Análise e Projeto de sistemas

Diagrama de Classes

Jorge Roberto Nogueira

INTRODUÇÃO

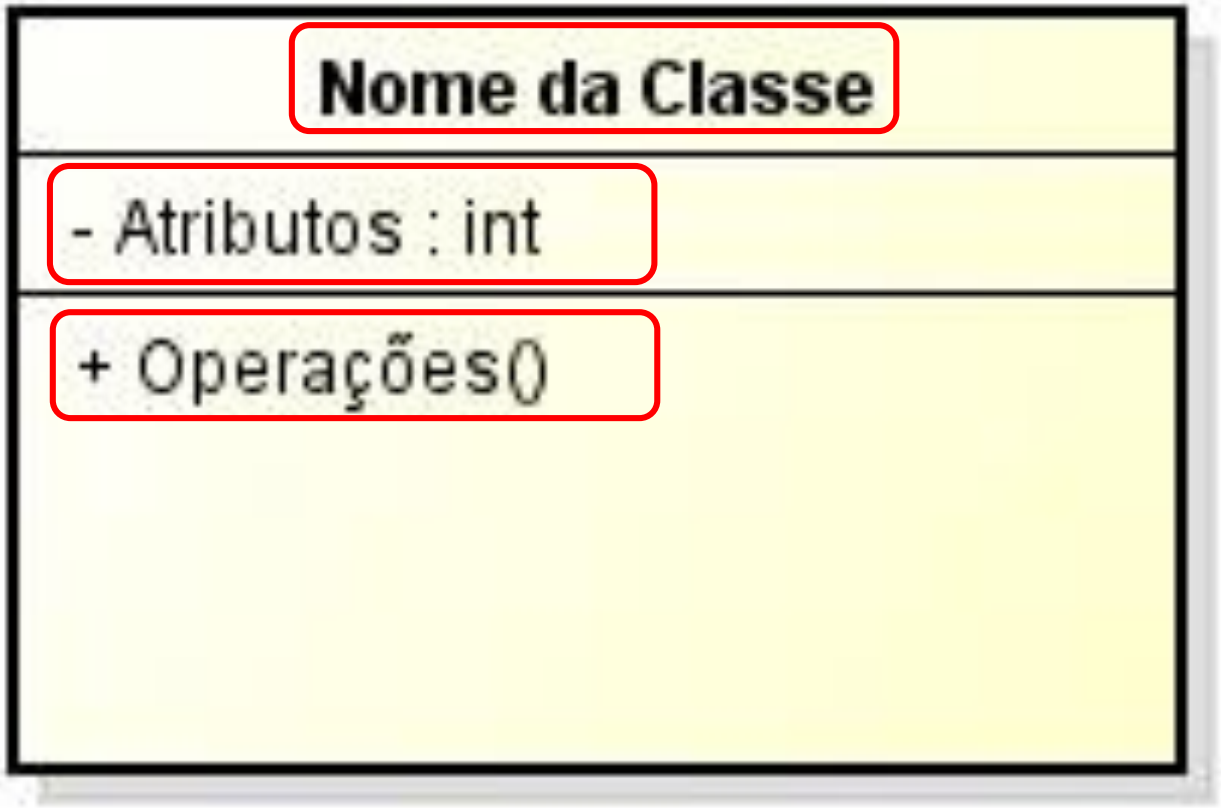
- Diagrama de Classes: o mais utilizado e o mais importante da UML.
- Utilizado junto com o Modelo de Casos de Uso.
  - mas podem ter complementos
  - nem sempre funcionam sozinhos

FINALIDADE:

- permitir a visualização das classes do sistema com seus atributos e métodos bem como seus relacionamentos.
- Visão estática da organização do projeto.
- Serve como base para muitos outros diagramas da UML e para o seu projeto junto com o Diagrama de Caso de Uso.

- Enfatiza os dados que serão necessários para a construção do sistema de informação.
- Este diagrama possui o maior numero de símbolos para a sua representação
- Ele não é novo, é uma evolução do modelo de entidade e relacionamento (E-R).
- É um dos mais importantes de todos os diagramas da UML.
- Seu objetivo é permitir a visualização dos componentes da base de dados do sistema.

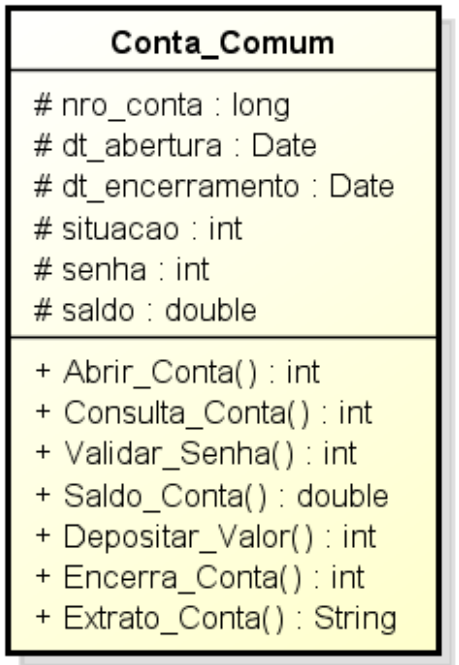
CLASSE



VISIBILIDADE

- Identifica por quem uma propriedade (atributo ou operação) pode ser utilizada.

Classe Conta Comum com seus atributos e Operações (Métodos)



Definimos a visibilidade de uma propriedade (atributo ou operação) por palavras-chaves ou ícones.

+ ou public (publico) - A propriedade será vista e usada dentro da classe na qual foi declarada, em qualquer elemento externo e nas classes descendentes.

ou protected (protegido) – A propriedade será vista e usada apenas dentro da classe na qual foi declarada e pelas classes descendentes.

- Ou private (privado) – A propriedade será vista e usada apenas dentro da classe na qual foi declarada.

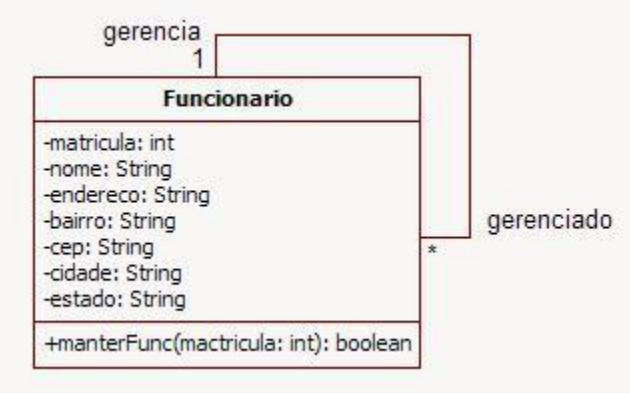
~ ou package (pacote) – A propriedade poderá ser vista e usada por elementos que estejam declarados dentro do mesmo pacote no qual está inserida a classe que a declarou. É como a visibilidade pública. Só que não generaliza a qualquer elemento externo, mas apenas aos elementos externos localizados no mesmo pacote.

