

ASSOCIAÇÕES ENTRE CLASSES

1. APRESENTAÇÃO

Nesta aula iremos abordar e exercitar o que são associações entre classes e como usá-las. Iremos abordar aspectos como: associações, nomes, papéis, multiplicidade, qualificadores, navegabilidade e associações reflexivas.

2. ASSOCIAÇÕES

A Necessidade de Relacionamentos

Todos os sistemas abrangem muitas classes e objetos

Objetos atuam no comportamento de um sistema colaborando entre eles

- A Colaboração é realizada através de relacionamentos Ocorrem dois tipos importantes de relacionamentos durante a análise
 - Associação e
 - Generalização

Associações

Uma associação é uma conexão semântica bi-direcional entre classes

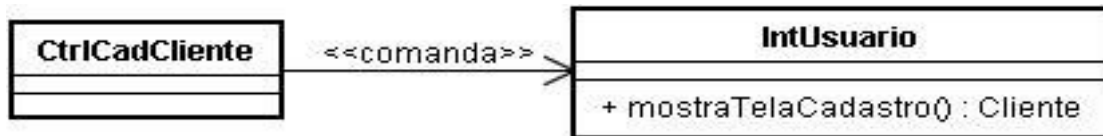
Isto implica na existência de uma ligação (link) entre os objetos das classes associadas. Associações são representadas no diagrama de classes por uma linha ligando as classes associadas. Em um link os dados podem fluir em ambas as direções



Navegação

Uma associação é um relacionamento bi-direcional

- Dada uma instância de GerenteDeMatrícula há um objeto Curso associado
- Dada uma instância de Curso há um objeto OficialDeMatrícula associado



Dando Nomes às Associações

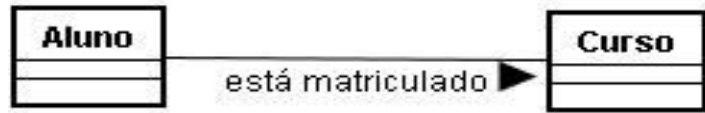
Para tornar seu significado mais claro, uma associação pode receber um nome

O nome é representado como uma etiqueta (label) colocada ao longo da linha de associação, no meio da linha, entre os ícones das classes

Um nome de associação geralmente é um verbo ou uma frase verbal

Dando Nomes às Associações

Para tornar seu significado mais claro, uma associação pode receber um nome. O nome é representado como uma etiqueta (label) colocada ao longo da linha de associação, no meio da linha, entre os ícones das classes. Um nome de associação geralmente é um verbo ou uma frase verbal.



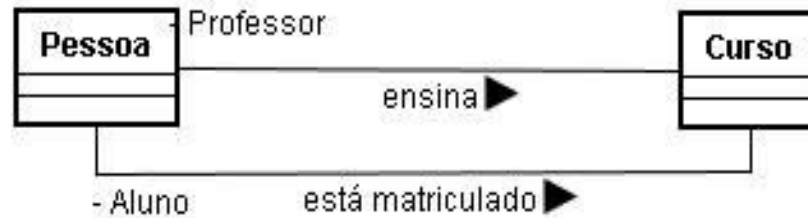
Definindo Papéis

Um papel denota o propósito ou capacidade em que uma classe se associa com outra.

- Nomes de papéis são tipicamente substantivos ou frases substantivas
- Um nome de papel é colocado ao longo da linha de associação, próximo da classe referenciada
- Um ou ambos os lados da associação podem ter nomes de papéis

Associações Múltiplas

- Pode existir mais de uma associação entre duas classes
- Se houver mais de uma associação entre duas classes, elas DEVEM ser nomeadas



3. 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

Por exemplo, na conexão entre Pessoa, atuando no papel de professor, e Curso

- Para cada instância de Pessoa, muitos (i.e., zero ou mais) Cursos podem ser ministrados
- Para cada instância de Curso, exatamente uma Pessoa é o professor

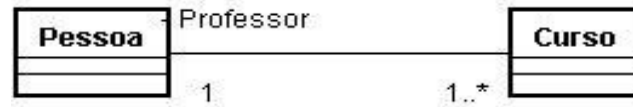
Muitos	<u>*</u>
Exatamente Um	<u>1</u>
Zero or mais	<u>0..*</u>
Um ou Mais	<u>1..*</u>
Zero or um	<u>0..1</u>
Intervalo Específico	2..4

Indicadores de Multiplicidade

Cada lado de uma associação contém um indicador de multiplicidade

Indica o número de objetos que participam no relacionamento

Exemplo: Multiplicidade

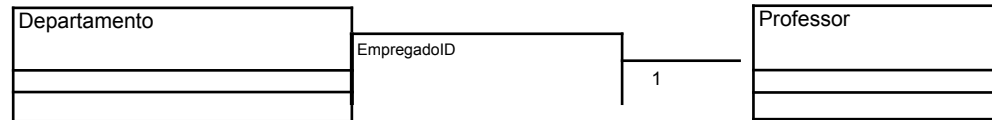


Decisões de Multiplicidade expõem muitas suposições, antes ocultas, sobre o problema sendo modelado

- Um professor pode estar indisponível?
- Um curso pode ter dois professores?

