

Modelagem de classes e objetos

Análise e Projeto Arquitetural de
Software

Prof. Thiago

Objetivos

- Após esta aula, você deverá ser capaz de:
 - Compreender a estruturação de classes e objetos;
 - Modelar a estrutura de classes utilizando a notação do Diagrama de Classes.

Conceitos básicos de OO

- Vamos estudar agora os principais conceitos empregados na modelagem e programação orientadas a objetos
 - **Classes (e objetos)**
 - **Atributos**
 - **Métodos**
 - **Mensagens**

Classe

- **Definição:** Uma *classe de objetos* descreve um grupo de objetos com propriedades (atributos) em comuns, comportamento (operações) em comum, relacionamentos comuns com outros objetos.
- **Exemplo:** Pessoas.

Objeto

- **Definição:** Um objeto é uma *instância* de uma classe. Ou seja, um elemento que pertence àquela classe.
- **Exemplo:** Maria é um objeto da classe Pessoas.

Atributos

- **Definição:** São propriedades comuns aos objetos de uma classe, porém cada objeto pode ter um valor diferente para seus atributos.
- **Exemplo:** Nome é um atributo da classe Pessoas. Mas uma pessoa chamada Maria tem nome diferente de uma pessoa chamada João.

Método

- **Definição:** comportamento de uma classe de objetos. Um método pode acessar e alterar atributos.
- **Exemplo:** o método TrocarDeRoupa da classe Pessoa faz com que o atributo Roupa se altere.

Mensagem

- **Definição:** chamada a um objeto para invocar um de seus métodos.
- **Exemplo:** efetivamente “fazer” com que uma pessoa troque de roupa.
- **Lembrete:**
 - Método: *descrição* de uma operação
 - Mensagem: *execução* de uma operação

“Teatro de objetos”

- Um exemplo divertido utilizando esses conceitos é a modelagem de cobranças de pênalti em um jogo de futebol
- Classes:
 - Goleiro
 - Batedor de pênalti
 - Torcedor educado
 - Torcedor mal-educado
 - Juiz

“Teatro de objetos”

- Cada objeto de cada classe tem alguns atributos e entende algumas mensagens:

Classe	Atributos	Mensagens
Goleiro	Time NumeroDaCamisa	SuaVez; CobrançaAutorizada; VenceuOTime (X)
Batedor de pênalti	Time NumeroDaCamisa	SuaVez; CobrançaAutorizada; VenceuOTime (X)
TorcedorEducado	Time	Ação; VenceuOTime (X)
TorcedorMalEducado	Time	Ação; VenceuOTime (X)
Juiz		CoordenarCobrança

Cartão de Responsabilidade e Colaboração (CRC)

- Uma das primeiras técnicas para modelagem de sistemas OO
- Consiste em definir, para cada classe de objetos candidata:
 - Os **atributos** que descrevem objetos dessa classe;
 - As **responsabilidades** de um objeto dessa classe;
 - **Que classes colaboram** para a classe cumprir suas responsabilidades.

CRC – um exemplo

Classe	Atributos
PedidoDeVenda	-Data do pedido -Hora do pedido -Vendedor -Produtos -Cliente
Responsabilidades	Colaborações
-Calcular o valor do frete -Calcular o percentual de ICMS do destino -Fornecer informações para o cálculo da Fatura e da Nota Fiscal	-Fatura -NotaFiscal