RELACIONAMENTOS OU ASSOCIAÇÕES

- As classes compartilham informações entre si e colaboram para a execução do processos executados pelos sistemas.
- Uma associação descreve um vinculo que ocorre entre os objetos de uma ou mais classes.
- As associações são representadas por linhas ligando as classes envolvidas.
- Essas linhas podem ter nomes ou títulos para auxiliar a compreensão do tipo de vinculo entre os objetos.

ASSOCIAÇÃO UNÁRIA OU REFLEXIVA

- Este tipo de associação ocorre quando existe um relacionamento de um objeto de uma classe com objetos da mesma classe.



MULTIPLICIDADE

- Indica uma faixa de cardinalidade permitida a um elemento, isto é, a quantidade de instâncias possíveis em um relacionamento.

  Exemplo:

  - Numa classe Pessoa com o atributo cônjuge podemos afirmar que sua multiplicidade é de no mínimo zero e no máximo um cônjuge.

  - Numa classe Disciplina que se relacione com uma classe Aluno podemos afirmar que para cada instância de Disciplina há um relacionamento com no mínimo zero e no máximo vários alunos; e para cada instância de Aluno há um relacionamento com no mínimo zero (aluno está com matricula trancada) e no máximo várias disciplinas.

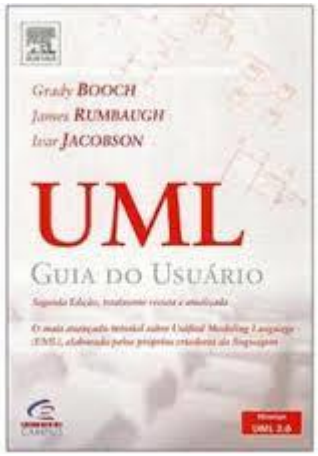
TABELA DE MULTIPLICIDADE

Multiplicidade	Significado
0..1	No mínimo zero (nenhum) e no máximo um. Indica que os objetos das classes associadas não precisam obrigatoriamente estar relacionados, mas se houver relacionamentos indica que apenas uma instancia da classe relaciona-se com as instâncias da outra classe (ou da outra extremidade da associação, se esta for unária).
1..1	Um e somente um. Indica que apenas um objeto da classe relaciona-se com os objetos da outra classe.
0..*	No mínimo nenhum e no máximo muitos. Indica que pode ou não haver instancia da classe participando do relacionamento.
*	Muitos. Indica que muitos objetos da classe estão envolvidos na associação.
1..*	No mínimo um e no máximo muitos. Indica que há pelo menos um objeto envolvido no relacionamento, podendo haver muitos objetos envolvidos.
3..5	No mínimo 3 e no máximo cinco. Estabelece que existem pelo menos três instancias envolvidas no relacionamento e que podem ser quatro ou cinco as instancias envolvidas, mas não mais do isso.

PERGUNTAS

Continua na próxima aula

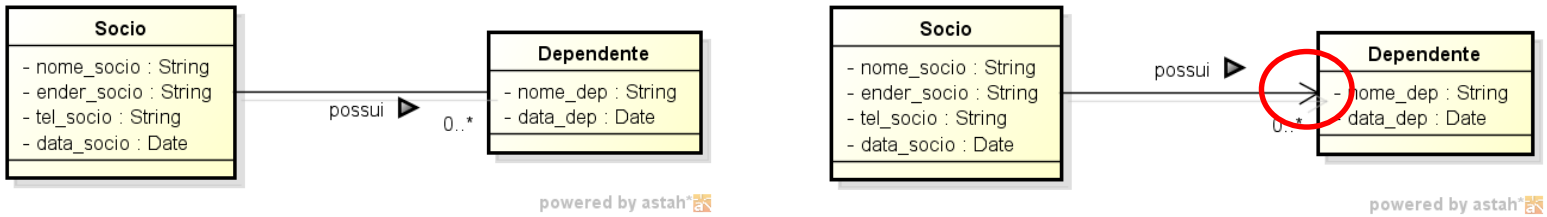
Aprenda UML por meio de Estudo de caso
Autor: Wilson Moraes Góes
Editora: Novatec



UML Guia do Usuário
Autor: Grady Booch, James Runbaugh e Ivar Jacobsin
Editora: Campus

ASSOCIAÇÃO BINÁRIA

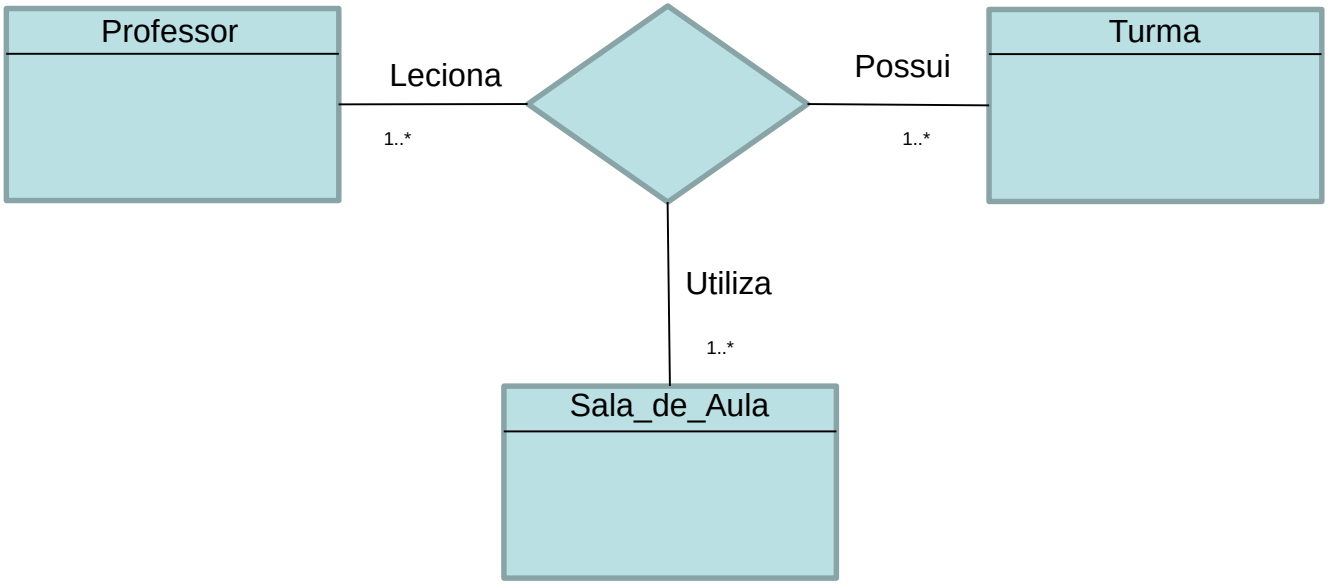
- Associação binária ocorrem quando são identificados relacionamentos entre objetos de duas classes distintas.
- Esse tipo de associação é a mais comumente encontrada.
- Podemos indicar a navegabilidade que é representada por uma seta em uma das extremidades da associação, identificando o sentido em que as informações são transmitidas.
- Navegabilidade é uma informação que só deve ser acrescentada na fase de projeto. Algumas ferramentas CASE é representada como um triangulo em forma de seta.



