Projeto de Software

Recontextualização do Diagrama de Classes e Conceitos de Orientação por Objetos

Lesandro Ponciano

2024

Objetivos da Aula

- Recontextualizar o diagrama de classes em análise e projeto
- Recordar conceitos de orientação a objetos
- Discutir os componentes do diagrama de classes
- Apresentar exemplos de diagrama

Orientação a Objetos

- Grupo, suas características e elementos
- Classificação
  - A percepção de grupos, classes, categoria
- Abstração
  - Detectar termos gerais e perceber elementos, que embora diferentes, são partes de uma mesma classe
- Instanciação
  - Definição de um elemento que pertence ao grupo

Classes e Objetos

- Uma classe representa uma categoria
- Os objetos são os membros ou exemplos dessa categoria
  - São instâncias de classes
  - São abstrações de dados
  - Modelam entidades da aplicação
  - Tem estado (estrutura interna)
  - São manipulados apenas por operações
- Cada objeto implementa uma parte do comportamento da aplicação

Diagrama de Classes

- Diagrama parte da Linguagem de Modelagem Unificada (UML, *Unified Modeling Language*)
- É o diagrama central da modelagem orientada a objetos
- Tem ênfase nas classes e em seus relacionamentos
- É utilizado nas atividades de especificação do sistema e no projeto do sistema

Uso do Diagrama de Classes

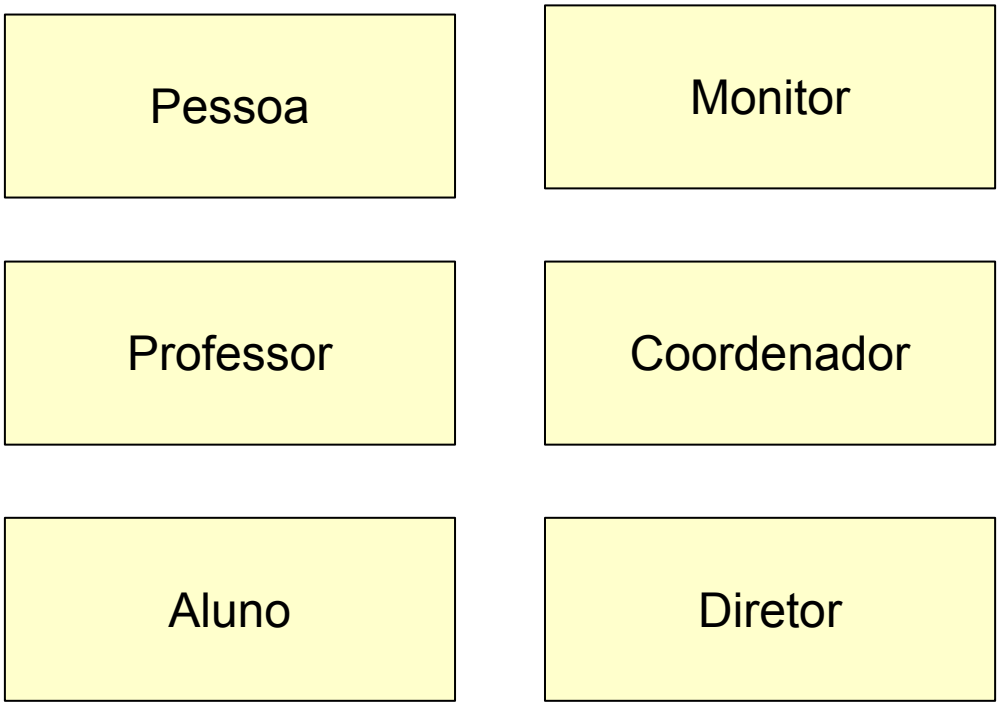
- Domínio do problema vs domínio da solução
  - Problema, representação de elementos do mundo real (ex.: Professor, Aluno, Turma)
  - Solução: Elementos lógicos (ex.: Pilha, Lista, Fila, Janela)
- Diagrama de Classe no Nível de Análise
  - Classes ligadas ao domínio do problema
  - Classes, atributos e operadores
- Diagrama de Classe no Nível de Projeto
  - Detalhes de atributos e operadores, parâmetros e retornos
  - Classes do domínio do problema e do domínio da solução