O que a classe faz

Quais classes colaboram
para a classe atingir seu
objetivo

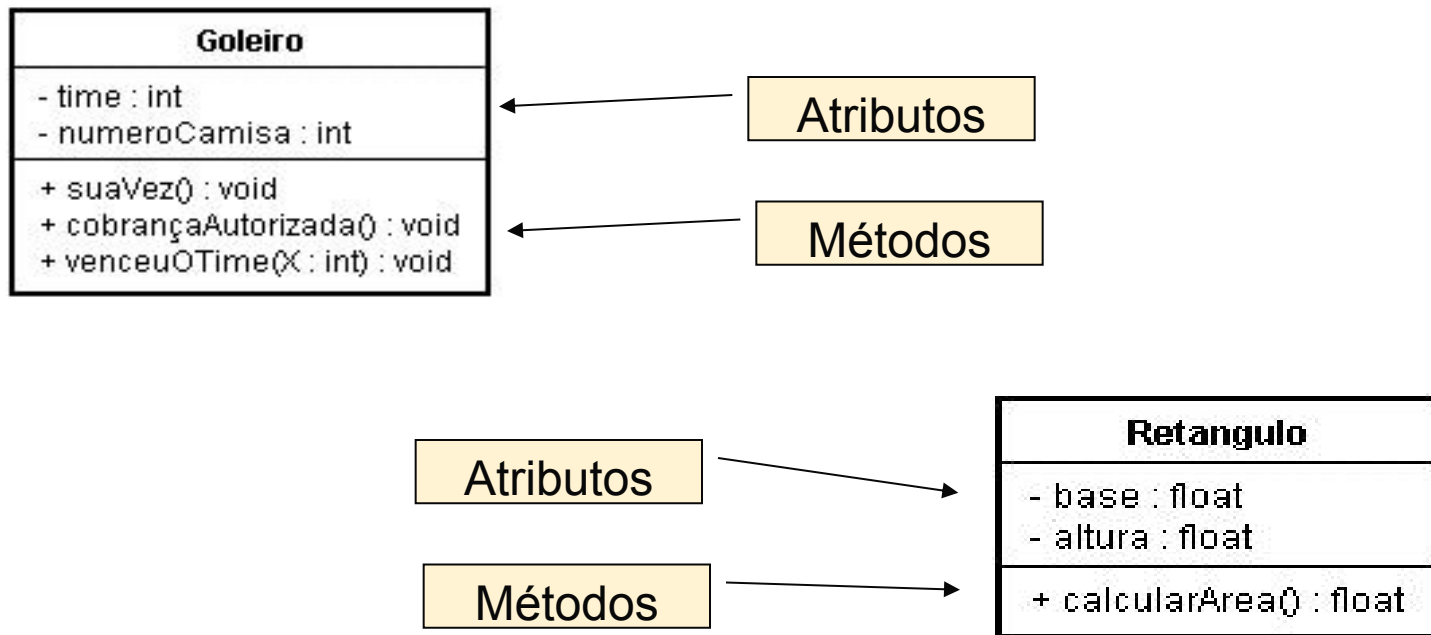
Diagrama de classes

- É o principal diagrama estrutural do sistema na UML
- Exibe as classes, seus métodos e atributos, e os possíveis relacionamentos entre as classes

Classe

- É representada por um retângulo com três compartimentos
- O nome da classe normalmente é uma ou mais palavras curtas, sem espaçamento e iniciadas por letra maiúscula
- Os atributos e métodos são palavras curtas iniciadas por letra minúscula
 - Essas regras não são indispensáveis na UML mas acabam facilitando o mapeamento posterior para uma linguagem de programação como Java

Classe - exemplos



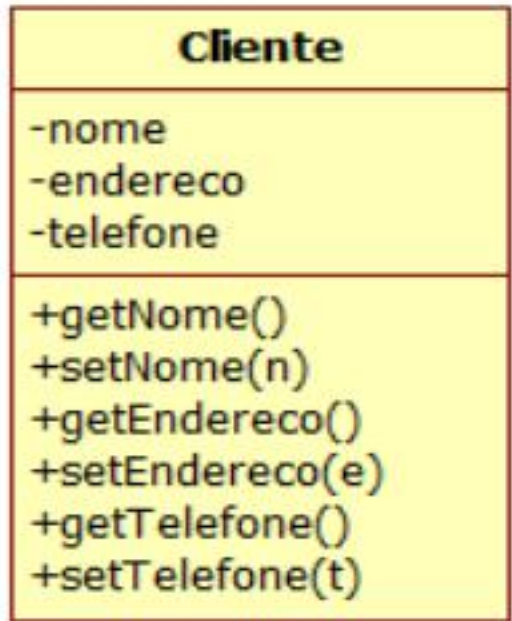
Classe - exemplos

- O padrão UML não obriga, mas permite que os atributos tenham definidos os seus tipos de dados
- Os métodos podem indicar os parâmetros esperados e seus tipos, bem como o tipo do retorno
- Esses tipos estarão normalmente associados à linguagem da implementação

Encapsulamento

- É a capacidade de um objeto ocultar do mundo exterior dados e detalhes de sua implementação que não dizem respeito ao mundo exterior
- Ajuda a separar a funcionalidade que um objeto fornece da *forma* como ela é implementada internamente
- É implementada através do nível de visibilidade de métodos e atributos (que podem ser públicos ou privados)
 - Públicos: visíveis e disponíveis para quem acessa o objeto
 - Privados: visíveis e disponíveis apenas no código interno do objeto