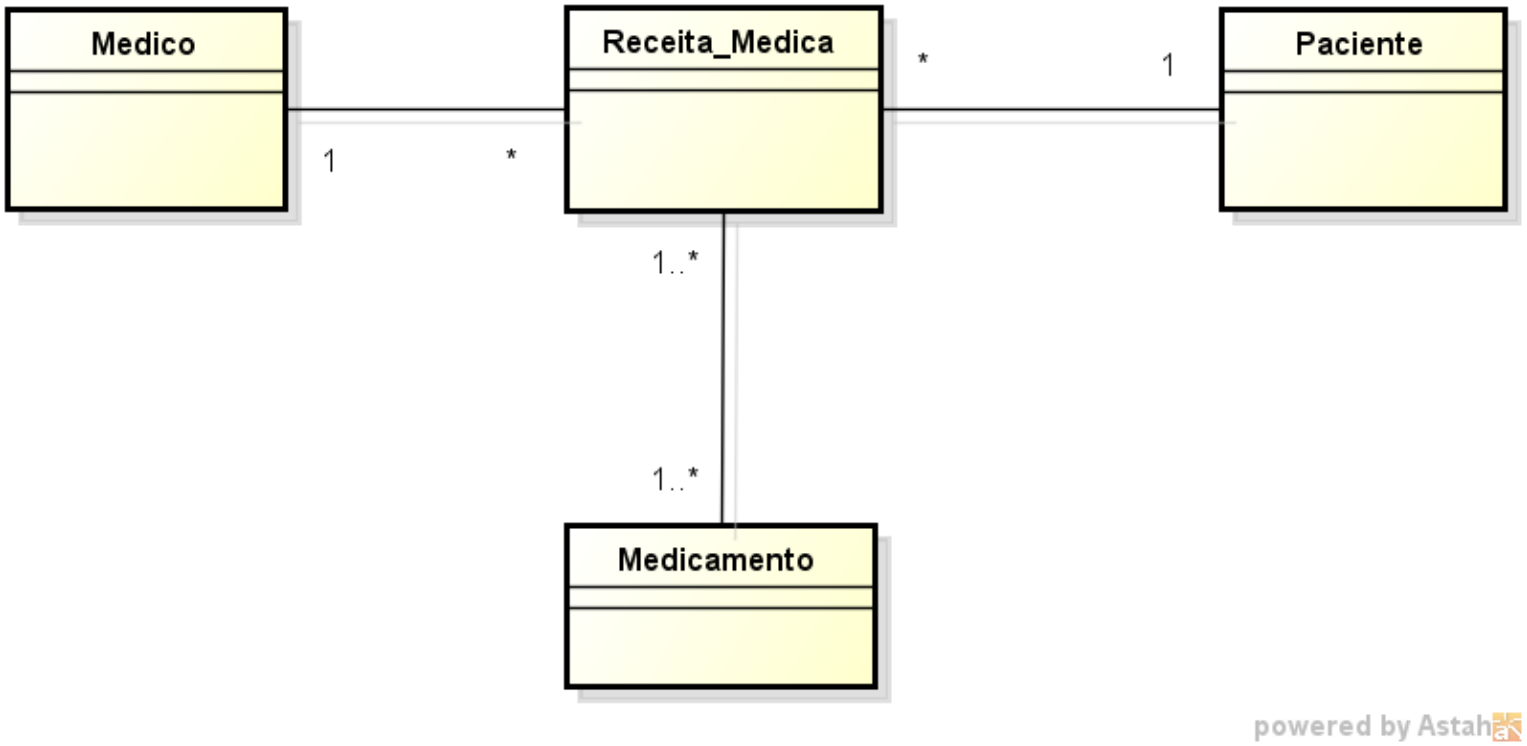
ASSOCIAÇÃO TERNÁRIA OU N-ÁRIA

- São associações que conectam objetos de mais de duas classes.
- São representadas por um losango para onde convergem todas as ligações da associação.

Devem ser evitadas pois sua leitura é, por vezes, difícil de ser interpretadas



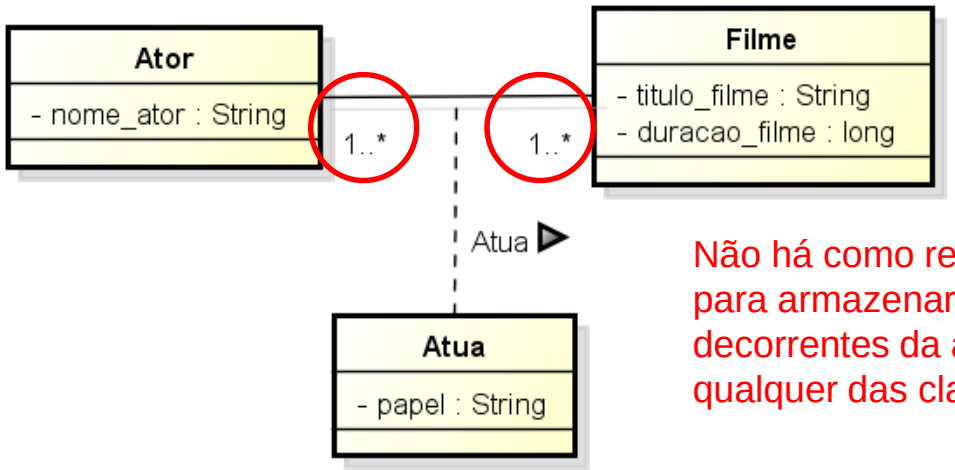
ASSOCIAÇÃO TERNÁRIA OU N-ÁRIA



Classe Associativa

- São produzidas quando existe ocorrências de associações que tenham multiplicidade muito(*) em todas as suas extremidades.

