

Curso de Ciência da Computação UNIVALI

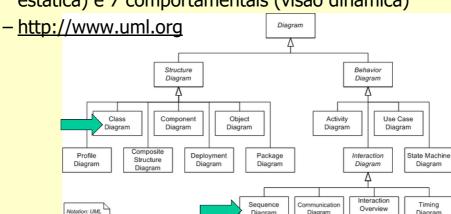
Algoritmos e Programação de Computadores 2per Unified Modeling Language (UML)

Profa. Fernanda dos Santos Cunha

(Material baseado no livro UML - Guia Do Usuário, Booch, e nos slides POO 3p, prof. Thiry)

UML

- UNIVALI
- Linguagem de Modelagem Unificada
 - Padrão OMG (*Object Management Group*), 1997.
 - 14 diagramas, sendo 7 estruturais (visão estática) e 7 comportamentais (visão dinâmica)



Unified Modeling Language **UML**

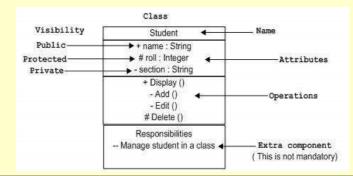


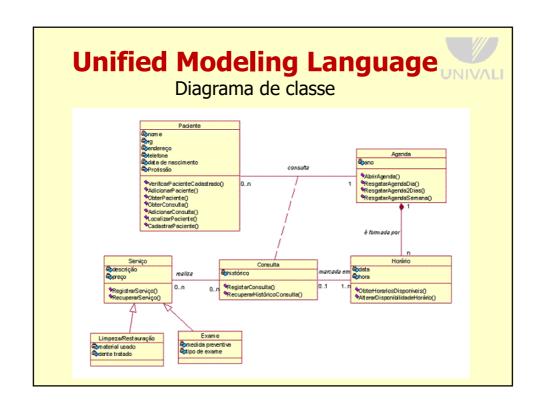
- É um modelo de linguagem para modelagem de sistemas orientados a objetos (principalmente).
- Modelagem visual de maneira que os relacionamentos entre os componentes do sistema sejam melhor compreendidos e documentados.
- Linguagem para especificar, construir, visualizar e documentar um sistema de software que surgiu com a fusão das metodologias já anteriormente usadas.

Diagrama de Classe



- Descreve estrutura de um sistema, pois permite visualizar:
 - Classes do sistema, seus atributos e operações;
 - Relacionamento entre classes; e
 - Relacionamentos entre classes e interfaces.





Principais relacionamentos UNIVALI



- Generalização/especialização (herança) √

- Associação
- Agregação e Composição

Associação



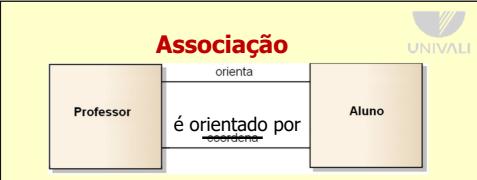
 Relação entre ocorrências (objetos) das classes



- Objetos "Professor" estão associados com objetos "Aluno"
- Mas qual é esta relação ???

Associação orienta Professor coordena Aluno

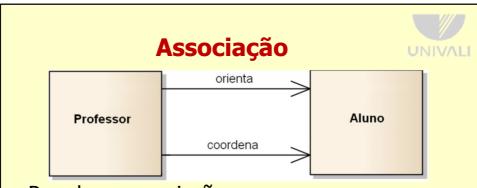
- Nome da associação indica a semântica da conexão entre os objetos, e deve ser sempre uma frase verbal
- Note que a semântica é diferente nas associações acima



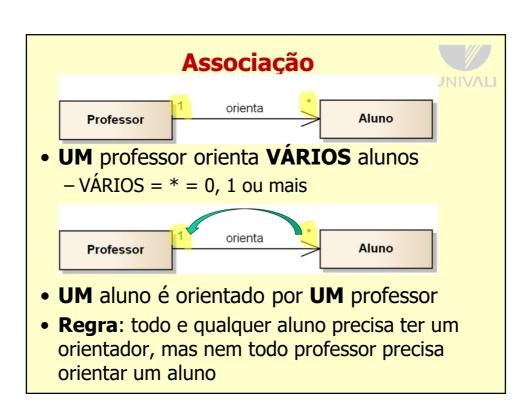
- E neste caso, a semântica ainda é diferente ???
- Se a semântica é a mesma, então é a mesma associação!!!