Encapsulamento



O sinal de menos indica que os atributos, por padrão, são **privados**

O sinal de mais indica que os métodos, por padrão, são **privados**

O acesso e modificação dos valores dos atributos é feito através dos **métodos de acesso** (ou getters e setters), que garantem o encapsulamento

Relacionamentos entre classes

- Qualquer sistema orientado a objetos envolverá a existência de diversos objetos, de diversas classes, que vão trabalhar em **colaboração** para atingir um objetivo
- O diagrama de classes permite a definição de como as classes se relacionarão para executar um trabalho coordenado

Relacionamentos entre classes

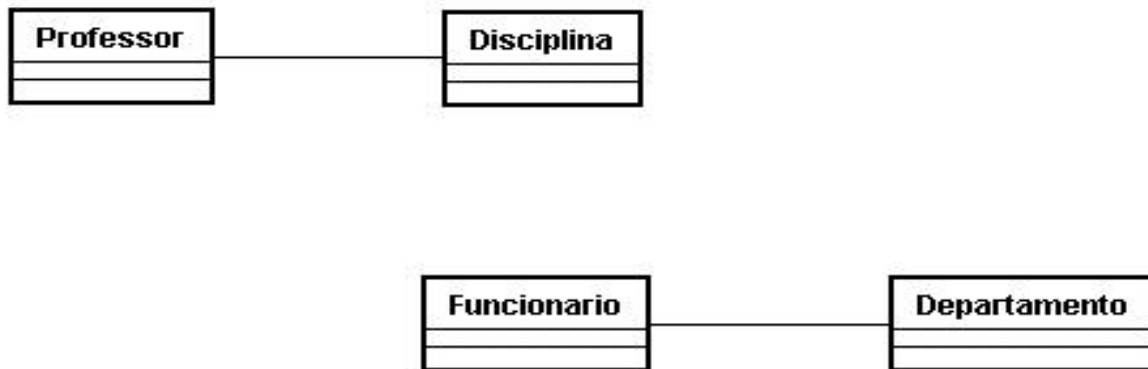
- Os possíveis relacionamentos entre classes são:
 - Dependência
 - Associação
 - Agregação e composição
 - Especialização

Associação

- A associação modela qualquer relação lógica entre duas classes
 - Por exemplo, Classe “tem” Aluno, Conta bancária “tem” Correntistas, Funcionário “está subordinado a” Departamento
- Muito semelhante ao conceito de relacionamento no Diagrama Entidade-Relacionamento

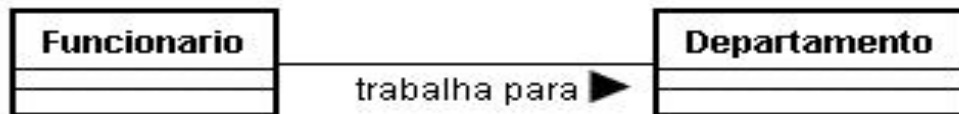
Associação

- A associação é indicada por uma linha contínua conectando as classes



Associação

- Uma associação pode ter um nome, definido junto ao centro da linha
- A direção de leitura do nome dado pode ser indicada por meio de uma seta



Associação

- A *cardinalidade* (quantidade de objetos de cada classe que podem ou devem participar do relacionamento) pode ser também indicada



Uma disciplina é ministrada por exatamente um professor.
Um professor ministra no mínimo uma disciplina e pode ministrar várias.

Uma instância de uma classe está associada à quantidade de instâncias da outra classe indicada no lado oposto.