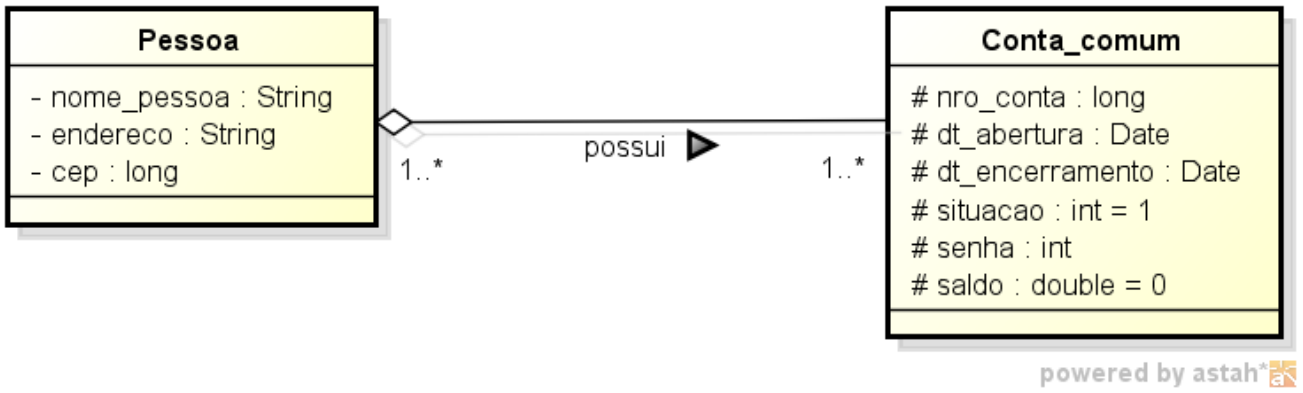
- Classes associativas são necessárias nos casos em que existem atributos relacionados à associação que não podem ser armazenados por nenhuma das classes envolvidas.



AGREGAÇÃO

- Este tipo de associação demonstra que as informações de um objeto (chamado objeto-todo) precisam ser completadas pelas informações contidas em um ou mais objetos de uma outra classe (chamados objetos-parte).

- O símbolo de agregação difere do de associação por conter um losango na extremidade da classe que contém os objetos-todo.

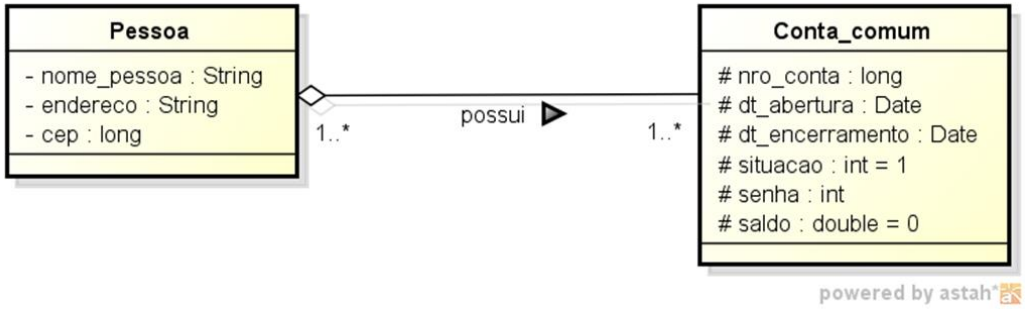


AGREGAÇÃO

O exemplo demonstra uma associação de agregação existente entre uma classe Pessoa e uma classe Conta_comum.

- Sempre que uma pessoa for consultada, além de suas informações pessoais serão apresentadas todas as contas que ela possui.

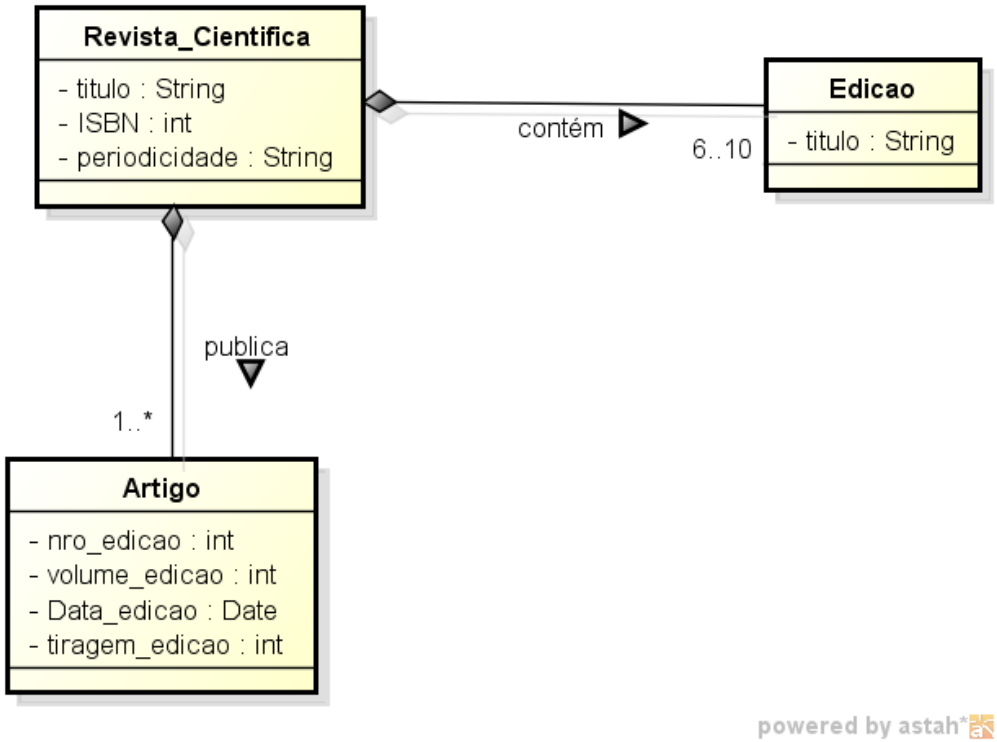
Observação: Uma pessoa pode possuir muitas contas, uma conta pode ser possuída por muitas pessoas. Isso é característico das agregações nas quais os objetos-partes podem ser compartilhados por mais de um objeto-todo.



