

UML – Unified Modeling Language

Profº Ms Gustavo Molina

UML –

Linguagem de Modelagem Unificada

- Principais autores do processo: Grady Booch, James Rumbaugh, Ivar Jacobson
- Chamados os 3 amigos
- Aproveitar o melhor das características das notações preexistentes
- *Notação da UML é uma união das diversas notações preexistentes com alguns elementos removidos e outros adicionados com o objetivo de torna-la mais expressiva.*

UML –

Linguagem de Modelagem Unificada

- UML
 - é uma linguagem visual para modelar sistemas Orientados a Objetos
 - Define elementos gráficos que podem ser utilizados na modelagem de sistemas
 - Através dos elementos definidos na linguagem podem-se construir diagramas para representar diferentes perspectivas de um sistema
 - Cada elemento gráfico possui uma
 - Sintaxe: forma predeterminada de desenhá-lo
 - Semântica: O que significa o elemento e com que objetivo deve ser usado
 - A sintaxe e a semântica são extensíveis

UML –

Linguagem de Modelagem Unificada

- UML
 - É independente de linguagens de programação e de processo de desenvolvimento
 - Definição completa:
 - www.uml.org
 - Especificação de leitura complexa voltada a pesquisadores ou desenvolvedores de ferramentas de suporte

UML –

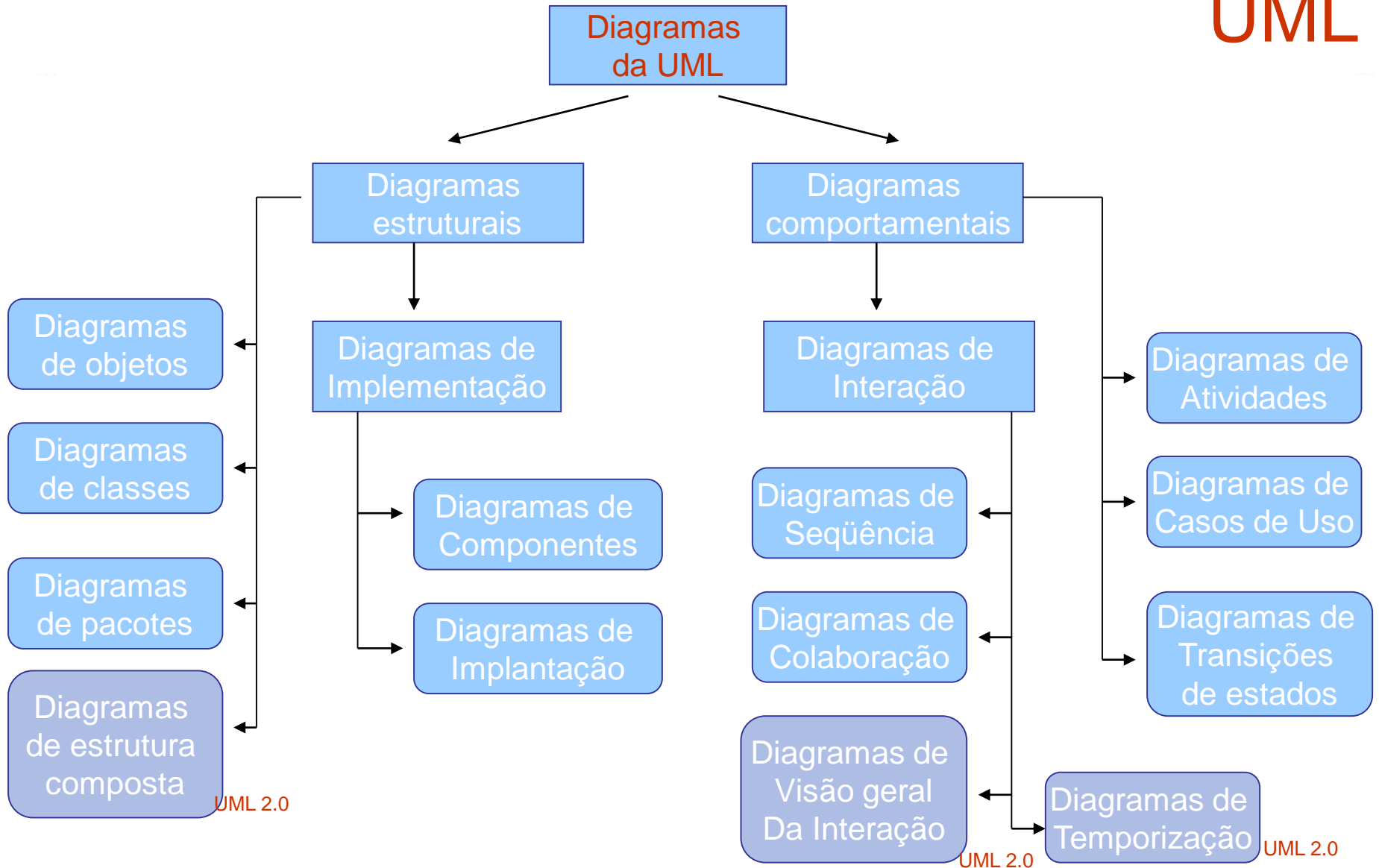
Linguagem de Modelagem Unificada

- Visões de um sistema
 - Um sistema complexo pode ser examinado a partir de diversas perspectivas.