Estrutura do Diagrama de Classes

- As classes formam o núcleo de um software orientado a objetos

- Uma classe é a representação de um conjunto de objetos que compartilham os mesmos
  - Atributos
  - Operações
  - Relacionamentos

Exemplos de Classes

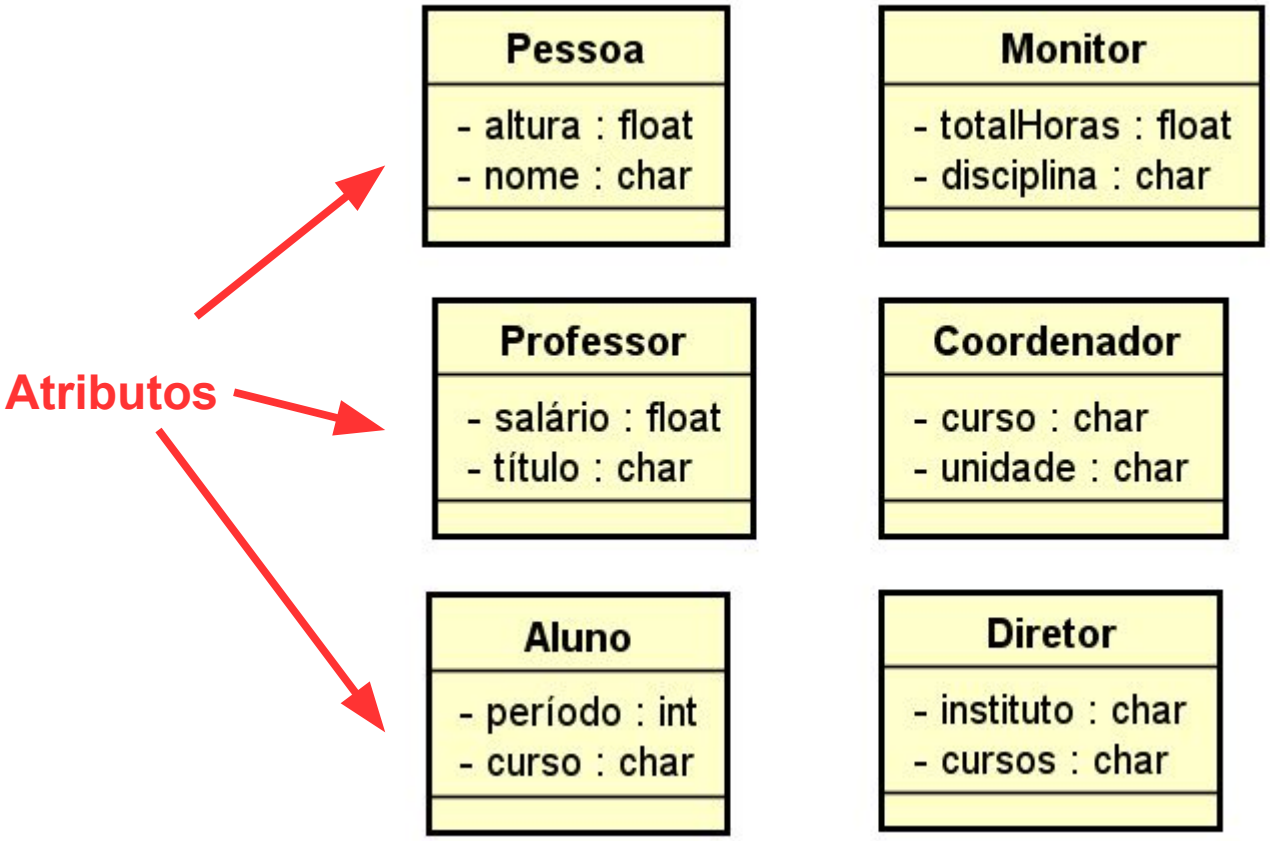


Atributos ou Propriedades

- Conjunto de valores que os objetos da classe podem atribuir a esse atributo

- Os atributos representam características de uma classe
  - Todas as instâncias de uma classe têm exatamente os mesmos atributos, mas os atributos podem assumir valores diversos
  - Peculiaridades que costumam variar de um objeto para outro
  - Permitem diferenciar entre objetos de mesma classe
  - Ex.: carro de cor vermelha vs. carro de cor verde

