Fatec Ourinhos

Curso: Análise e Desenvolvimento de Sistemas

Profa. Me. Viviane de Fatima Bartholo

Email: viviane.bartholo@fatecourinhos.edu.br

Fatec
Ourinhos

Introdução a Modelagem de Software

UML – Unified Modeling Language

Fatec
Ourinhos

Diagrama de Classes
e
Diagrama de Objetos

Fatec Ourinhos

Introdução

- **Externamente ao sistema, os usuários visualizam resultados de cálculos, relatórios produzidos, confirmações de requisições, etc.**
- **Internamento, o sistema orientado a objetos é composto por um conjunto de objetos que cooperam entre si.**
- **Cooperação:**
  - Aspecto estrutural : Apresenta como o sistema está internamente estruturado.
  - Aspecto dinâmico: Apresenta as interações entre os objetos.
- **O aspecto estrutural de um sistema é representado pelo *diagrama de classes*.**

Fatec Ourinhos

Introdução

- **Desenvolvimento inclui:**
  - Requisitos –> Análise –> Projeto -> Implementação
- **O Modelo de classes (MC) evolui durante o desenvolvimento iterativo**
- **Estágios de abstração**
  - Análise
    - Atenção sobre o que o sistema deve fazer
  - Especificação
  - Implementação

Fatec Ourinhos

Introdução

- **Classes de Análise (MCA)**
  - Atenção sobre o que o sistema deve fazer
  - Não leva em consideração restrições associadas a tecnologia a ser utilizada na resolução de um problema
  - O MCU e o MCA são os dois principais modelos da fase de análise
- **Classes de especificação(MCE)**
  - É um detalhamento do modelo de classes de análise
  - É também conhecido como Modelo de classes de projeto
  - A atenção é sobre como o sistema deve funcionar
  - Novas classes começam a aparecer
  - Ex: Analogia com uma casa: Classes de análise são salas, quartos, banheiro, porta. Classes de projeto são encanamento, parte elétrica, encaixe das portas.
  - Parte visível X parte menos evidente do modelo

Introdução

- **Classes de implementação (MCI)**
  - É a implementação das classes em uma linguagem de programação (C, Java, C#, etc.)
  - Construído na implementação.
  - É o próprio código fonte como um modelo.

- **O nível de abstração diminui a cada estágio**

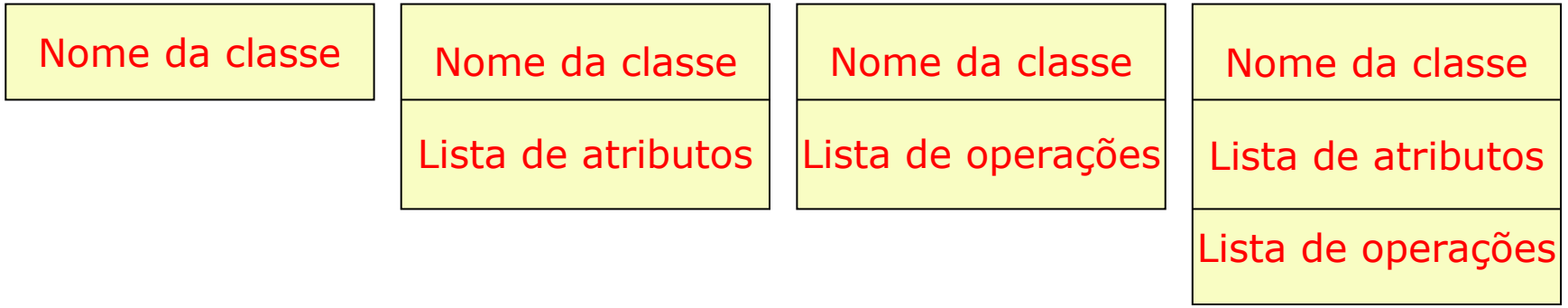
Fatec Ourinhos

Introdução

- **Notação para o MC (recomendações)**
  - Identificadores: Espaços em branco e preposições são removidas
  - Nomes de classes e relacionamentos:
    - Palavras começando por letra maiúscula
    - Ex: Cliente, Pedido, ItemPedido
  - Nomes de atributos e operações
    - Palavras começando com letra minúscula
    - Duas (ou mais) palavras separadas por letra maiúscula
    - Siglas inalteradas
    - Ex: quantidade, precoUnitario, CPF