 - Autores da UML definem 5 visões:
 - **Visão de Casos de uso**: Visão externa do sistema que define a interação entre o sistema e agentes externos.
 - **Visão de Projeto**: Características estruturais e comportamentais do sistema.
 - **Visão de Implementação**: gerenciamento de versões construídas pelo agrupamento de módulos e subsistemas.
 - **Visão de Implantação**: Distribuição física do sistema.
 - **Visão de Processo**: Características de concorrência, sincronização e desempenho do sistema.

UML –

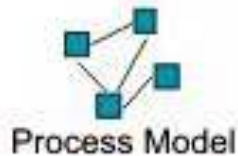
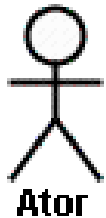
Linguagem de Modelagem Unificada

- Diagramas:
 - Os documentos gerados em um processo de desenvolvimento são chamados de artefatos na UML
 - Os artefatos compõe as visões do sistema
 - A UML define 13 diagramas
 - Esta quantidade de diagramas é justificada pela necessidade de analisar o sistema por meio de diferentes perspectivas
 - Cada diagrama fornece uma perspectiva parcial do sistema.



UML – Mecanismos gerais

- Estereótipos Gráficos
- Estereótipos Textuais



`<<document>>`

`<<interface>>`

`<<entity>>`

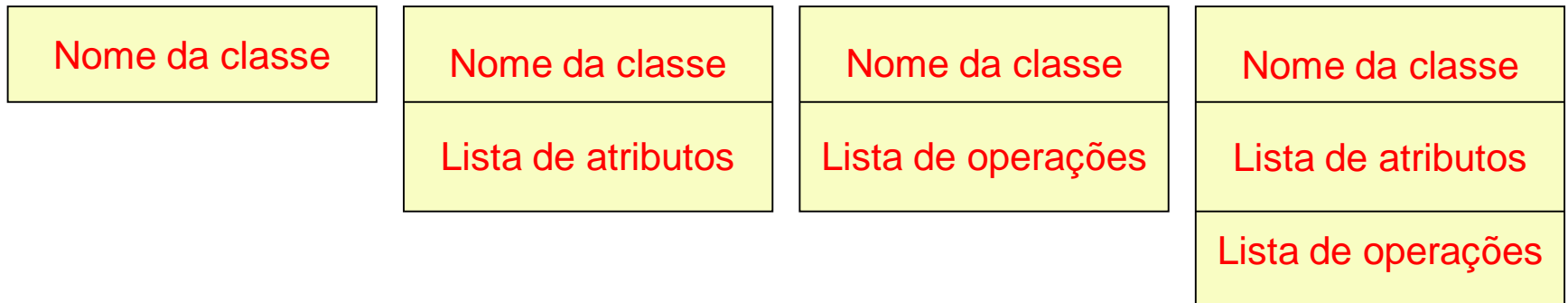
`<<satisfaz>>`

`<<realiza>>`

Diagrama de Classes

- ◆ Classes

- ◆ Representada por uma caixa com 3 compartimentos no máximo:



- ◆ O grau de abstração determina quando usar uma notação

Diagrama de Classes

- ◆ Classes
 - ◆ Exemplo:

