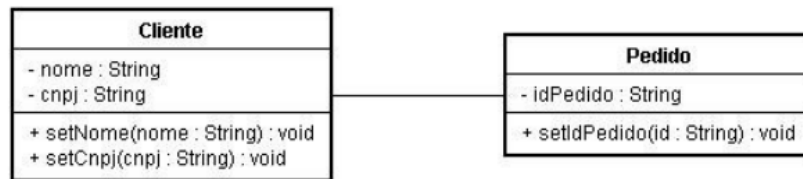


Design de Software

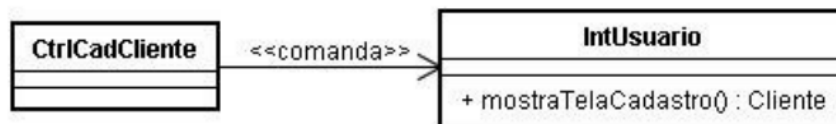
Diagramas de Classes Visão de Projeto - Continuação

- Associações

- Todos os sistemas abrangem muitas classes e objetos, onde os objetos atuam no comportamento de um sistema colaborando entre eles
- A colaboração é realizada através de relacionamentos
- Ocorrem dois tipos principais de relacionamentos
 - Associação
 - Generalização
- Uma associação é uma conexão semântica bidirecional entre classes
- Associações são representadas no diagrama de classes por uma linha ligando as classes associadas, sendo que nesse link os dados podem fluir em ambas as direções



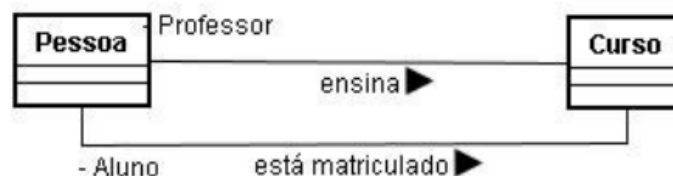
- Navegação: quando se deseja adicionar uma restrição de navegação na associação usa-se uma seta que indica um sentido único da navegação



- Para tornar o significado de uma associação mais claro, uma associação pode receber um nome (geralmente um verbo ou frase verbal)
- Papéis
 - Um papel denota o propósito ou capacidade em que uma classe se associa com outra



- A atribuição de um papel para uma classe numa relação indica que ela está executando mais de um papel no sistema e que ela não está coesa
- Associações Múltiplas
 - Pode existir mais de uma associação entre duas classes, sendo que elas DEVEM ser nomeadas
 - Normalmente devem ser questionadas, pois o melhor seria dividir a classe em outras 2



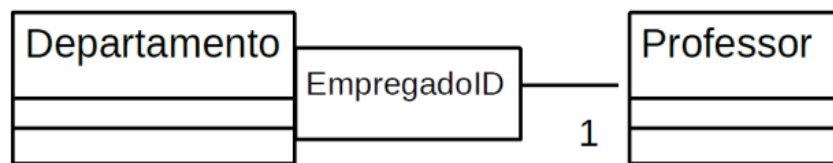
- Multiplicidade para associações
 - Multiplicidade é o número de instâncias de uma classe relacionada a UMA instância da outra classe

- Para cada associação, devem ser tomadas duas decisões de multiplicidade: uma para cada lado da associação

Muitos	<u>*</u>
Exatamente Um	<u>1</u>
Zero or mais	<u>0..*</u>
Um ou Mais	<u>1..*</u>
Zero or um	<u>0..1</u>
Intervalo Específico	<u>2..4</u>

- Qualificadores

- Um qualificador é um atributo de uma classe que pode ser usado para reduzir a multiplicidade da associação

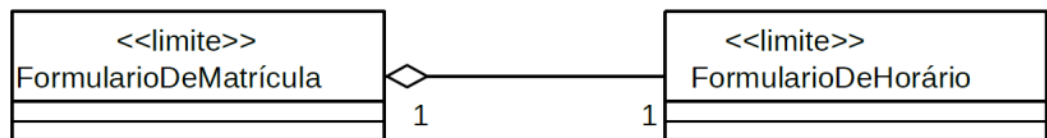


- Restrições

- Uma restrição expressa alguma condição que deve ser preservada, mostrada normalmente entre chaves

- Agregação

- Agregação é uma forma especializada de associação no qual o todo está relacionado a sua(s) parte(s)
- Conhecida como “parte-de” ou relacionamento de conteúdo



- Tipos de agregação

- Agregação por referência
 - Conhecida como relacionamento “tem-um”
 - Envolve partes/componentes que existem independente de seus agregados
 - Exemplo: A área X da empresa tem os empregados A, B , C
 - Representado pelo losango não preenchido

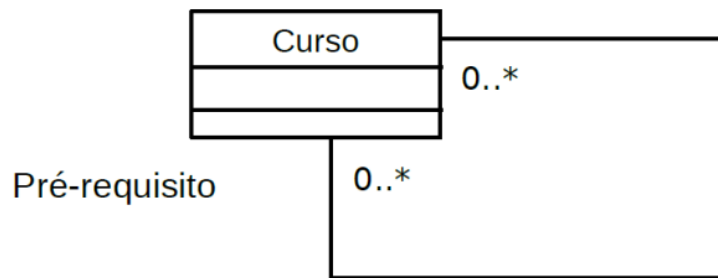
- Composição

- Conhecido como relacionamento “contém um”
- Especifica que as partes/componentes só tem um dono
- Exemplo: um automóvel possui 2 a 5 portas
- Representado pelo losango preenchido