Exemplo de Classes e Atributos

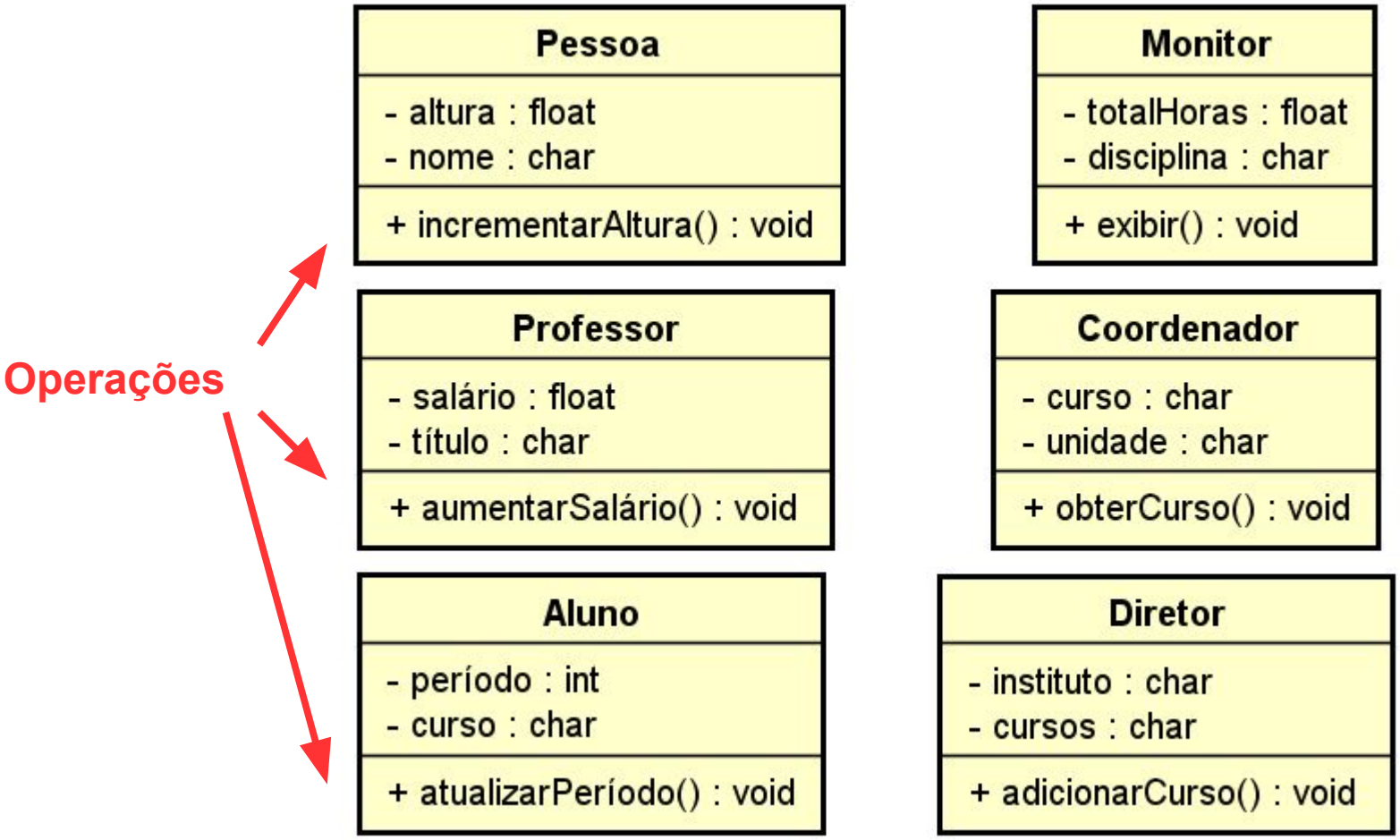


Métodos, Operações ou Comportamentos

- Uma operação representa uma atividade que um objeto de uma classe pode executar
  - Conjunto de instruções que são executadas quando a operação é chamada

- Uma operação é um serviço que pode ser requisitado por qualquer objeto da classe para obter um comportamento
  - Ex.: Atualizar período do Aluno