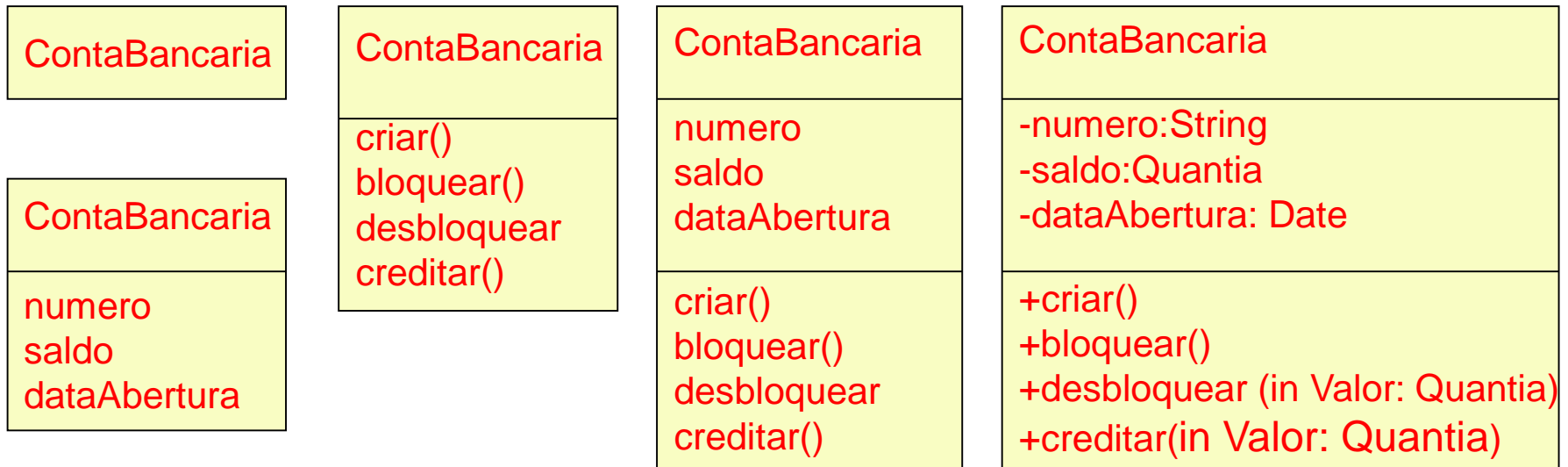


Diagrama de Classes

- ◆ Classes

- ◆ Os atributos correspondem à descrição dos dados armazenados pelos objetos de uma classe.
 - ◆ Cada objeto tem os seus próprios valores
- ◆ As operações correspondem a descrição das ações que os objetos de uma classe sabem realizar.
 - ◆ Objetos de uma classe compartilham as mesmas operações

Diagrama de Classes

◆ Associações

- ◆ Objetos podem se relacionar com outros, possibilitando a troca de mensagens entre eles.
- ◆ O relacionamento entre objetos são representados no diagrama de classes por uma *Associação*.
- ◆ Uma Associação é representada por uma linha ligando as classes.
- ◆ Ex: Um cliente compra produtos

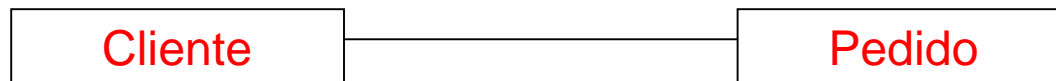


Diagrama de Classes

- ◆ Relacionamentos
 - ◆ Associação
 - ◆ Agregação e Composição
 - ◆ Generalização e Especialização

Diagrama de Classes

- ◆ Associações
 - ◆ Características das associações:
 - ◆ Multiplicidade
 - ◆ Nome
 - ◆ Direção de leitura
 - ◆ Papéis
 - ◆ Tipo de participação
 - ◆ Conectividade

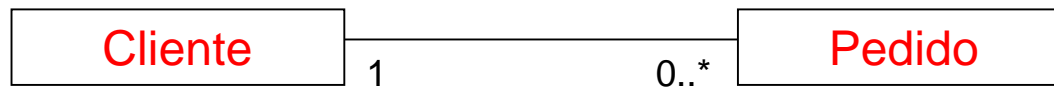
Diagrama de Classes

- ◆ Multiplicidade:
 - ◆ Representa as informações dos limites inferior e superior da quantidade de objetos aos quais outro objeto pode estar associado.