COMPOSIÇÃO

- Este tipo de associação constitui em uma variação da agregação onde é apresentado um vinculo mais forte entre os objetos-todo e os objetos-parte.
- Demonstra os objetos-parte têm de estar associados a um único objeto-todo.
- Em uma composição os objetos-parte não podem ser destruídos por um objeto diferente do objeto-todo ao qual estão relacionados.
- O símbolo de composição diferencia-se do símbolo de agregação por utilizar um losango preenchido.

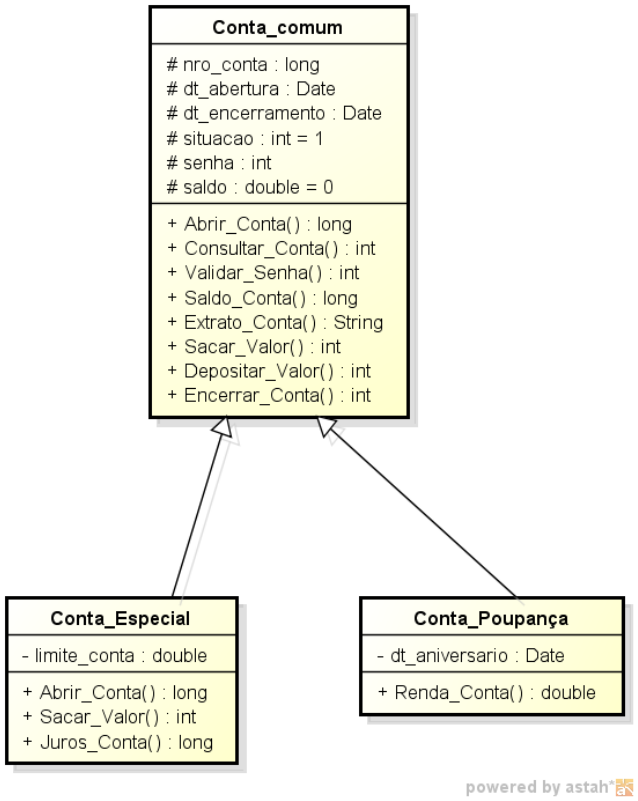
COMPOSIÇÃO


Revista_Cientifica
- titulo : String
- ISBN : int
- periodicidade : String
contém
6..10
Edicao
- titulo : String
publica
1..*
Artigo
- nro_edicao : int
- volume_edicao : int
- Data_edicao : Date
- tiragem_edicao : int
powered by astah*

GENERALIZAÇÃO/ESPECIALIZAÇÃO

- Esta associação é de mesmo nome utilizada no diagrama de caso de uso.
- Representa a ocorrência de herança entre as classes, identificando as classes mãe (ou superclasses), chamada gerais, e classes-filhas (ou subclasses) chamadas de especializadas.
- Demonstra a hierarquia entre as classes e possivelmente métodos polimórficos nas classes especializadas.

GENERALIZAÇÃO/ESPECIALIZAÇÃO



Composição

objetos-parte têm de estar associados a um único objeto-todo.

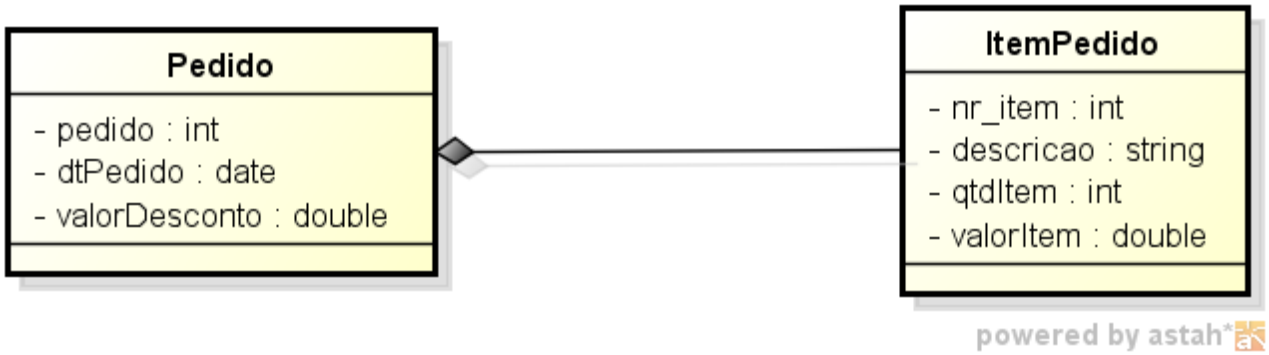
EXEMPLO DE AGREGAÇÃO

O time é formado por atletas, ou seja, os atletas são parte integrante de um time, mas atletas existem independentemente de existir um time.

- Na agregação, a existência do objeto-parte faz sentido, mesmo não existindo o objeto-todo.

EXEMPLO DE AGREGAÇÃO

Um pedido é composto por um ou vários itens, mais um produto não é item de um pedido se não existir pedido.

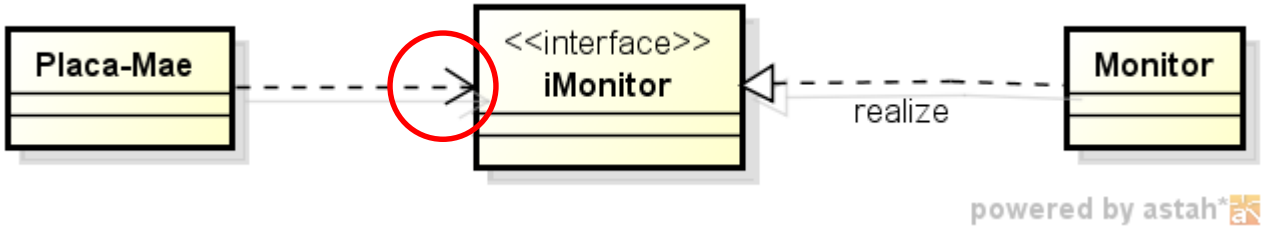


- É uma agregação mais forte, a existência do objeto-parte não faz sentido se o objeto-todo não existir.

DEPENDÊNCIA

- Indica certo grau de dependência de uma classe em relação à outra. O relacionamento de dependência é representado por uma linha tracejada entre duas classes, contendo uma seta apontando para a classe da qual a classe posicionada na outra extremidade do relacionamento é dependente.

