

Associação, Agregação e Composição

Programação Orientada a Objetos — QXD0007



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS QUIXADÁ

Prof. Atílio Gomes Luiz
gomes.atilio@ufc.br

Universidade Federal do Ceará

1º semestre/2022



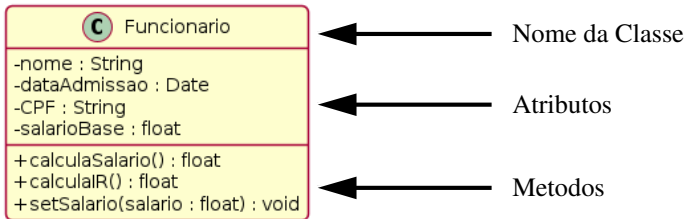
Leituras para esta aula

- **Seções 5.1 e 5.2** do livro **Princípios de Análise e Projeto de Sistemas com UML**, Eduardo Bezerra, Segunda Edição. Disponível no link:
<http://library.lol/main/375E243CF3C35569CD1231E96092F7B9>

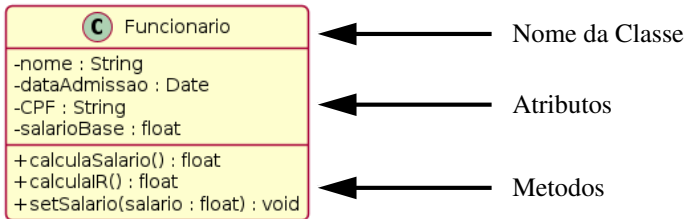
UML e Diagrama de Classes

- **Unified Modeling Language** é uma linguagem-padrão para a elaboração da estrutura de projetos de software.
- O **Diagrama de classes** é um dos tipos mais úteis de diagramas UML.
 - Oferece uma notação gráfica formal para a modelagem de classes e seus relacionamentos.
- Uma classe refere-se à descrição de um conjunto de objetos que compartilham os mesmos atributos, operações e relacionamentos.

Notação UML para classes



Notação UML para classes



- **Visibilidade** (modificadores de acesso):

- + **public**: o que pode ser visto pelas operações de outras classes.

- **private**: o que pode ser visto apenas pelas operações da própria classe.

- # **protected**: o que pode ser visto pelas operações da própria classe e por suas classes herdeiras.

- ~ **package**: o que pode ser visto pelas operações da própria classe e pelas classes no mesmo pacote.

O que vimos até agora

- Até o presente momento, lidamos com objetos simples, compostos por atributos de tipos primitivos, ou que se relacionam apenas com outro objeto.
- Um sistema orientado a objetos é composto por dezenas de classes que se relacionam entre si para solucionar o problema proposto.
- Objetos do mundo real relacionam-se uns com os outros de diversas formas:
 - Um objeto motor **é parte de** um objeto carro
 - Um objeto turma **tem vários** objetos alunos
 - Um objeto botão **tem um** objeto tratador de eventos

Relacionamento entre objetos

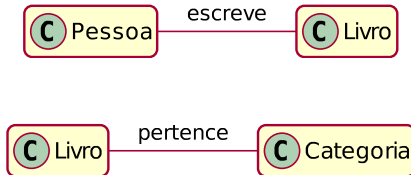
- Tipos de relacionamento:
 - Associação
 - Agregação
 - Composição
 - Herança

Associação

- Uma **associação** é um relacionamento entre duas ou mais classes.
- Uma associação indica que os objetos de uma classe têm um relacionamento como os objetos de outra classe, em que essa conexão tem um significado especificamente definido.

Associação

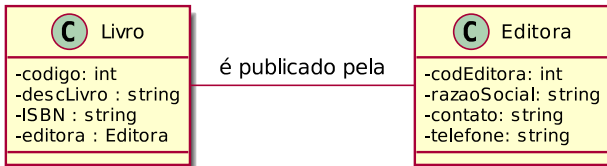
- Uma **associação** é um relacionamento entre duas ou mais classes.
- Uma associação indica que os objetos de uma classe têm um relacionamento como os objetos de outra classe, em que essa conexão tem um significado especificamente definido.



Representamos as associações por meio de **linhas retas** que ligam as classes envolvidas. A associação pode também receber um **nome**.