Exemplo de Operações

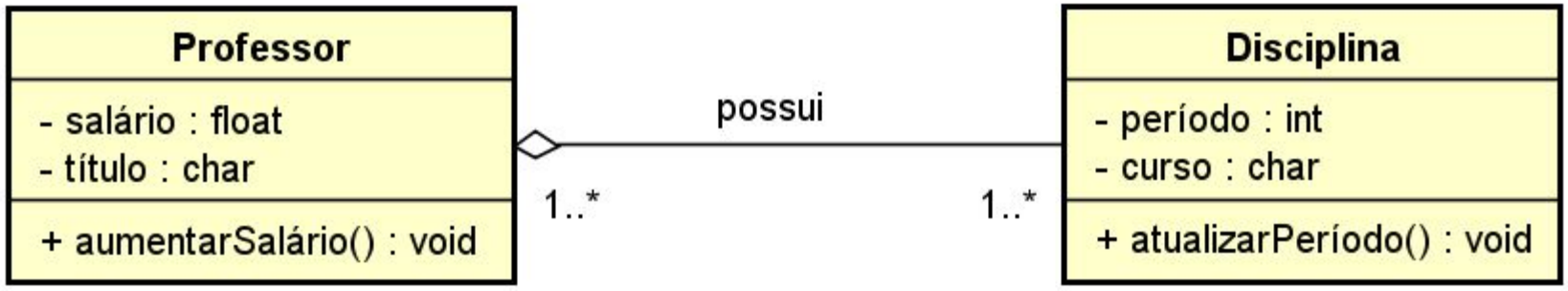


Relacionamento ou Associações

- As classes costumam ter relacionamentos, ou associações
  - Compartilhamento de informações
  - Colaboração para execução de processos
- O comportamento do sistema é obtido através da interação entre instâncias das classes
  - Monitor monitora Aluno
  - Professor leciona para Aluno
- Associações geralmente possuem nome
  - Texto que indica o significado da relação

Exemplo de Relacionamento

- Professor possui Disciplina



Multiplicidade

- Informação do limite inferior e superior da quantidade de objetos aos quais outro objeto pode estar associado
  - Professor possui uma ou muitas Disciplinas

Nome	Simbologia
Apenas Um	1
Zero ou Muitos	0..*
Um ou Muitos	1..*
Zero ou Um	0..1
Intervalo Específico	$l_i..l_s$

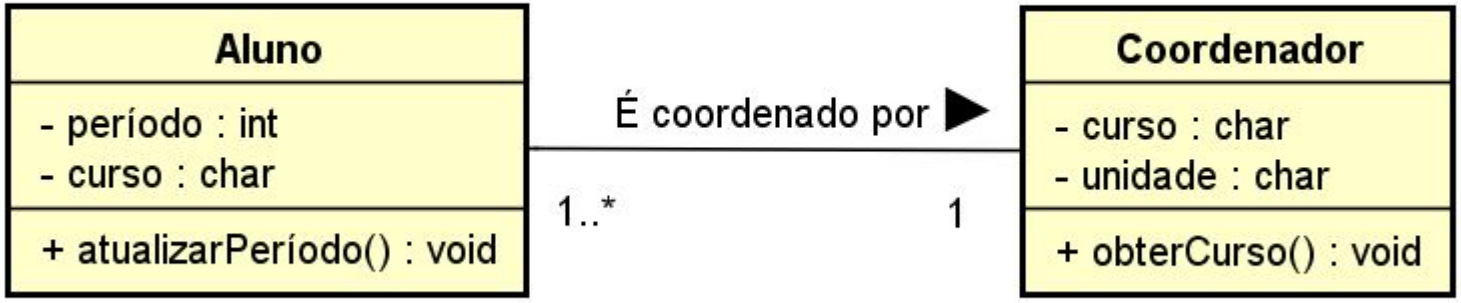
Conectividade

- Tipo de associação entre duas classes
  - Depende dos símbolos de multiplicidade que são utilizados na associação