- O relacionamento de dependência é utilizado também em diversos outros diagramas, tendo algumas variações definidas por estereótipos, como include ou extend no diagrama de casos de uso, instantiate no diagrama de objetos ou merge no diagrama de pacotes.

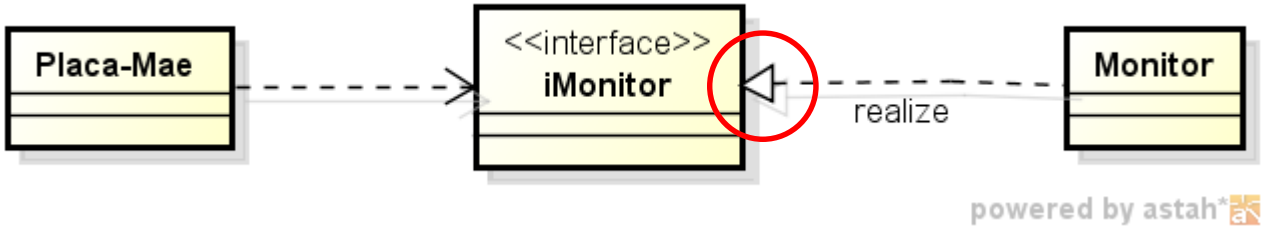


REALIZAÇÃO

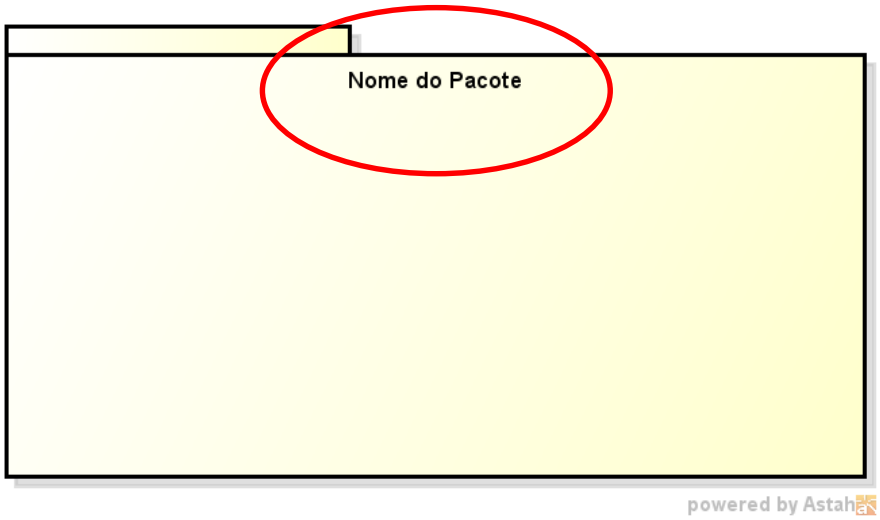
- Trata-se de um tipo de relacionamento especial que mistura características dos relacionamentos de generalização e dependência, sendo usada para identificar classes responsáveis por executar funções para outras classes.

- Esse tipo de relacionamento herda o comportamento de uma classe, mas não sua estrutura.

- O relacionamento de realização é representado por uma linha tracejada contendo uma seta vazia que aponta para a classe que tem uma ou mais funções que devem ser realizadas por outra. Enquanto na outra extremidade da linha é definida a classe que realiza esse comportamento.



- Pacote



- Elemento básico organizador de um modelo de software.
- Em seu conteúdo pode ser representados:
  - Casos de usos
  - Classes
  - Outros pacotes
  - Diagramas inteiros
- Não permite que um objeto com o mesmo nome coexista dentro dele, mas em pacotes distintos isso é possível.

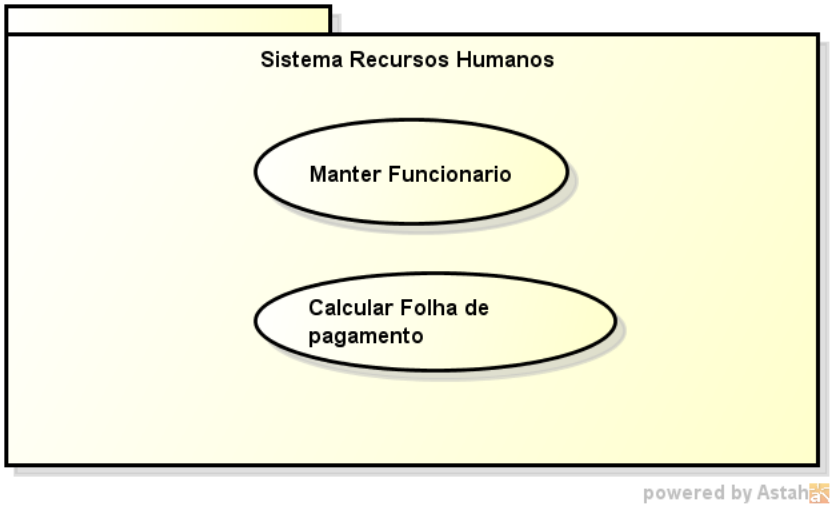
Para se referenciar a um objeto existente no pacote:

Sintaxe: <<nome do pacote>> seguido de :: e <<nome do objeto>>

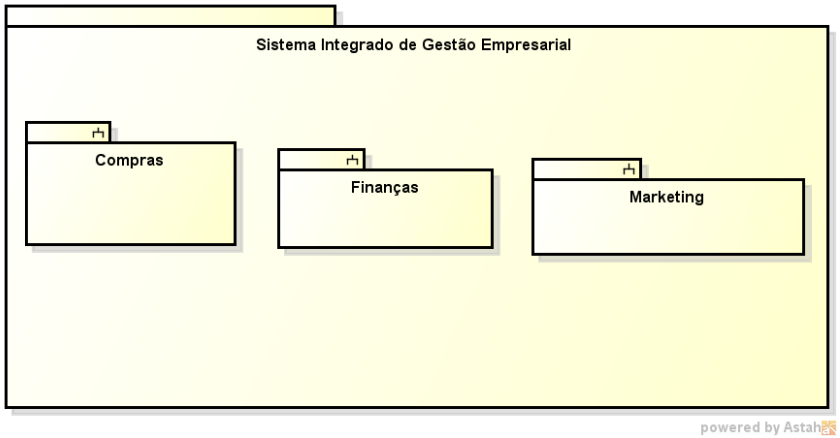
Exemplo: Sistema Recursos Humanos::Manter Funcionários