Associação

- A forma mais comum de implementar associação é ter um objeto como atributo de outro.



No exemplo acima, temos uma associação entre a classe **Livro** e a classe **Editora**. No código, cria-se um objeto do tipo **Livro** e outro do tipo **Editora**. Um dos atributos do **Livro** é a **Editora**.

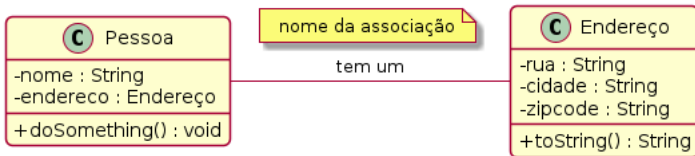
Características da associação



Características da Associação

- Nome
- Multiplicidade
- Conectividade
- Navegabilidade
- Papéis

Nome da associação



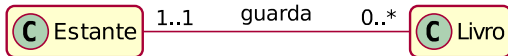
- O nome da associação é posicionado no meio da linha da associação.
- Ele deve fornecer algum significado semântico à mesma.

Multiplicidades

- A **multiplicidade** define quantas instâncias de uma classe **A** podem estar relacionadas a uma instância da classe **B**.

Multiplicidades

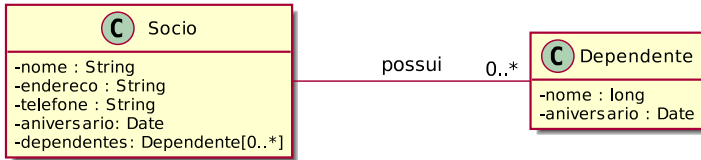
- A **multiplicidade** define quantas instâncias de uma classe **A** podem estar relacionadas a uma instância da classe **B**.
- Cada associação em um diagrama de classes possui duas multiplicidades, uma em cada extremo da linha que a representa.



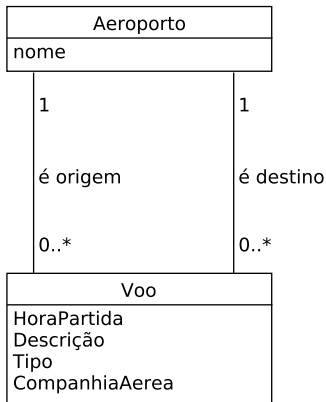
Multiplicidades

Multiplicidade	Significado
0..1	Zero ou um. Indica que os objetos das classes associadas não precisam obrigatoriamente estar relacionados, mas se estiverem, indicará que apenas uma instância da classe relaciona-se com as instâncias da outra classe
1 ou 1..1	Um. Indica que apenas um objeto da classe relaciona-se com os objetos da outra classe
0..* ou *	Zero ou muitos. Indica que pode ou não haver instâncias da classe participando do relacionamento
1..*	Um ou muitos. Indica que há pelo menos um objeto envolvido no relacionamento, podendo haver muitos objetos envolvidos
$m..n$	No mínimo m e no máximo n, onde $m \leq n$.

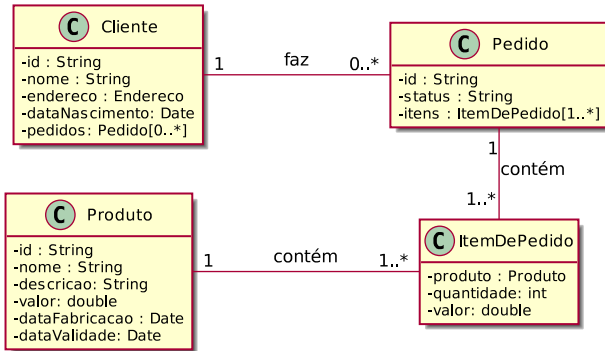
Multiplicidade de associações – Exemplo



Multiplicidade de associações – Exemplo



Multiplicidade de associações – Exemplo

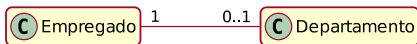


Conectividade

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..*

Conectividade

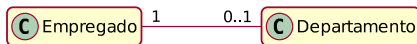
Conectividade	Multiplicidade de um extremo	Multiplicidade do outro extremo
Um para um	0..1 ou 1	0..1 ou 1
Um para muitos	0..1 ou 1	* ou 1..* ou 0..*
Muitos para muitos	* ou 1..* ou 0..*	* ou 1..* ou 0..*



um empregado pode gerenciar no máximo um departamento

Conectividade

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..*



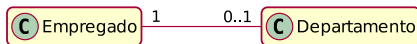
um empregado pode gerenciar no máximo um departamento



um empregado está lotado em um único departamento, mas um departamento pode ter diversos empregados