| Nome | Simbologia |
|----------------------|----------------------------|
| Apenas Um | 1 (ou 1..1) |
| Zero ou Muitos | 0..* (ou *) |
| Um ou Muitos | 1..* |
| Zero ou Um | 0..1 |
| Intervalo específico | 1i..1s |

Diagrama de Classes

- ◆ Multiplicidade:

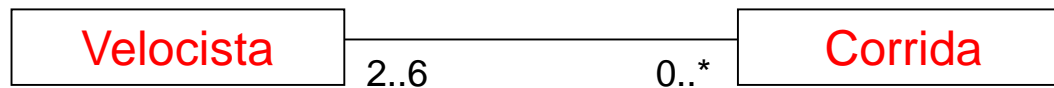


- ◆ Pode haver algum objeto da classe Cliente que está associado a *vários objetos* da classe Pedido (representado por * do 0..*)
- ◆ Pode haver algum objeto da classe Cliente que *NÃO* está associado a classe Pedido (representado por 0 do 0..*)
- ◆ Objetos da classe pedido está associado a **UM** e somente um objeto da classe Cliente

Cliente José tem os pedidos 1, 2 e 3
Cliente Ana tem os pedidos 4 e 5
Cliente Maria não tem pedidos
O pedido 1 está associado somente a José

Diagrama de Classes

- ◆ Multiplicidade:



- ◆ O velocista pode participar de várias corridas (*) ou não participar de nenhuma (0)
- ◆ Em uma corrida deve haver no mínimo DOIS velocistas e no máximo SEIS velocistas
- ◆ Uma lista de intervalos também pode ser especificada na multiplicidade de uma associação. Ex: [1,3,5..9,11]
- ◆ Os valores especificados em uma multiplicidade devem sempre estar em ordem crescente.

Diagrama de Classes

- ◆ Multiplicidade:

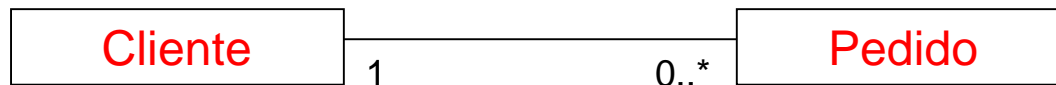
- ◆ As associações podem ser agrupadas em 3 tipos. Estes tipos são denominados *Conectividade*:

| Conectividade | Multiplicidade de um extremo | Multiplicidade do outro extremo |
|--------------------|------------------------------|---------------------------------|
| Um para Um | 0..1 ou 1 | 0..1 ou 1 |
| Um para Muitos | 0..1 ou 1 | * ou 1..* ou 0..* |
| Muitos para Muitos | * ou 1..* ou 0..* | * ou 1..* ou 0..* |

Diagrama de Classes

◆ Participações

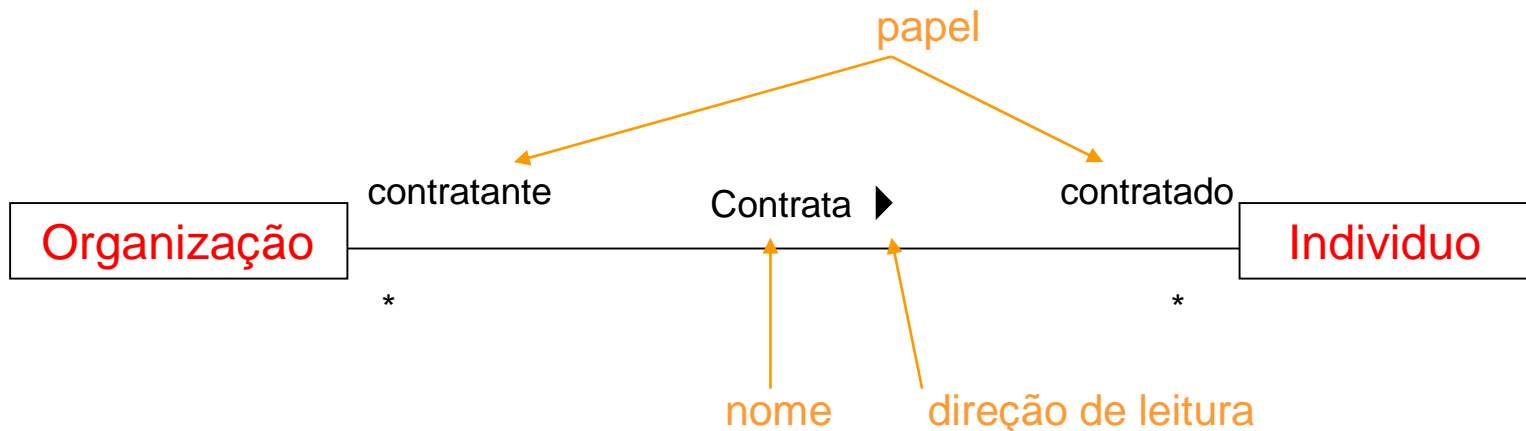
- ◆ Necessidade ou não da existência dessa associação entre objetos.
- ◆ Obrigatória:
 - ◆ Se o valor mínimo da multiplicidade é igual a **Um**
- ◆ Opcional
 - ◆ Se o valor mínimo puder ser **Zero**