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..*

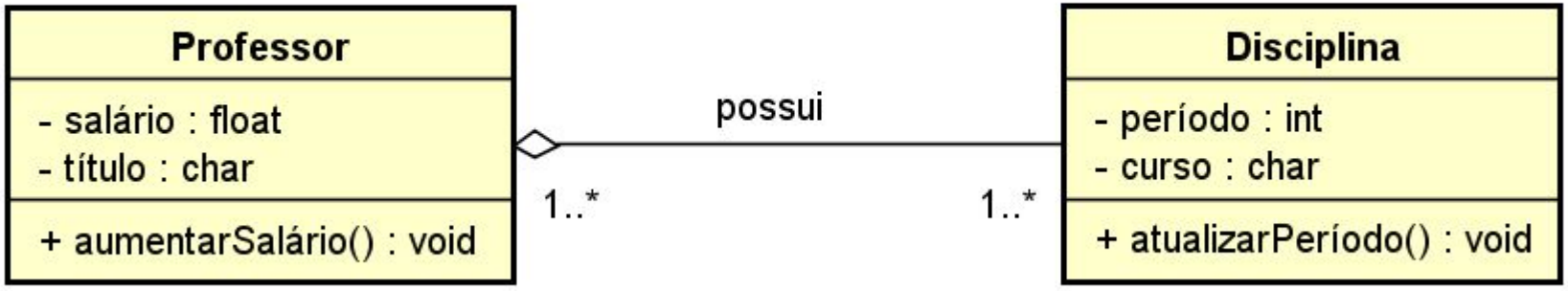
Associações Binárias

- Relacionamento entre duas classes



Agregação

- Agregação é uma forma especial de associação onde "o todo" está relacionado às suas "partes"
  - As partes complementam o todo
  - Losango vazio na extremidade da classe que contém o todo



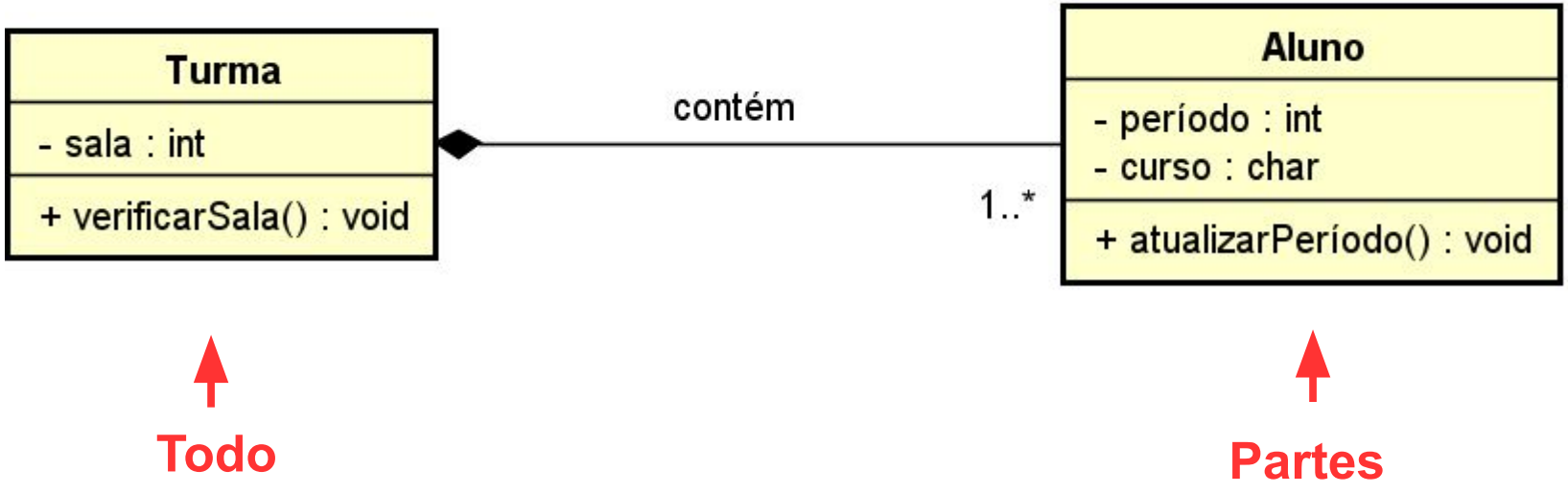
As informações de Professor são complementadas pelas informações de Disciplina

Composição

- É uma variação de agregação onde há um vínculo mais forte entre “o todo” e “suas partes"
  - Objeto-parte tem de estar associados a um único objeto-todo
  - Losango cheio na extremidade da classe que contém o todo
  - Ex.: A frase “é composto por” é utilizada para descrever o relacionamento
- O “todo” é chamado de classe forte e a “parte” é chamada de classe fraca
- Os ciclos de vida de objetos “todo” e “parte” são dependentes; a parte depende do todo para existir