Conectividade

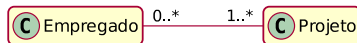
Conectividade	Multiplicidade de um extremo	Multiplicidade do outro extremo
Um para um	0..1 ou 1	0..1 ou 1
Um para muitos	0..1 ou 1	* ou 1..* ou 0..*
Muitos para muitos	* ou 1..* ou 0..*	* ou 1..* ou 0..*



um empregado pode gerenciar no máximo um departamento



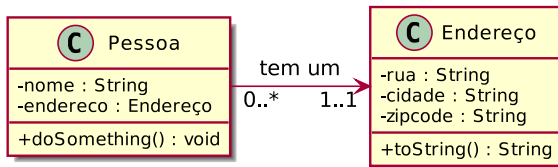
um empregado está lotado em um único departamento, mas um departamento pode ter diversos empregados



um projeto pode ter diversos empregados. Além disso, um empregado pode trabalhar em diversos projetos

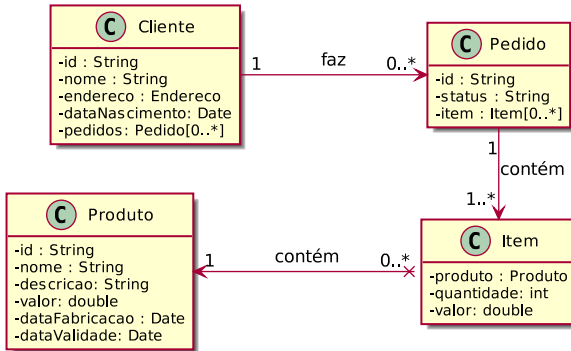
- A **navegabilidade** indica como a associação deve ser lida.
- As associações entre classes UML podem ter uma direção navegável, de uma classe para a outra ou em ambas as direções (bidirecional).
- As associações também podem ter direção não especificada, que na maioria dos casos será a mesma que bidirecional, mas esta opção pode gerar dúvidas.
- A navegabilidade descreve como as implementações do modelo podem ser navegadas e é simbolizada por setas no diagrama.

Navegabilidade



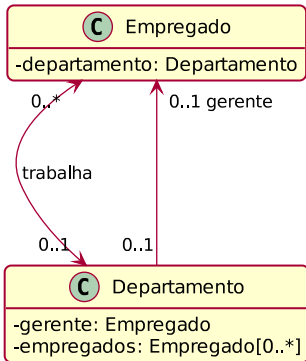
- Pessoa sabe o seu Endereço
- Endereço não sabe quem vive lá

Navegabilidade



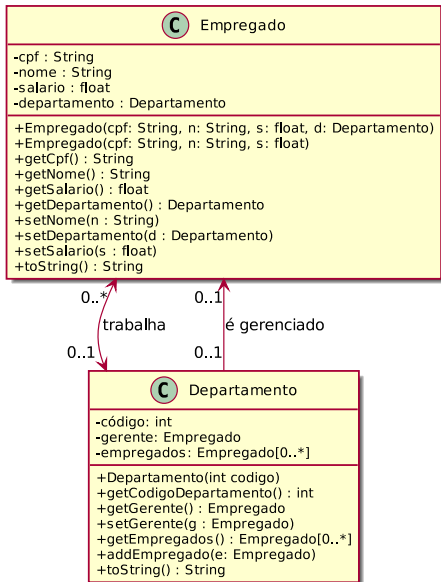
- Quando um objeto participa de uma associação, ele tem um **papel** específico nela.





Um departamento precisa saber quais são seus empregados e quem é o seu gerente. Nesse caso, há duas associações diferentes, embora as classes envolvidas sejam as mesmas.

Exercício 1



Implementar as classes correspondentes ao diagrama ao lado.

Obs.: exemplos de departamentos são: Administrativo, Recursos Financeiros, Setor Comercial, Recursos Humanos, entre outros.