Diagrama de Classes

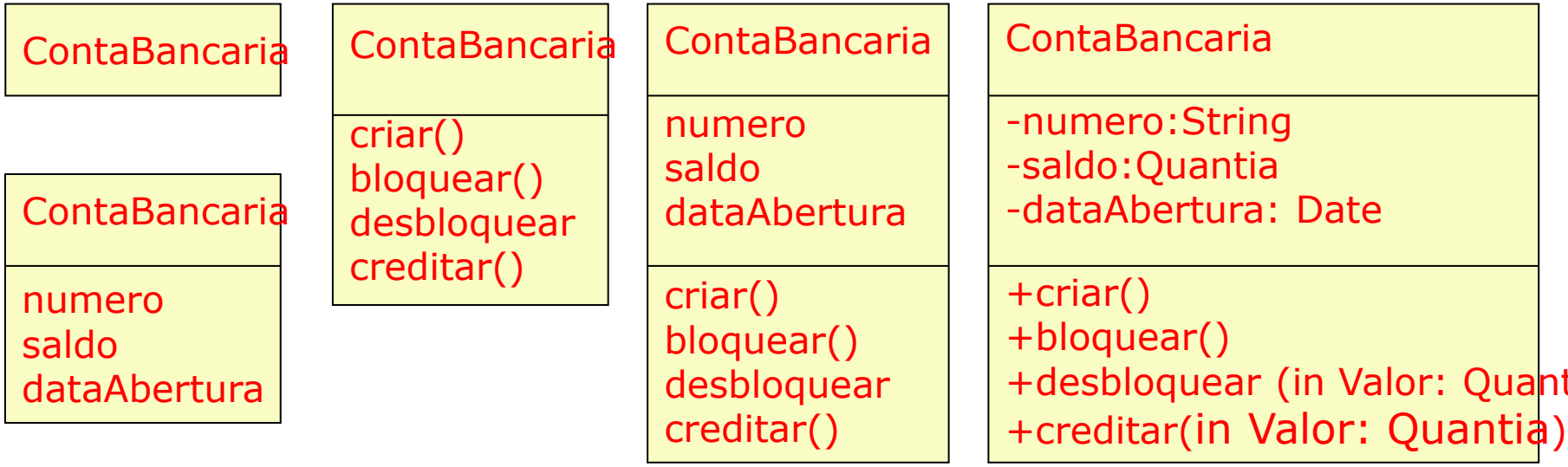
- **Classes**
  - Representada por uma caixa com 3 compartimentos no máximo:



  - O grau de abstração determina quando usar uma notação

Diagrama de Classes

- **Classes**
  - Exemplo:



Fatec Ourinhos

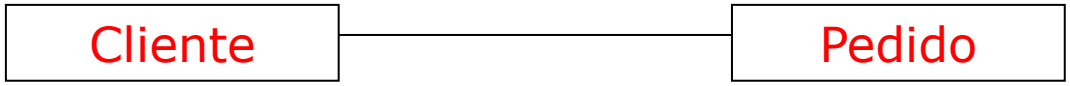
Diagrama de Classes

- **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

Fatec Ourinhos

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



Fatec Ourinhos

Diagrama de Classes

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

Fatec Ourinhos

Diagrama de Classes

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

Fatec Ourinhos

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.