Para objetos da classe pedido a participação é **obrigatória**: Um objeto da classe Pedido só existe se estiver associado a classe Cliente.

Diagrama de Classes

- ◆ Nome da associação, direção de leitura e papéis
 - ◆ Servem para esclarecer melhor o significado de uma associação
 - ◆ Só usar quando o significado de uma associação não for clara. Evitar usar em associações claras ou óbvias.



- ◆ Uma organização (faz o papel de contratante) contrata indivíduos (faz o papel de contratado)

Diagrama de Classes

- ◆ Nome da associação, direção de leitura e papéis
 - ◆ Podemos representar mais de uma associação entre objetos



- ◆ Uma organização precisa saber quem são os empregados e quem é o gerente

Diagrama de Classes

◆ Classes Associativas

- ◆ Classes ligadas a associações em vez de estar ligada a outras classes.
- ◆ Necessário quando se quer manter informações sobre a associação de duas ou mais classes.
- ◆ Pode estar ligada associação de qualquer conectividade.
- ◆ Pode ser substituída por uma classe com associação para as outras duas classes.

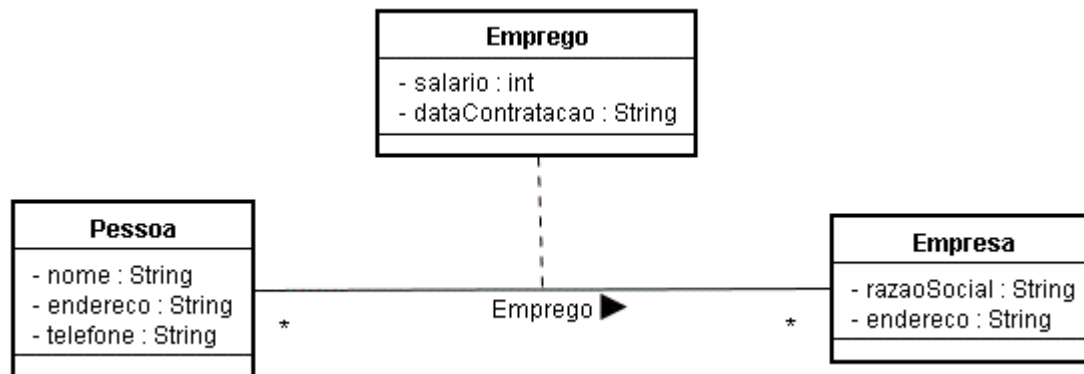


Diagrama de Classes

- ◆ Associações reflexivas (auto-associação)
 - ◆ Associa objetos da mesma classe
 - ◆ Cada objeto tem um papel distinto na associação
 - ◆ O uso de papéis é importante neste caso

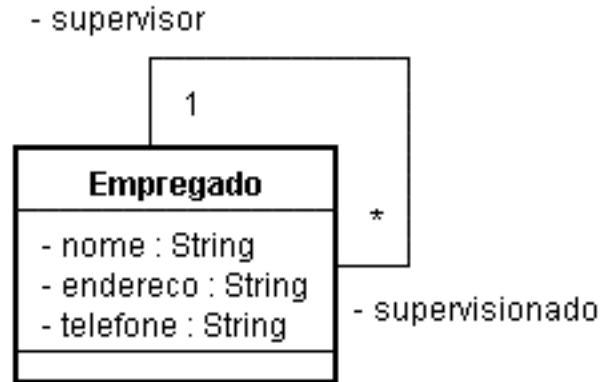


Diagrama de Classes

- ◆ Generalizações e Especializações
 - ◆ Usa-se vários termos: SuperClasse e SubClasse, Supertipo e SubTipo, Classe Base e Classe Herdeira.
 - ◆ Representa o conceito de Herança.
 - ◆ Não somente atributos e operações são herdados, mas as associações também.
 - ◆ Notação:

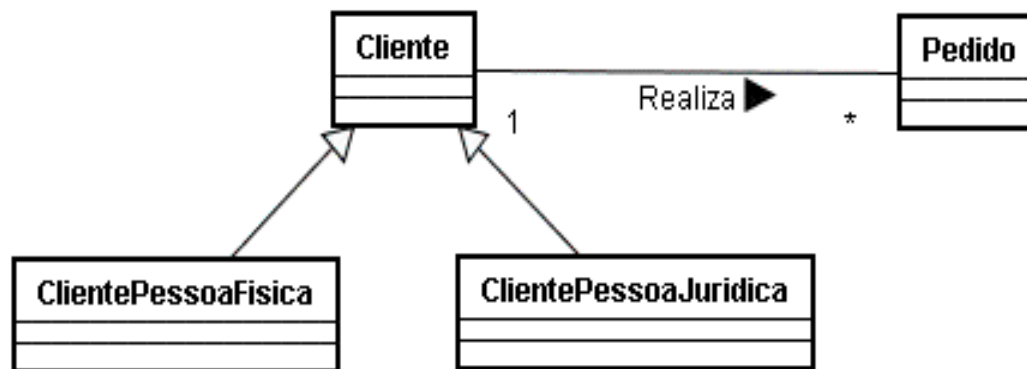
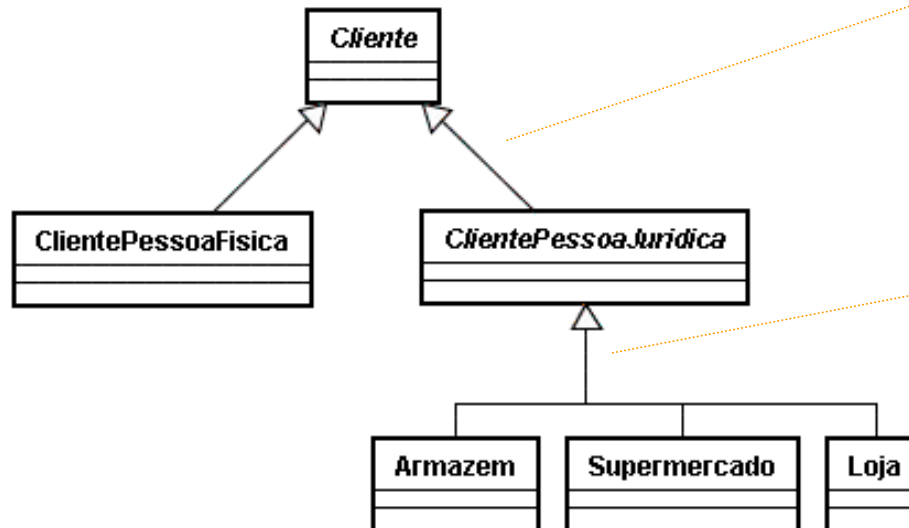


Diagrama de Classes

- ◆ Generalizações e Especializações
 - ◆ Classes Abstratas:
 - ◆ É usada para organizar a hierarquia de classes.
 - ◆ Não geram objetos diretamente
 - ◆ Muito utilizada nas Classes de Projetos
 - ◆ Notação: O nome é definido em *Itálico*



Esta notação é
igual a
esta

Diagrama de Classes

- ◆ Herança X Associação
 - ◆ O relacionamento de herança acontece entre classes
 - ◆ Os relacionamentos de Associação, Agregação / Composição e Associação ocorre entre as instâncias das classes (os objetos).
- ◆ Propriedades de relacionamentos de herança
 - ◆ Transitividade
 - ◆ Se A é uma generalização de B e B é uma generalização de C, então C herda características de B e A.
 - ◆ Assimetria
 - ◆ Se A é uma generalização de B, B não pode ser uma generalização de A
- ◆ Deve-se evitar hierarquias muito profundas, com mais de 3 níveis, pois dificulta a leitura.