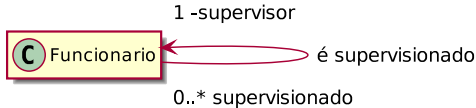
Crie um programa que cadastre vários empregados para os departamentos da empresa. Antes de cadastrar os empregados, os departamentos já devem ter sido criados.

Associações Reflexivas



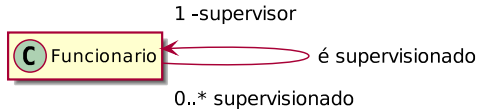
Associações Reflexivas

- Uma **associação reflexiva** associa objetos da mesma classe.
- Cada objeto tem um papel distinto nessa associação.

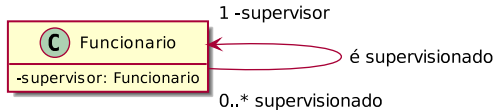


Associações Reflexivas

- Uma **associação reflexiva** associa objetos da mesma classe.
- Cada objeto tem um papel distinto nessa associação.



Um refinamento do diagrama:



Agregação e Composição



Relação todo-parte

- Uma **relação todo-parte** entre dois objetos indica que um dos objetos **está contido** no outro. Ou então, que as informações de um objeto (o todo) é composta pelas informações de um outro objeto (a parte).

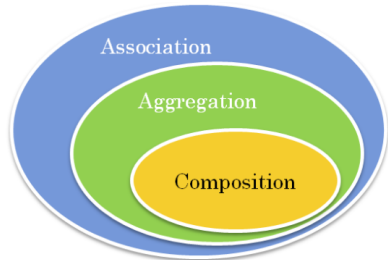


Relação todo-parte

- Uma **relação todo-parte** entre dois objetos indica que um dos objetos **está contido** no outro. Ou então, que as informações de um objeto (o todo) é composta pelas informações de um outro objeto (a parte).



Agregação e **Composição** são dois tipos especiais de associação que relacionam um objeto (o todo) com suas partes.



Agregação

- A **agregação** é um caso especial de associação que relaciona um objeto (o todo) com suas partes.



Uma classe agrega outra (não exclusivamente)

Agregação

- A **agregação** é um caso especial de associação que relaciona um objeto (o todo) com suas partes.



Uma classe agrega outra (não exclusivamente)

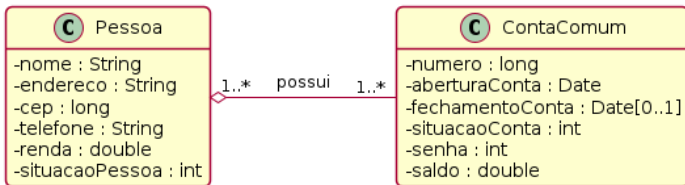
Na agregação, a destruição de um objeto **todo** não implica necessariamente a destruição do objeto **parte**. Por exemplo, na relação acima, a remoção de uma estante não necessariamente implica na destruição dos livros guardados nela.

Agregação – Exemplo



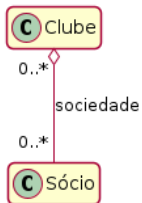
- Uma associação esportiva é formada por várias equipes.
Cada equipe é formada por diversos jogadores.
Por outro lado, um jogador pode fazer parte de diversas equipes.
 - Se uma das equipes das quais o jogador é membro for extinta, este jogador ainda poderá continuar membro de outras equipes.

Agregação – Exemplo



- Uma pessoa pode possuir muitas contas e uma conta pode ser possuída por muitas pessoas, como no caso de uma conta conjunta.
- Isso é característico das agregações nas quais os objetos-parte podem ser compartilhados por mais de um objeto todo.

Agregação – Exemplo



- Analisar projeto [SocioClube](#)

Composição

- A **composição** é um caso especial de associação que relaciona um objeto (o todo) com suas partes. Porém, em uma composição, a destruição do objeto **todo** implica a destruição do objeto **parte**.



Relação existencial (exclusiva) entre a parte e o todo

Composição

- A **composição** é um caso especial de associação que relaciona um objeto (o todo) com suas partes. Porém, em uma composição, a destruição do objeto **todo** implica a destruição do objeto **parte**.