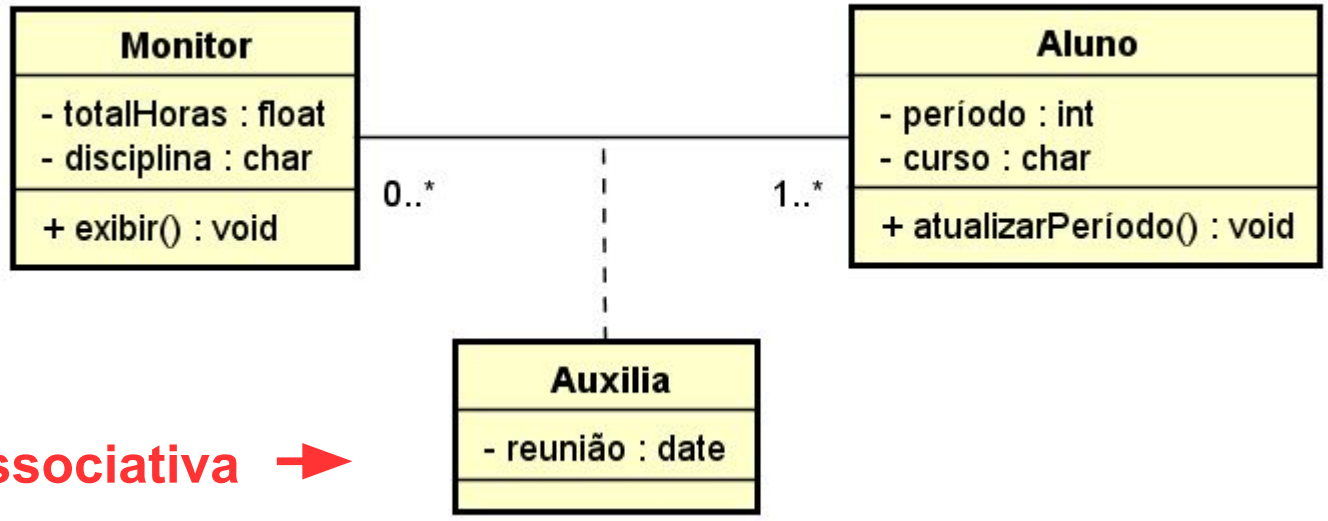
Composição

- Exemplo de composição
  - Turma é composta por alunos
  - Não existe aluno sem turma



Classe Associativa

- São aquelas produzidas quando da ocorrência de conectividade do tipo “muitos para muitos"



Generalização/Especialização

- Generalização/especialização
  - Tipo especial de relacionamento
  - Identificar a ocorrência de Herança entre classes
    - Existência de superclasse e subclasse