- Quem orienta quem ?
- Quem coordena quem ?
- Deve-se conhecer o negócio para indicar o sentido da leitura



- Para ler a associação:
 - Professor orienta Aluno
 - Professor coordena Aluno
- Mas, um professor pode orientar quantos aluno ?



Multiplicidade

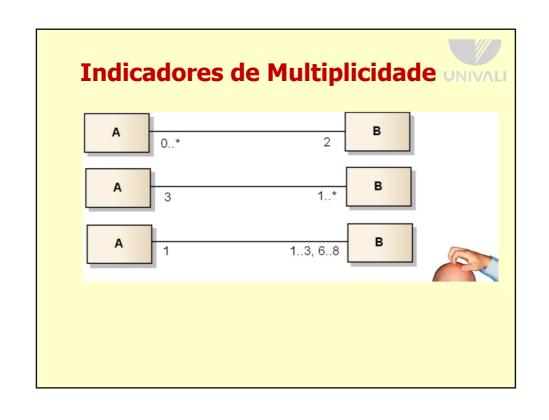


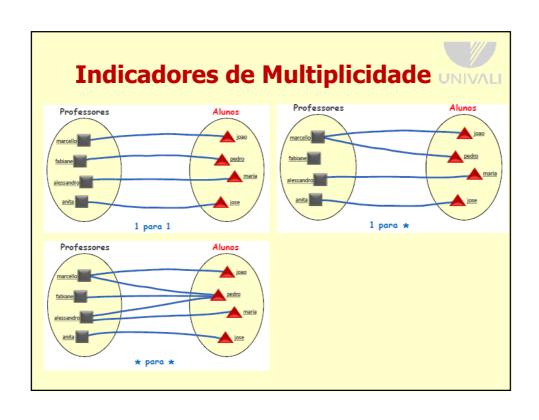
- Define quantos objetos participam em um relacionamento
- O número de objetos de uma classe relacionada a UM objeto da outra classe
- Deve ser especificada em cada lado da associação
- Multiplicidade x Cardinalidade
 - Cardinalidade: número de elementos em um conjunto
 - Multiplicidade: especificação do intervalo de valores de cardinalidade (tamanho) permitidas que um conjunto pode assumir

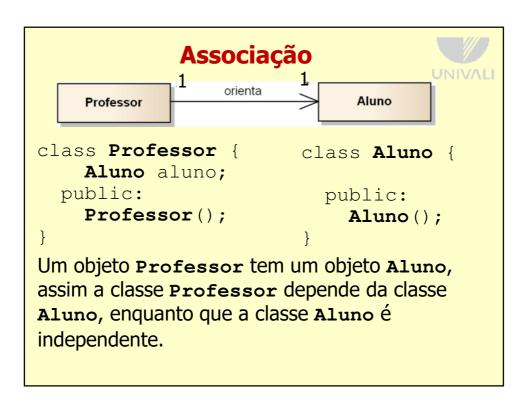
Indicadores de Multiplicidade UNIVALI



- Vários/Muitos/Zero, um ou mais: *
- Um ou mais: 1..*
- Zero ou um: 0..1
- Exatamente um: 1
- Exatamente três: 3
- Faixa especificada: 2..4
- Múltiplas faixas: 2..4, 6..8

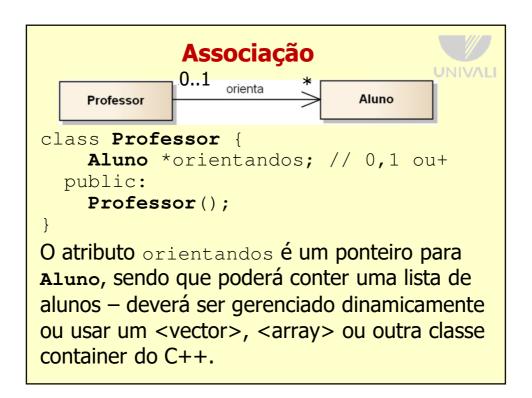








- Nem todo professor precisa ter um orientando (*=0, 1 ou +)
- Um professor pode ter vários orientandos (*=0, 1 ou +)
- Um aluno pode ter ou n\u00e3o um professor orientador, mas nunca mais de um orientador (0..1)



Navegabilidade

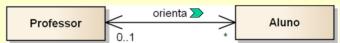


- Os dados podem fluir em uma ou ambas direções pela associação
 - Canal de comunicação pelo qual objetos conversam (trocam msgs)
 - Uma mensagem pode ser uma requisição por informação ou para executar uma ação (objetos chamador invoca uma operação de um objeto receptor)
- Qdo há associação bidirecional o nome valerá apenas para uma das direções !!