Fatec Ourinhos

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..***

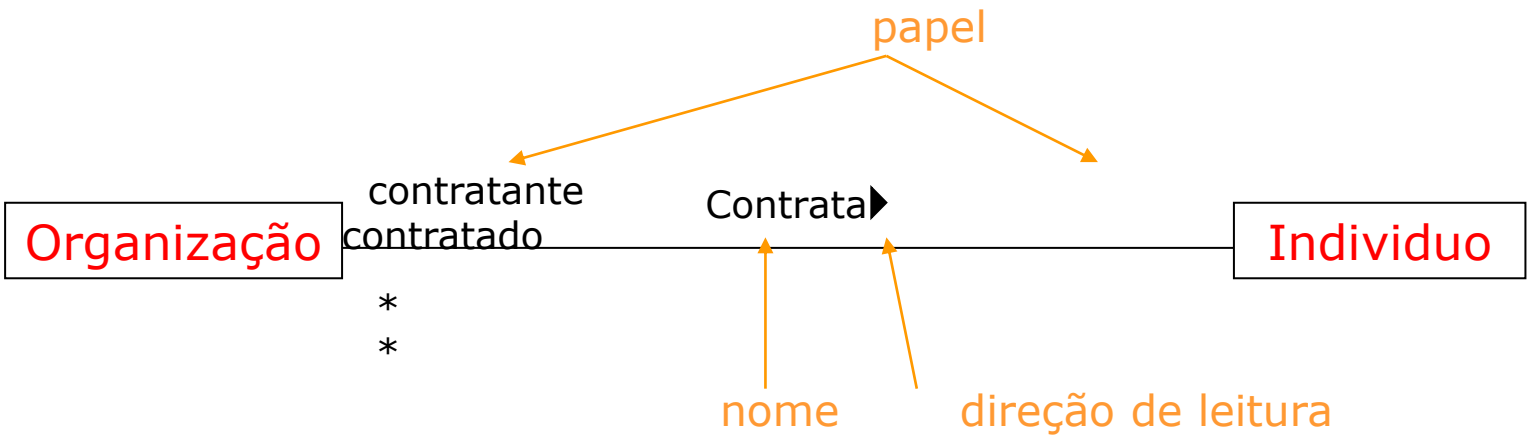
Diagrama de Classes

- 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

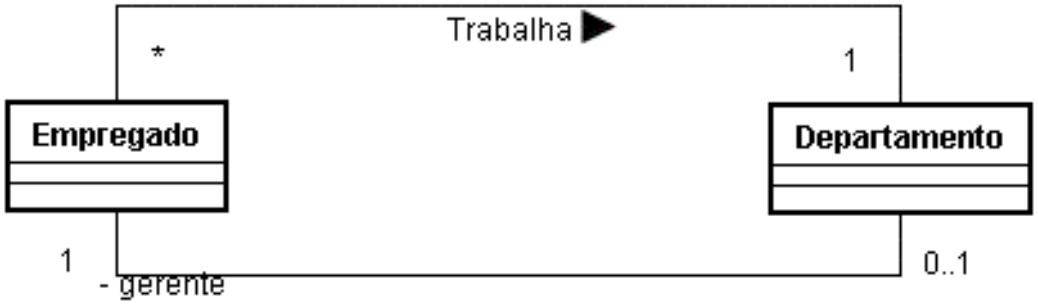
- Nome da associação, direção de leitura e papéis
  - Servem para esclarecer melhor o significado de uma associação
  - Só usar quando o significado de uma associação não for clara. Evitar usar em associações claras ou óbvias.



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

Diagrama de Classes

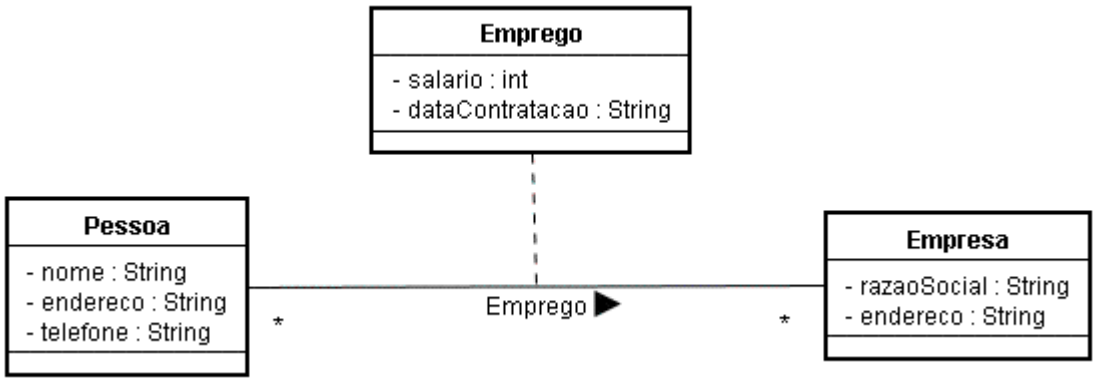
- Nome da associação, direção de leitura e papéis
  - Podemos representar mais de uma associação entre objetos



  - 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.