- Classes suportam os conceitos de:
  - Polimorfismo
  - Hierarquia

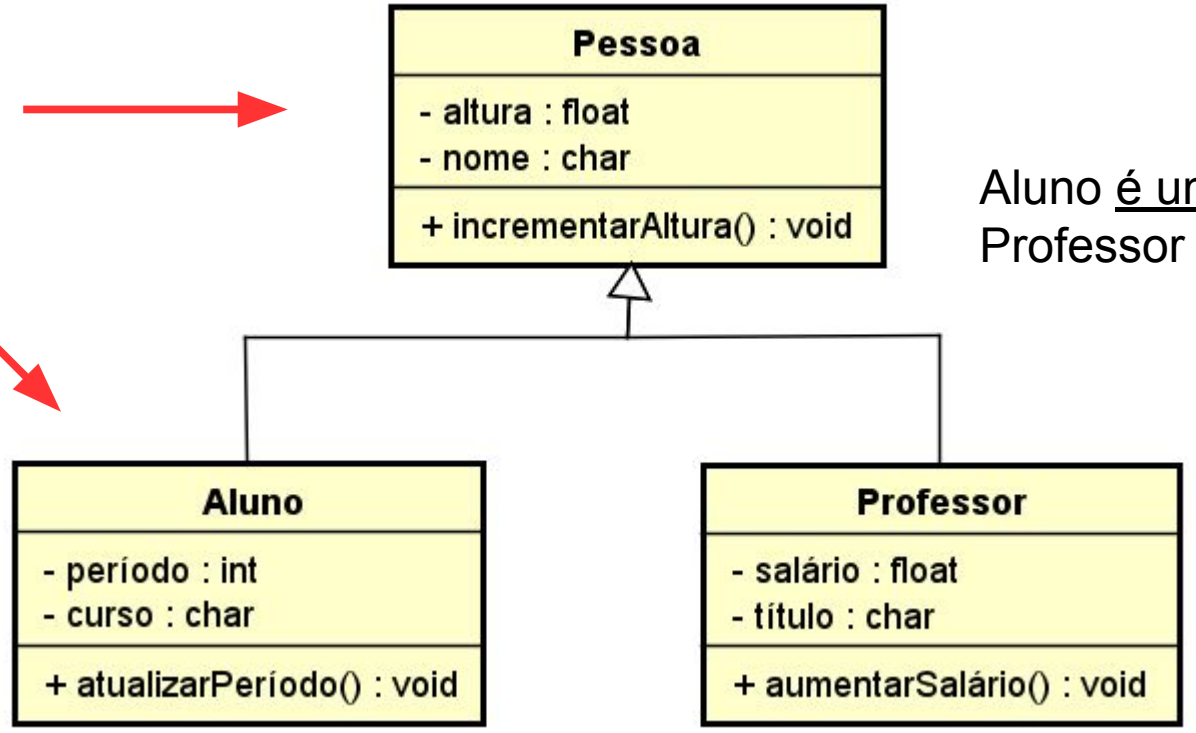
Herança

- Relacionamento entre classes onde uma classe compartilha a estrutura e o comportamento de uma ou mais classes
- Define uma hierarquia de abstrações na qual a subclasse herda de uma ou mais superclasses
  - Herança simples
  - Herança múltipla
- Uma herança é um relacionamento do tipo “é um tipo de”