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
 - Uma restrição é mostrada entre chaves.

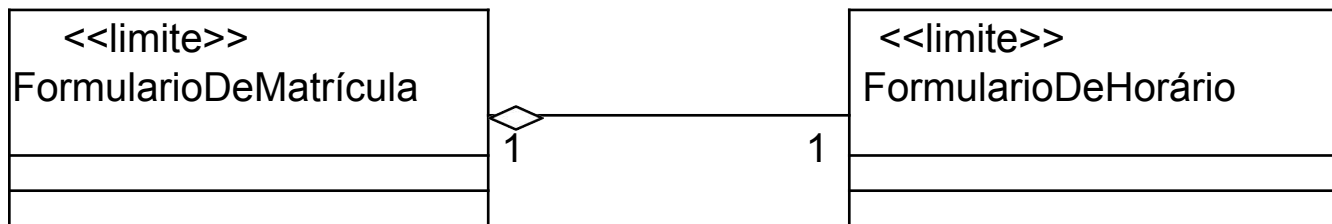
9.4. ATIVIDADE

1. Faça um Diagrama de Classes do subsistema de Operação de Trem, sem utilizar herança e considerando o seguinte:
 - Um trem pode ser de carga, passageiros ou ambos e pode ser movido por um ou mais vagões com força motriz.
 - Um vagão com força motriz possui um determinado tipo de motor, com potência específica.
 - O trem de passageiros pode possuir várias portas por vagão, que devem abrir e fechar automaticamente, ao chegar na estação e antes de partir, respectivamente.
 - A capacidade de passageiros de um trem deve estar registrada em quantidade, e a capacidade de cada vagão de carga em metros cúbicos
 - Cada trem possui uma rota que deve ser seguida durante a viagem.
 - A data de início de trabalho de cada vagão deve ser registrada.
 - A velocidade do trem deve ser controlada automaticamente. Ao arrancar ele deverá estar ganhando velocidade, e ao longo da viagem deverá manter outra velocidade e estando a uma determinada distância da estação deverá reduzir a velocidade para parar.
 - Os segmentos de trilho iniciam e terminam em um determinado quilômetro.
 - A localização do trem numa ferrovia deve ser possível.

AULA 10 AGREGAÇÕES E ASSOCIAÇÕES

1. AGREGAÇÃO

- Agregação é uma forma especializada de associação na qual o todo está relacionado a sua(s) parte(s).
- Agregação é conhecida como “parte-de” ou relacionamento de conteúdo.
- Uma agregação é representada como uma associação com um losango perto da classe que denota a agregação (todo).
- A multiplicidade é representada da mesma maneira que nas outras associações



Testes de Agregação

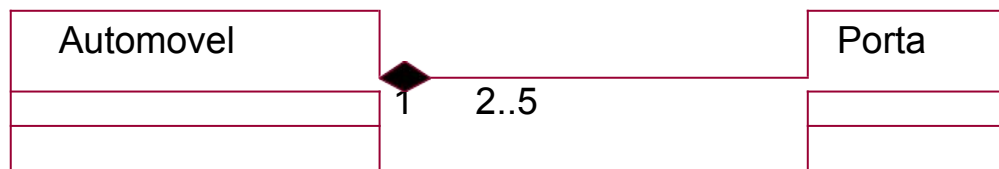
- A frase “parte de” é usada para descrever o relacionamento?
 - Uma Porta é “parte de” um Carro
- Algumas operações no todo são automaticamente aplicadas nas partes?
 - Mova o Carro, Mova a Porta
- Alguns valores de atributos são propagados do todo para algumas ou todas as suas partes?
 - O Carro é azul, a Porta é azul
- Há uma assimetria intrínseca no relacionamento onde uma classe é subordinada a outra?
 - Uma Porta É parte de um Carro, um Carro NÃO É parte de uma Porta

Tipos de Agregação

- Existem dois tipos de agregação:
 - Agregação propriamente dita, ou também chamada “agregação por referência”
 - Conhecida como relacionamento “tem-um(a)”
 - Envolve partes componentes que existem independentemente de seus agregados
 - Exemplo: A área “X” da empresa tem os empregados “A”, “B” e “C”
 - Visualizando o conceito de agregação, o agregado contém uma referência aos seus componentes



- Composição, ou também chamada “agregação por valor”
 - Conhecida como relacionamento “contém um(a)”
- Especifica que as partes componentes só tem um dono
- Especifica que o composto possui suas partes componentes
- Especifica que as partes componentes existem, ou “vivem e morrem” com seu composto proprietário
- Exemplo: Um automóvel possui de 2 a 5 portas



- Visualizando o conceito de composição, o composto contém os seus componentes
- Se dois objetos são estreitamente ligados por um relacionamento parte-todo
- O relacionamento é uma agregação
- Se dois objetos são usualmente considerados independentes, mesmo que eles estejam frequentemente ligados
- O relacionamento é uma associação

A diferença é apenas conceitual.