Uma pessoa trabalha como empregado em várias empresas. Para cada relação empregado empregador, é possível saber o salário e a data de contratação

Diagrama de Classes

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

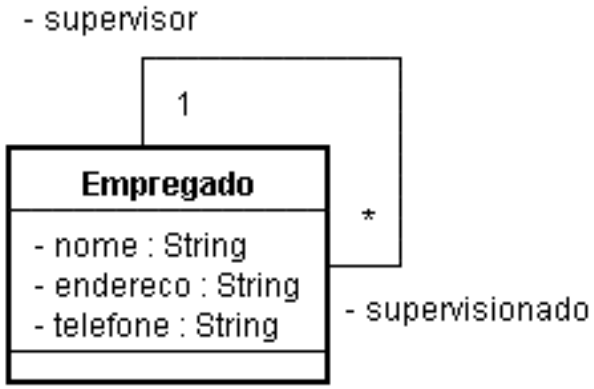
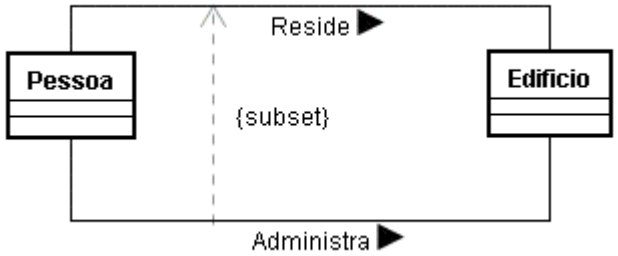
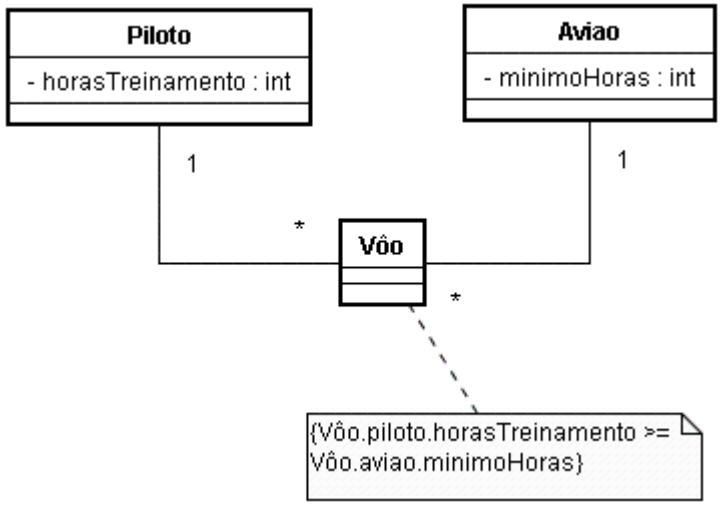


Diagrama de Classes

- Restrições sobre as associações



Objetos da associação administra são um subconjunto dos objetos da associação Reside



Usando OCL

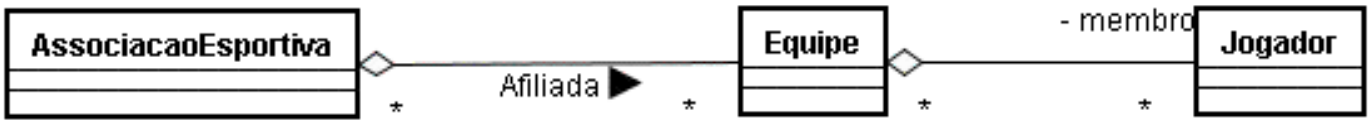
Fatec Ourinhos

Diagrama de Classes

- Agregações e Composições
  - Representa uma relação *todo-parte*
  - Uma relação *todo-parte* significa que um objeto está contido em outro. Ou um objeto contém outro.
  - Características:
    - São assimétricas: Se A é parte de B, B não pode ser parte de A
    - Propagam comportamentos: O comportamento do todo se aplica as partes.
    - As partes são normalmente criadas e destruídas pelo todo. Isto é no Todo são definidas as operações de *Adicionar* e *Remover* as partes.
  - Tipos de relacionamentos todo-parte:
    - Agregação
    - Composição