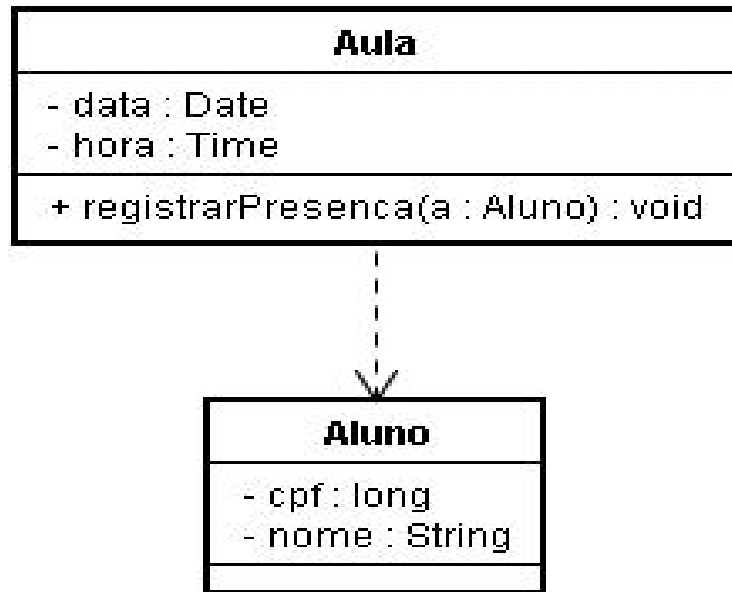
Associação

■ Exemplos de cardinalidade:

Cardinalidade	Significado
0..1	No mínimo zero e no máximo um
1..1	Um e somente um
0..*	No mínimo zero e no máximo vários
*	
1..*	No mínimo um e no máximo muitos
3..5	No mínimo três e no máximo cinco

Dependência

- Quando uma classe A depende dos serviços de outra classe B para fornecer alguma funcionalidade, dizemos que há uma relação de *dependência* entre A e B
- Essa relação é indicada por uma seta tracejada
- Situação frequente: A usa um objeto do tipo B como parâmetro de um método

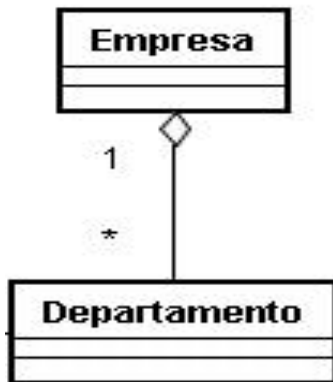


Agregação

- A agregação é um relacionamento entre classes do tipo “parte-todo”
- Uma objeto de uma classe “maior” tem vários objetos da classe “menor”

Agregação

- A agregação é indicada por um losango branco no lado da classe cujos objetos vão possuir várias instâncias dos objetos da outra
- A notação de cardinalidade continua válida

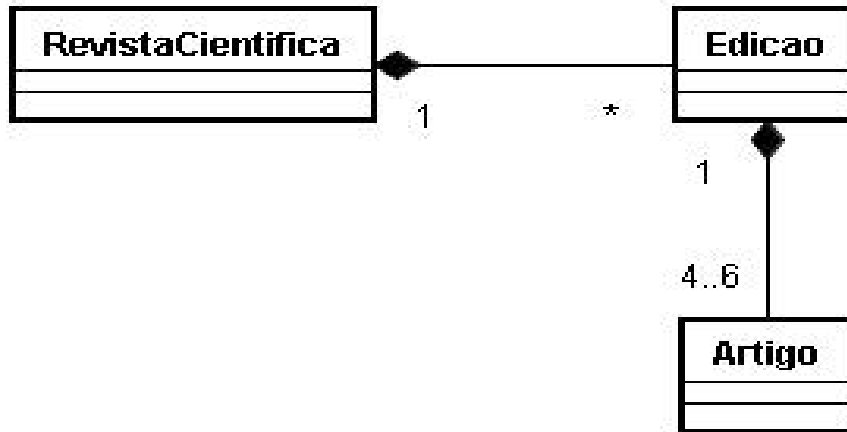


Composição

- A composição é um tipo especial de associação
- Nela os objetos da classe “menor” devem pertencer exclusivamente a um único objeto da classe “maior”
- A classe “maior” é responsável pela criação e destruição dos objetos da classe “menor”