- Associação vs Agregação

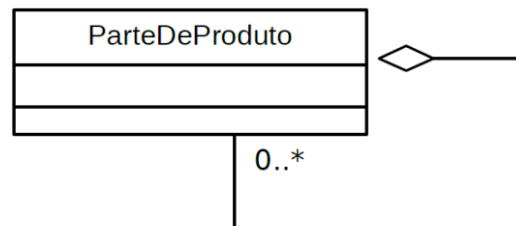
- Se dois objetos são estritamente ligados por um relacionamento parte-todo: agregação
- Se dois objetos são usualmente considerados independentes, mesmo que eles estejam frequentemente ligados: associação

- Associações reflexivas
 - Em uma associação reflexiva, objetos de uma mesma classe são relacionados
 - Exemplo



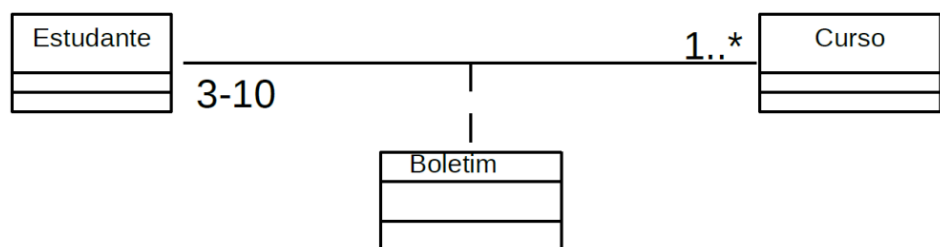
Um curso pode ter muitos pré-requisitos e pode ser um pré-requisito para muitos outros cursos

- Agregados também podem ser reflexivos (relacionamento recursivo)

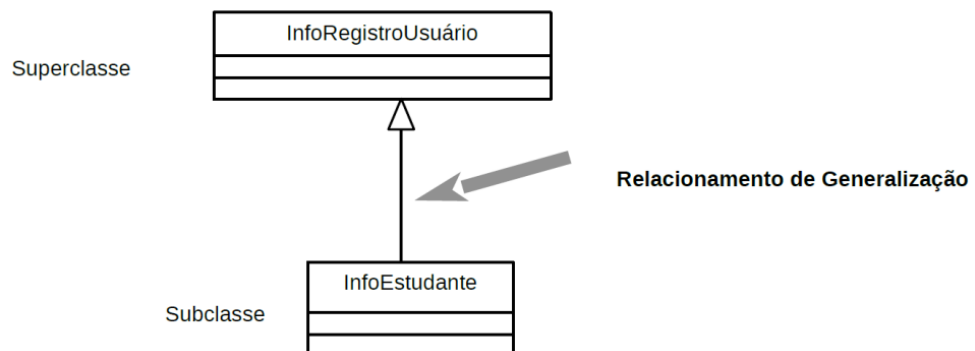


Um objeto **ParteDeProduto** é "composto de" zero ou mais objetos **ParteDeProduto**

- Classe de Ligação
 - Usada para conter a informação de uma ligação

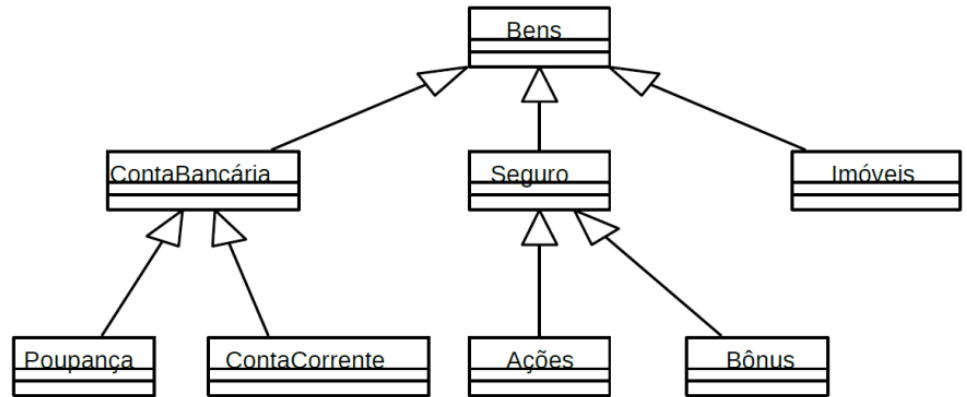


- Generalização
 - Generalização define um relacionamento entre classes onde uma classe compartilha a estrutura e/ou comportamento de uma ou mais classes
 - Com herança simples, a subclasse herda de apenas uma superclasse
 - Com herança múltipla, a subclasse herda de mais de uma superclasse
 - Generalização é um relacionamento "é um" ou "tipo de"



- Na prática, os níveis de hierarquia precisam ser bem limitados
- É herdado da superclasse: atributos, operações, relacionamentos e tipos

- Uma superclasse pode incluir atributos, operações e relacionamentos adicionais, além de redefinir operações herdadas
- Herança de atributos, operações e relacionamentos
 - Subclasses de uma classe herdam tudo, e podem adicionar novos atributos também
- Generalização de classes
 - Proporciona a capacidade de criar superclasses que reúnem estrutura e/ou comportamento comum a várias subclasses
 - Superclasses são mais abstratas que suas subclasses



- Especialização de classes
 - A especialização proporciona a capacidade de criar subclasses que representam refinamentos nos quais a estrutura e/ou comportamento da superclasse são adicionados ou modificados
- Hierarquias de Herança
 - Tanto generalização quanto especialização são usadas no desenvolvimento de uma hierarquia de herança
 - Durante o projeto, as hierarquias de herança são refinadas para aumentar reutilização, incorporar classes de implementação e incorporar bibliotecas de classes disponíveis
- Herança Múltipla
 - Nem todas as linguagens de programação orientadas a objetos suportam herança múltipla diretamente
 - Exemplo