Diagrama de Classes

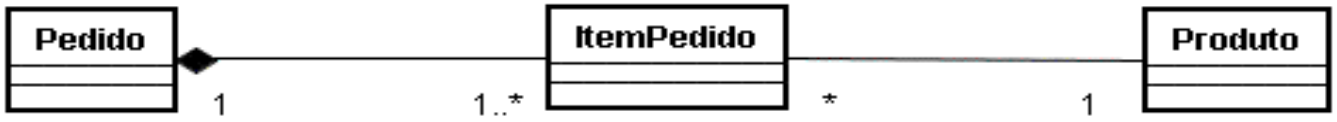
- Agregações
  - Notação:



  - Uma associação é formada por diversas equipes. Cada Equipe é formada por diversos Jogadores.

Diagrama de Classes

- Composições
  - Notação:



  - Um pedido inclui vários itens. Cada item diz respeito a um produto.

Fatec Ourinhos

Diagrama de Classes

- Agregações x Composições
  - As diferenças não são muito bem definidas
  - Diferenças mais marcante:
    - Na agregação, a destruição do objeto Todo não implica na destruição do objeto Parte. Na composição a destruição do Todo implica na destruição das partes.
      - *Ex: Se uma equipe deixar de existir o jogador ainda pode continuar a existir.*
    - Na composição, os objetos parte pertencem a um único todo. Por outro lado na agregação pode ser que um objeto parte participe como componente de vários outros objetos.
      - *Ex: Um item de produto só pode pertencer a um único pedido.*

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:

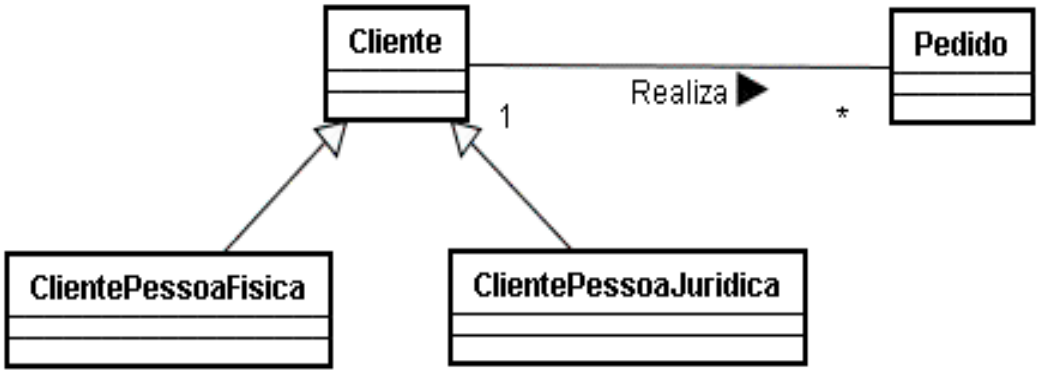


Diagrama de Classes

- 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*

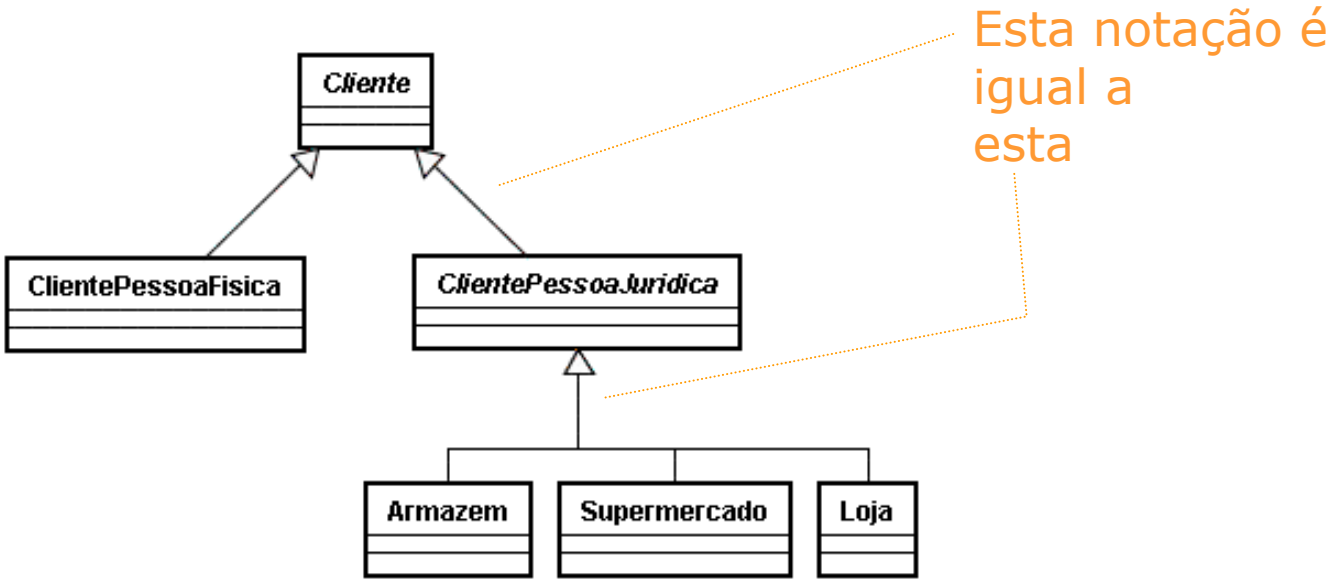


Diagrama de Classes

- 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.

Diagrama de Classes

- Restrições de Generalização e Especialização:
  - Sobreposta: Podem ser criadas subclasses que herdem de mais de uma subclasse
    - Ex: Atleta – (Nadador e Corredor)
  - Disjunta: As subclasses só podem herdar de uma subclasse
    - Ex: Figura geométrica – (Elipse, Quadrado, Circulo)
  - Completa: Todas as subclasses possíveis foram enumeradas.
    - Ex: Indivíduo – (Homem e Mulher)
  - Incompleta: Nem todas as subclasses foram enumeradas na hierarquia
    - Ex: Figura geo culo)

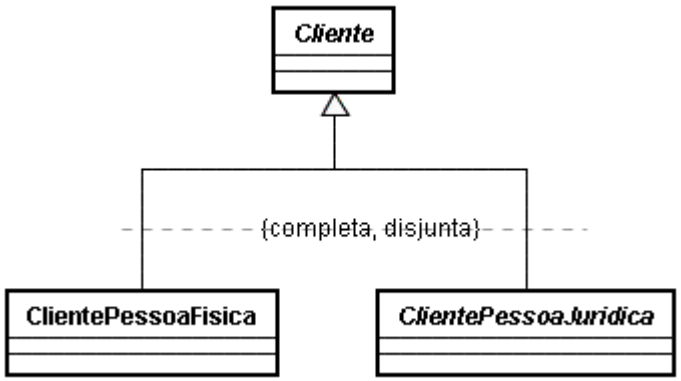


Diagrama de Objetos

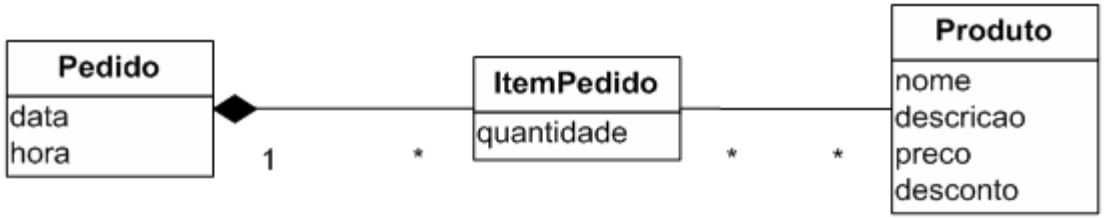
- São instâncias dos diagramas de classes, assim como os objetos são instâncias das classes.
- São estruturas estáticas
- Notação: Definido como “Nome do objeto” + “: (dois pontos)” + “Nome da classe”