Agregação

É quando um objeto possui outros objetos, ele não depende desses objetos para existir.

Exemplo:

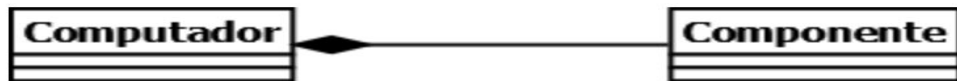


Uma Gaveta pode conter Meias, mas a Gaveta não é feita de Meias. Ou seja, mesmo sem Meias a Gaveta ainda existirá.

Composição

É quando um objeto é formado por outros objetos. Ou seja, suas partes o compõem, sem elas o objeto não existe.

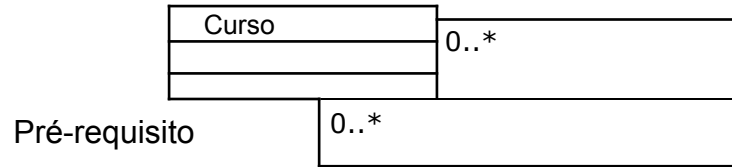
Exemplo:



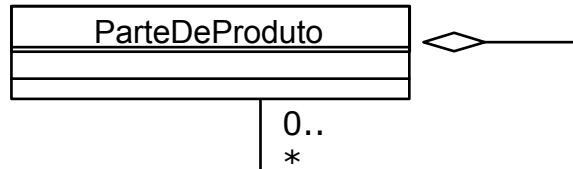
Um Computador é formado por seus componentes, como por exemplo placa-mãe, gabinete, hd, memória, placa de vídeo, etc. Sem todas essas peças não existe nosso Computador de acordo com a representação do diagrama. Logo, no nosso diagrama o Computador é um conceito, pois concretamente ele é composto por um conjunto diferentes componentes.

3. ASSOCIAÇÕES REFLEXIVAS

- Em uma associação reflexiva, objetos de uma mesma classe são relacionados
 - Indica que múltiplos objetos da mesma classe colaboram juntos de algum modo



- Um curso pode ter muitos pré-requisitos
- Um curso pode ser um pré-requisito para muitos outros cursos
- Agregados também podem ser reflexivos
 - Problema clássico de lista de materiais
- Isto indica um relacionamento recursivo



Um objeto ParteDeProduto é “composto de” zero ou mais objetos ParteDeProduto

ATIVIDADE

1) Construa um Diagrama de Classes inicial para controlar empréstimos e reservas de publicações em uma biblioteca, considerando o seguinte:

- A biblioteca pertence a uma empresa e somente os funcionários da empresa estão aptos a reservar ou emprestar publicações.
- A biblioteca tem acesso aos dados cadastrais dos funcionários.
- Uma publicação pode ser um livro, uma revista, um jornal, etc. A biblioteca pode possuir vários exemplares de uma mesma publicação.
- Qualquer tipo de publicação pode ser emprestada ou reservada.
- Não há número limite de reservas por publicação.
- Quando uma publicação for devolvida e para ela existirem reservas, o primeiro da fila deve ser avisado.
- A reserva expira um dia após o aviso da disponibilidade da publicação ao interessado.
Caso o mesmo não venha a emprestar a publicação, ela será considerada disponível e o próximo da fila deve ser avisado.
- Uma reserva pode ser excluída a pedido do usuário.
- Empréstimos de publicações não devolvidas 2 dias após o prazo serão consideradas irregulares e um funcionário da biblioteca deve resgatar o livro.

1) Crie o código (qualquer linguagem) para três das classes identificadas acima, que possuas relacionamento.

1) Crie o script SQL de criação dessas 3 classes (**opcional**).

BIBLIOGRAFIA BÁSICA

BECK, K; CUNNINGHAM, W.. A laboratory for teaching Object-Oriented thinking. *Anais do International Conference on Object-Oriented Programming, Systems, Languages and Applications OOPLSA89*. New Orleans, EUA: 1989.

BEZZERRA, E.. *Princípios de análise e projeto de sistemas com UML*. Rio de Janeiro, Brasil: Campus, 2002.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I.. *The Unified Modeling Language user guide*. 2a. ed. Westford, Massachusetts, EUA: Addison-Wesley, 2005.

JACOBSON, I.; BOOCH, G.; RUMBAUGH, J.. *The Unified Software Development Process*. Reading, Massachusetts, EUA: Addison-Wesley, 1999.

LARMAN, C.. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. 3a. ed. New Jersey, EUA: Addison-Wesley Professional, 2004.

OMG. *Unified Modeling Language - Superstructure specification Versão 2.0*. OBJECT MANAGEMENT GROUP: Northham, Massachusetts, EUA, 2005

PRESSMAN, R. S.. *Software Engineering: a practitioner's approach*. 6a. ed. New York, EUA: McGraw-Hill, 2005.

RUMBAUGH, J.; JACOBSON, I.; BOOCH, G.. *The unified modeling language reference manual*. 2a. ed. Reading, Massachusetts, EUA: Addison-Wesley, 2004.