Composição

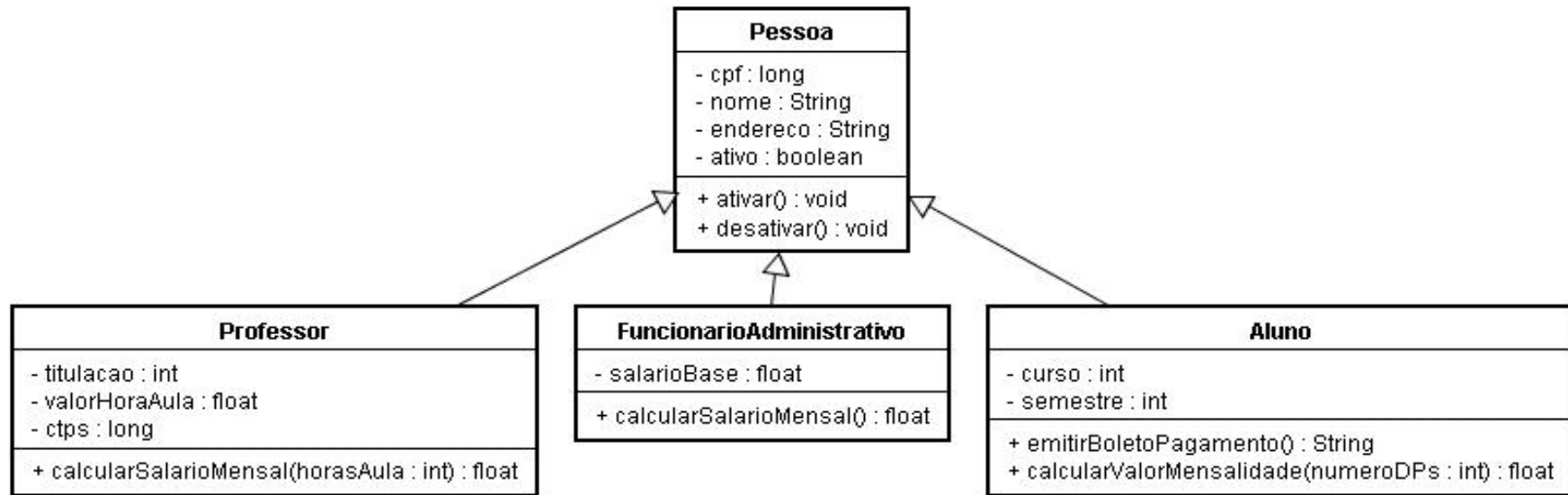
- A composição é indicada por um losango preenchido



Especialização

- A especialização é uma relação entre uma classe mais genérica e uma classe mais específica
- Representa um dos conceitos fundamentais da orientação a objetos: a herança entre classes
- A classe mais geral é chamada de *superclasse*; a mais específica, de *subclasse*
- A subclasse incorpora todos os atributos e métodos da sua superclasse, e mais os atributos e métodos que ela venha a definir

Especialização - exemplo

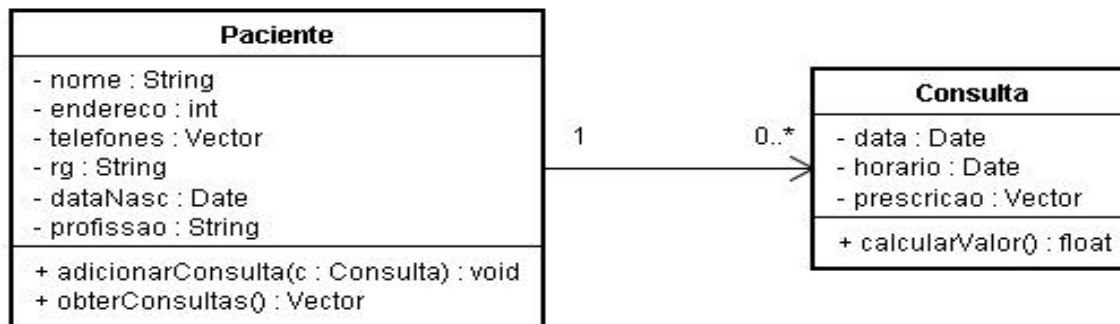


Especialização - exemplo

- No exemplo anterior, utilizamos as seguintes premissas:
 - Toda pessoa no sistema tem nome, CPF e endereço. Logo esses atributos serão herdados pelas três subclasses de pessoa
 - Toda pessoa pode ter seu cadastro ativo ou inativo. Assim, os métodos *ativar()* e *desativar()* também serão herdados pelas três subclasses de pessoa, bem como o atributo *ativo*
 - O professor tem seu salário dependente do valor da hora-aula e da quantidade de horas-aula ministradas
 - O funcionário tem um salário base fixo. O cálculo mensal do salário envolve descontos de INSS, Imposto de Renda, etc...
 - O aluno pertence a um curso e está em um semestre. O valor da sua mensalidade depende da quantidade de DP's cursadas

Navegabilidade de associações

- Uma associação pode ter uma seta em uma de suas extremidades $A \rightarrow B$
- Isso significa que a associação é **navegável** na direção da seta, ou seja, a partir de um objeto da classe A, pode-se obter o objeto da classe B associado.





Obrigado!

tsbarcelos@ifsp.edu.br