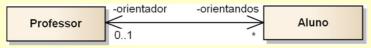
Navegabilidade

 Mas, na implementação teremos que considerar um atributo para cada lado, considerando os papéis assumidos:



orientandos Aluno[] orientador Professor

 Logo, deve-se nomear uma associação com nomes de papel (*role names*), abandonando o nome da associação:



(-) significa visibilidade privada

Associação

• Assim, quanto a representação da classe:



 Haveria duplicação de atributos com o mesmo nome. Então, não deve-se colocar sua implementação nas classes do diagrama:



Problemas com associação bidirecional

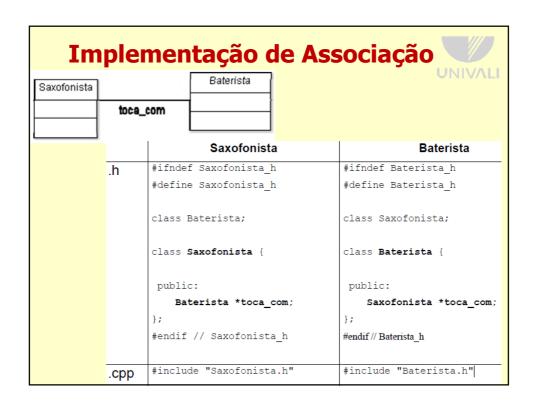


- Imagine um cenário onde o professor X orienta 2 alunos Y e Z
 - Iniciar criando o professor
 - Se o professor for obrigado a ter um orientando, haverá uma situação inconsistente
 - Então poder-se-ia criar o objeto Aluno antes e repassá-lo ao Professor no momento da instanciação
 - Mas, e se um Aluno tb tivesse que ter obrigatoriamente um orientador ???

Considerações com associação bidirecional



- Aumenta o acoplamento (dependência entre classes), reduzindo reusabilidade.
- Aumenta a complexidade da implementação, pois exigem que o sincronismo seja mantido nos dois lados da associação.
- Associação bidirecional definida como vários para vários aumenta ainda mais a complexidade da implementação.

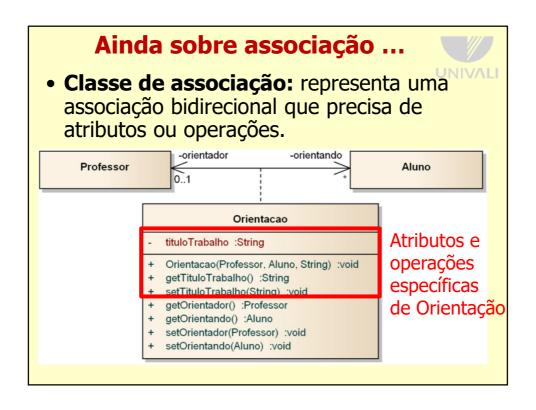


Ainda sobre associação ...





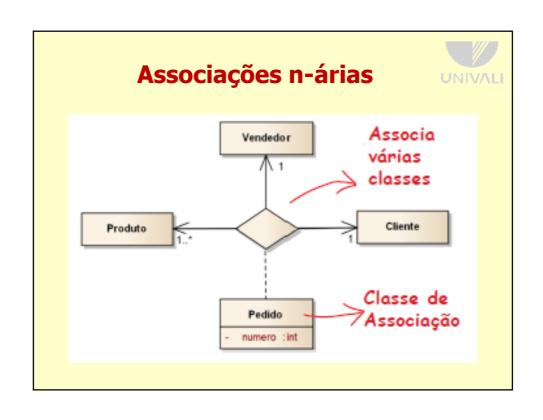
- Voltando ao exemplo acima, e se for preciso informações adicionais que deveriam estar na associação ?
- Se for necessário armazenar o título do TCC.
- Note que TCC não é uma informação do aluno e nem do professor, mas da associação de orientação entre estas duas classes