O nome da objeto é opcional

Formato	Exemplo
:NomeClasse	:Pedido
nomeObjeto: NomeClasse	umPedido: Pedido

Diagrama de Objetos

- Exemplo: Diagrama de Classes



- Diagrama de Objeto

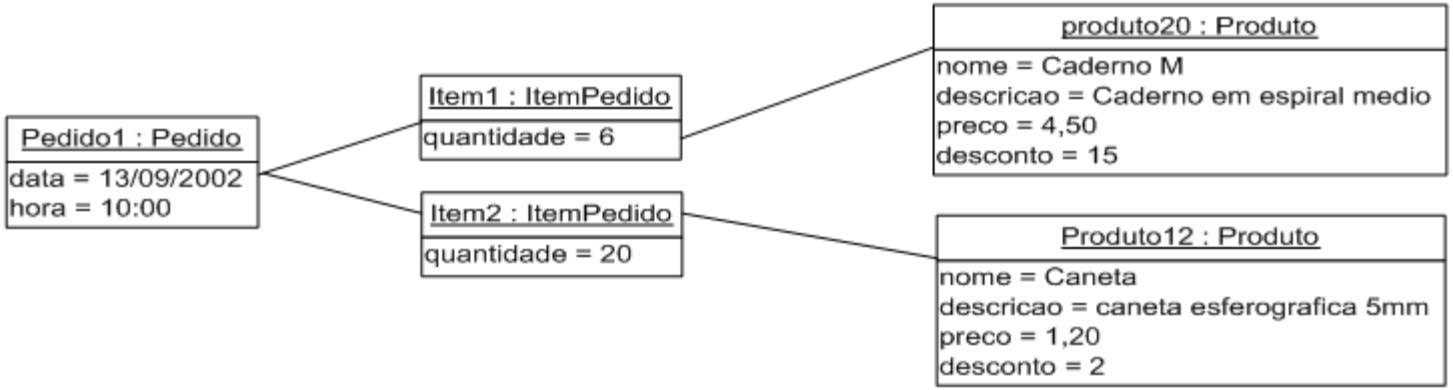
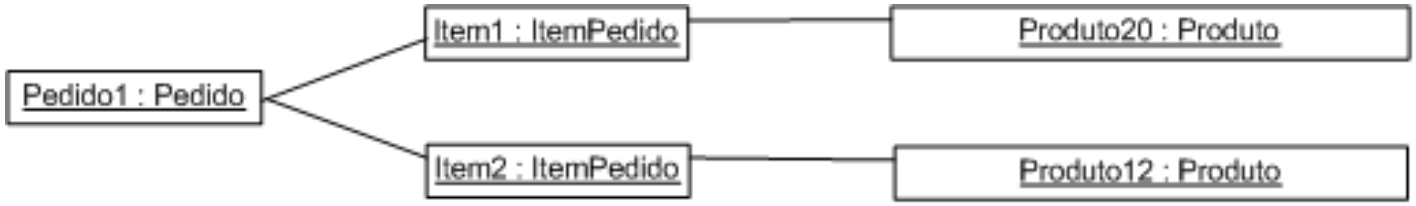


Diagrama de Objetos

- Exemplo 2: Diagrama de objetos



- A utilidade prática dos diagramas de objetos é ilustrar a formação de relacionamentos complexos

DÚVIDAS??

Fatec Ourinhos

Bibliografia

- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivan. UML: guia do usuário. Rio de janeiro: Campus, 2000. 472p.
- Guedes, G., UML 2.0 - Uma Abordagem Prática . Rio de Janeiro : Novatec /2009.
- Booch, G., Rumbaugh, J. and Jacobson, I., *Unified Modeling Language User Guide*, 2nd Edition, Addison-Wesley Object Technology Series, 2005.
- P. J. DEITEL, H. M. DEITEL. C++ Como Programar. Quinta edição. Pearson, 2006.
- P. J. DEITEL, H. M. DEITEL. Java Como Programar. Oitava edição. Pearson, 2010.
- B. MEYER. Object-Oriented Software Construction. Segunda Edição. PrentticeHall, 1997.
- Nota de aula de Marco Antonio Moreira de Carvalho. Disponivel em http://www.decom.ufop.br/marco/ensino/bcc221/material-das-aulas
- Notas de aula Prof. Thiago Affonso de M. N. Viana thiago_affonso_viana@yahoo.com.br