Diagrama de Classes

- Ilustração do pacote Sistema Recursos Humanos e seus dois objetos (caso de uso): Manter funcionário e Calcular folha de pagamento

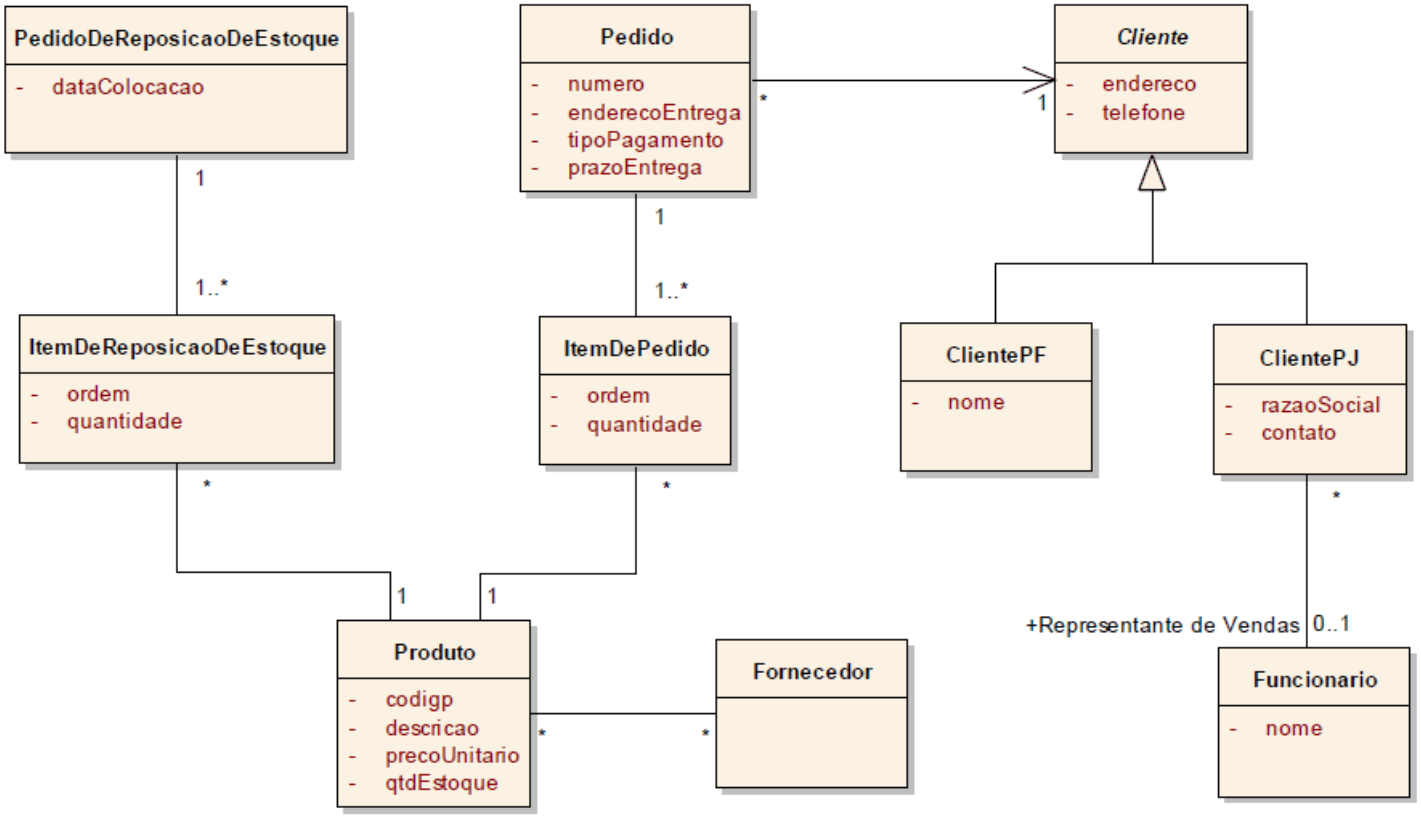


Um pacote também pode ser utilizado para representar subsistemas.