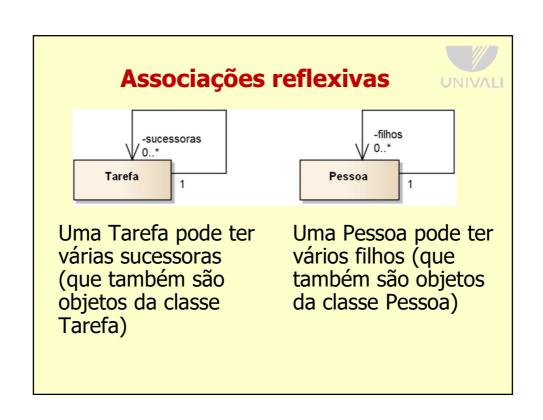


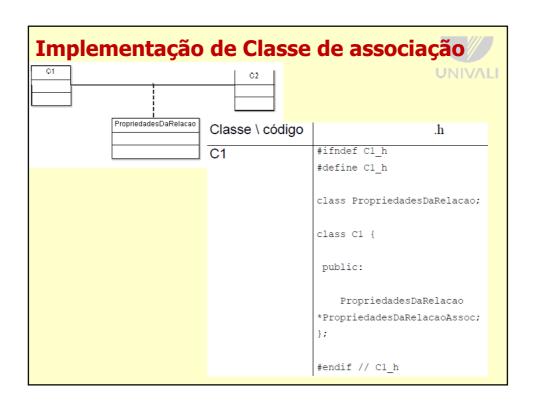
Classe de associação

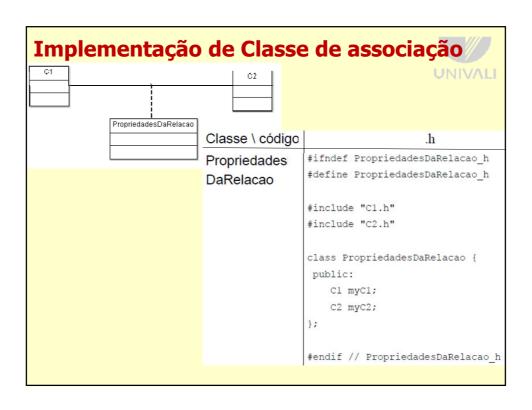


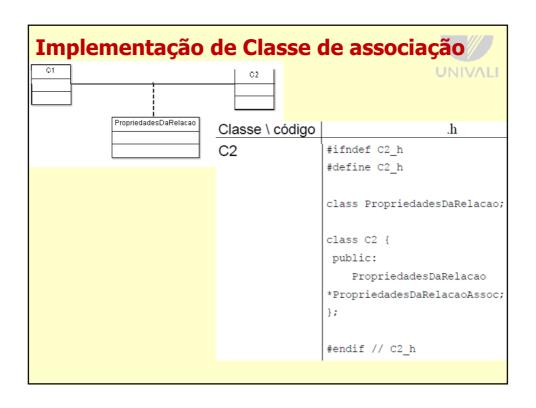
- Quando a associação precisa de atributos ou operações deve-se criar uma classe de associação.
- Note que utilizá-la acrescenta uma restrição (Fowler e Scott, 2003):
 - Poderá haver apenas uma instancia da classe de associação entre quaisquer objetos participantes
 - No exemplo, não poderiam existir 2 instancias de orientação que relacionem o mesmo professor X e aluno Y (mesmo que a associação seja vários para vários)