Herança Simples

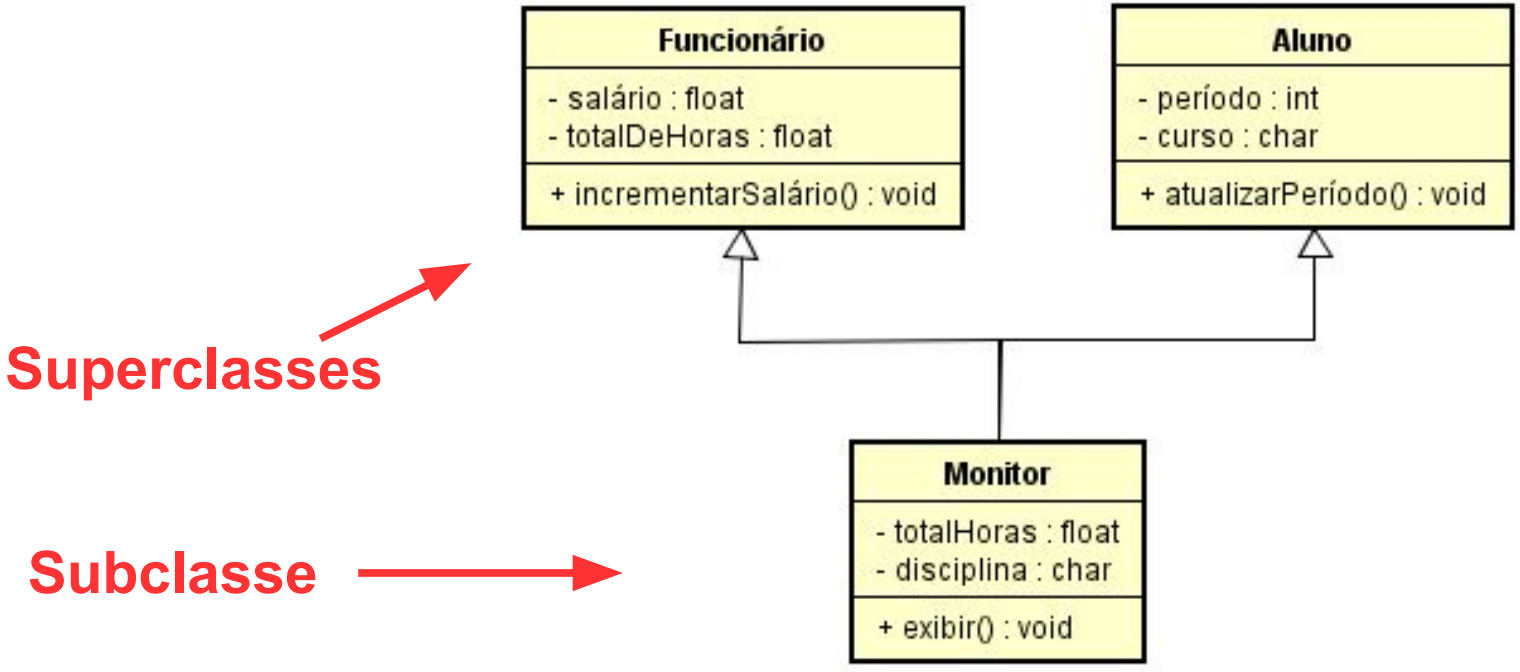
Superclasse

Subclasse



Aluno é um tipo de Pessoa
Professor é um tipo de Pessoa

Herança Múltipla



Monitor é um tipo de Aluno e é um tipo de Funcionário

Elementos Herdados

- A subclasse herda os atributos, operações e relacionamentos da superclasse

- Cada subclasse pode
  - definir novos atributos e/ou operações
  - redefinir operações da superclasse
  - participar de relacionamentos específicos

Polimorfismo

- Um mesmo objeto pode ser de vários tipos
  - Ex.: Uma Pessoa pode ser um Estudante ou um Professor
- Não é viável exigir que todos os outros objetos saibam todos os possíveis tipos de um determinado objeto
- Através do polimorfismo
  - o comportamento do objeto será definido pela reimplementação contida no objeto
  - instâncias de várias classes são tratadas de forma única em um sistema
  - todos os outros objetos devem reconhecer o objeto através de um único tipo

Atividade de Fixação

A relação existencial de dependência existente entre Estado e Municípios, de modo que não existe Estado sem Municípios, é descrita na relação entre classes na orientação por objetos como:

A) Herança
B) Polimorfismo
C) Agregação
D) Composição

Prática de Fixação
Sistema JF

US0 - Como professor, quero me cadastrar no sistema. Nesse cadastro, informo nome, login, senha e e-mail. A cada novo acesso ao sistema, as funcionalidades do sistema se tornam disponíveis após a autenticação via login e senha cadastrados. O login é um código alfanumérico entre 5 e 10 caracteres. A senha é uma sequência de caracteres que deve conter letras e números, obrigatoriamente. Meu login é único no sistema e não pode ser alterado uma vez definido. Os outros dados podem ser alterados.

US1 - Como aluno, quero me cadastrar no sistema. Nesse cadastro, informo nome, login, e-mail, curso e senha. A cada novo acesso ao sistema, as funcionalidades do sistema se tornam disponíveis após a autenticação via login e senha cadastrados. O login é um código alfanumérico entre 4 e 8 caracteres. A senha é uma sequência de caracteres que deve conter apenas números. Meu login é único no sistema e não pode ser alterado uma vez definido. Os outros dados podem ser alterados.

US2 - Como professor, quero uma funcionalidade de cadastrar turma. Uma turma tem uma disciplina, um código, um curso, uma unidade da universidade e os alunos que fazem parte da turma. Se eu sei o código do aluno, posso informá-lo para adicionar à turma. Se eu não sei esse código, devo ser capaz de pesquisar aluno cadastrado no sistema no referido curso informando o seu nome e, a partir do resultado da pesquisa, posso adicionar o aluno à turma.

Referências

- SOMMERVILLE, Ian. Engenharia de Software - 9a edição. Pearson 548 ISBN 9788579361081 (Capítulo 5)

- GUEDES, Gilleanes T. A. UML 2: uma abordagem prática. 2. ed. São Paulo: Novatec, c2011. 484 p. ISBN 9788575222812 (Capítulo 2 e 4)

- Bezerra, Eduardo. Princípios de Análise e Projeto de Sistema com UML. Vol. 3. Elsevier Brasil, 2007. (Capítulo 5)

- Os diagramas foram feitos usando a ferramenta Astah <http://astah.net/editions>

Projeto de Software

Prof. Dr. Lesandro Ponciano

https://orcid.org/0000-0002-5724-0094