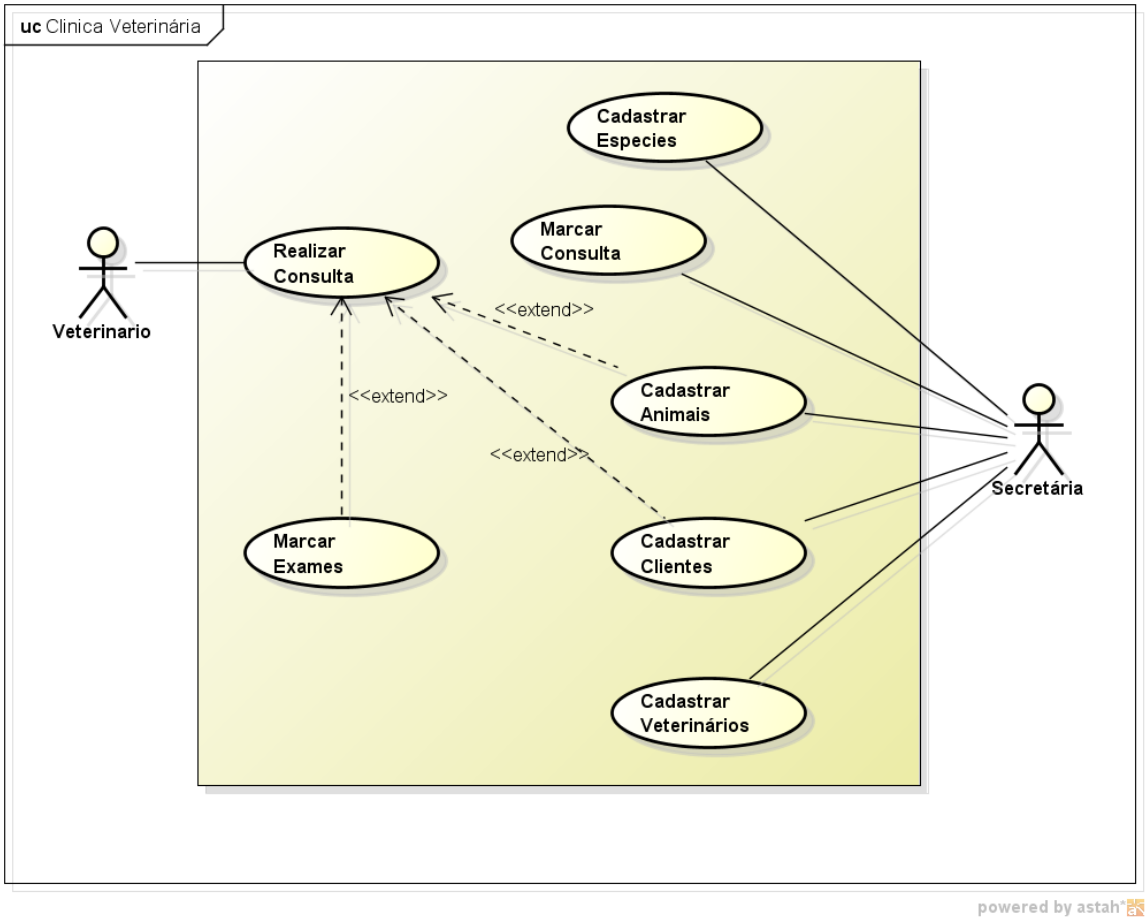


Exemplos de um sistema integrado de gestão empresarial (ERP)

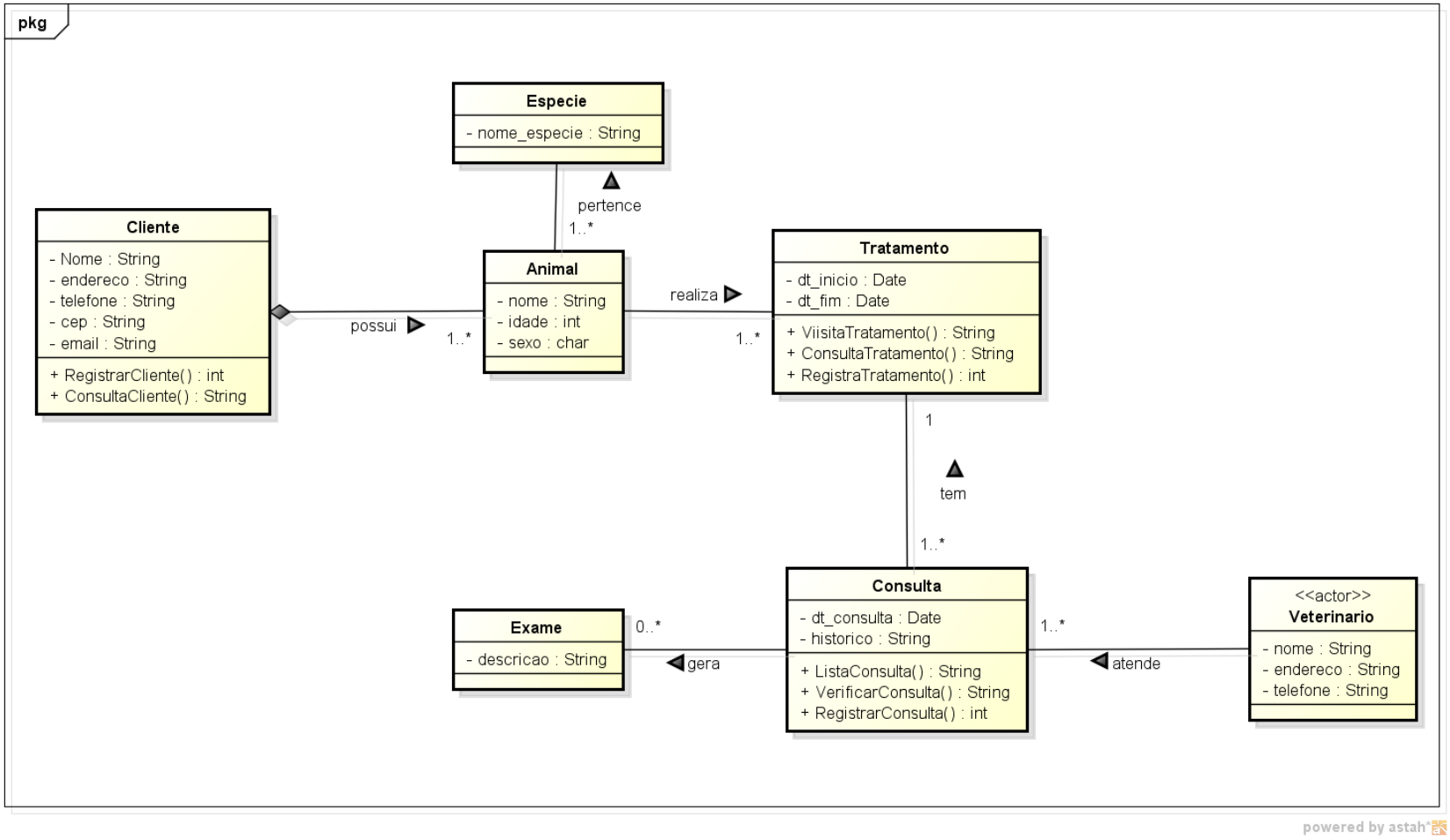
A classe é o elemento-chave em um diagrama de classes. No nível conceitual, uma classe representa um conceito do negócio. Assim, conforme ilustra a Figura abaixo, *Pedido*, *Cliente* etc. são conceitos que fazem parte de um contexto típico em uma empresa fictícia – à qual lida com pedidos de sua clientela.



Clinica Veterinária



Clinica Veterinária



ESTEREÓTIPOS DO DIAGRAMA DE CLASSES

São características poderosas que são utilizadas em todos os diagramas da UML e no diagrama de classes costumam exercer um papel muito importante.

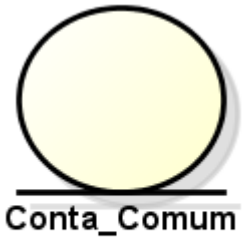
É uma maneira de destacar determinados componentes do diagrama, tornando explicito que tais componentes executam alguma função um pouco diferente dos demais componentes apresentados no diagrama.

Há três estereótipos predefinidos na linguagem UML, bastante utilizados nos diagramas de classes que merecem destaques: <<entity>>, <<boundary>> (fronteira) e <<control>>.

ESTEREÓTIPO <<entity>>

- Tem como objetivo tornar explicito que uma classe é uma entidade e que contem informações recebidas e armazenada pelo sistema.

- Fornecem informações de que normalmente terão muitos objetos e que os mesmos possivelmente terão um período de vida logo, isto é, existe a possibilidade de que os objetos dessas classes precisem ser persistidos, ou seja preservados fisicamente de alguma maneira.



PERGUNTAS

Continua na próxima aula