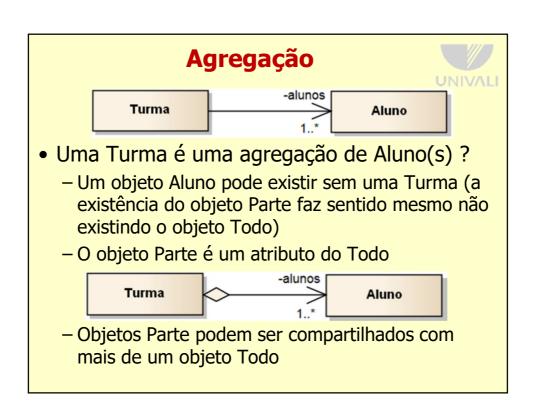


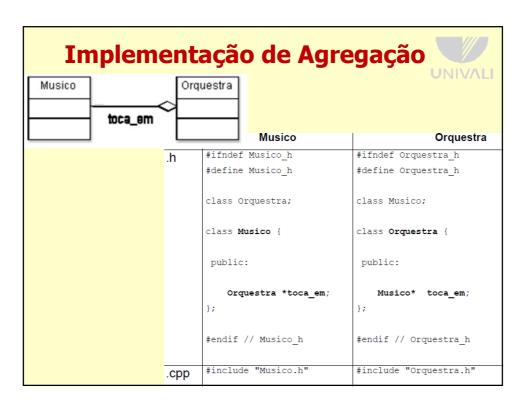


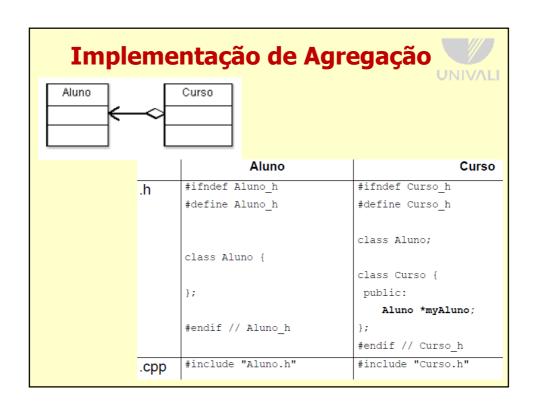










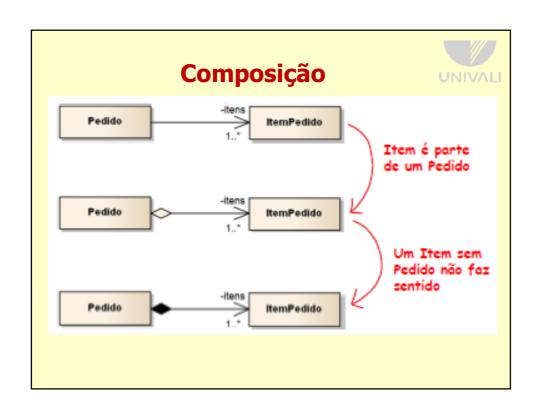


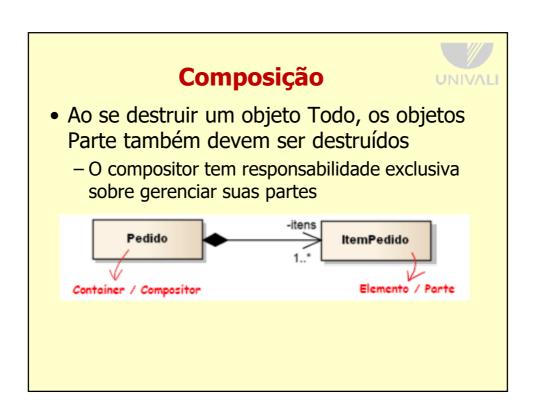
Composição



- Forma mais forte de Agregação
- Relacionamento de "contenção", onde as partes não podem ser compartilhadas
 - Um objeto Todo contém um ou mais objetos Parte
 - Um objeto Parte está contido em, pelo menos, um objeto Todo
 - Um objeto Parte sozinho não faz sentido







Composição

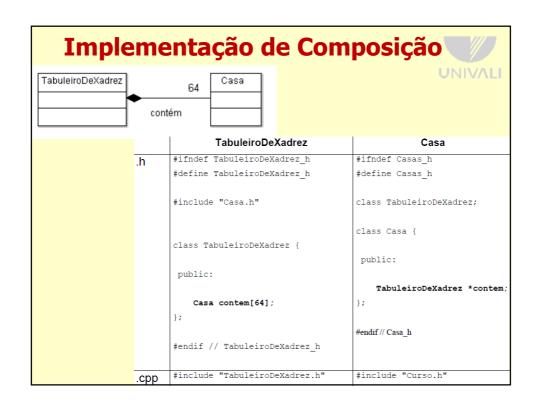


- Quando usar ?
 - Relacionamento é descrito com uma frase
 parte de:
 - Um botão é parte de uma janela
 - Algumas operações no Todo são automaticamente aplicadas a suas Parte(s).
 - Mover a janela, mover o botão
 - Excluir o pedido, excluir todos os itens do pedido

Composição



- Quando usar ?
 - Valores de atributos do Todo s\(\tilde{a}\)o propagados para todas ou algumas de suas Parte(s):
 - A fonte da janela é Arial, a fonte do botão é Arial
 - Existe uma assimetria inerente no relacionamento onde uma classe é subordinada a outra.
 - Um botão é parte de uma janela, uma janela NÃO é parte de um botão



Referências:



- Grady Booch, James Rumbaugh, Ivar Jacobson.
 UML Guia do Usuário. Tradução da 2ª ed.
 Elsevier, 2012.
- OMG (Object Management Group), OMG Unified Modeling Language v2.5, 2012. Disponível em http://www.omg.org/spec/UML/2.5/Beta1/PDF/
- Ricardo Pereira e Silva. UML 2 em Modelagem
 Orientada a Objetos. Visual Books, 2007.
- Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language. 3th ed. Addison-Wesley, 2003.