Relação existencial (exclusiva) entre a parte e o todo

- Na composição, os objetos **parte** pertencem a um único **todo**. Por essa razão, a composição é também denominada **agregação não compartilhada**.

Composição

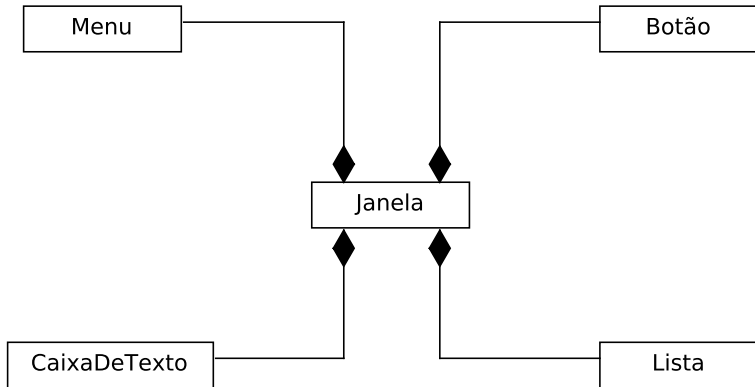
- A **composição** é um caso especial de associação que relaciona um objeto (o todo) com suas partes. Porém, em uma composição, a destruição do objeto **todo** implica a destruição do objeto **parte**.



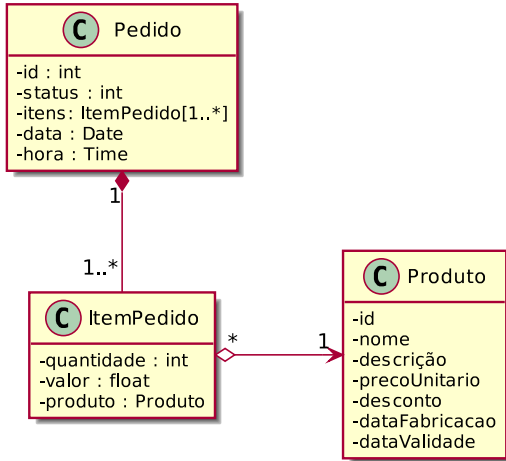
Relação existencial (exclusiva) entre a parte e o todo

- Na composição, os objetos **parte** pertencem a um único **todo**. Por essa razão, a composição é também denominada **agregação não compartilhada**.
- Por outro lado, em uma agregação, pode ser que um objeto participe como componente de vários outros objetos. Por essa razão, a agregação é também denominada **agregação compartilhada**.

Composição – Exemplo

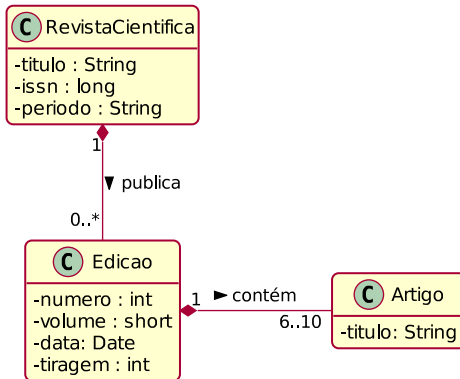


Composição – Exemplo

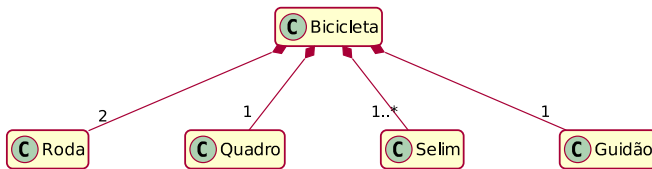


- Os itens não têm existência independente do pedido ao qual estão conectados. Quando o pedido deixa de existir, o mesmo acontece com os seus itens.

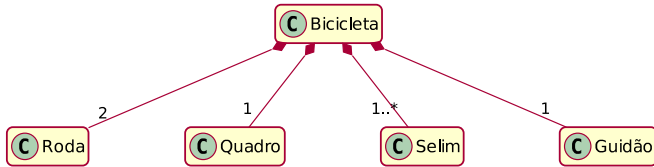
Composição – Exemplo



Composição – Exemplo



Composição – Exemplo



Analisar o projeto **Bike**

Exercício

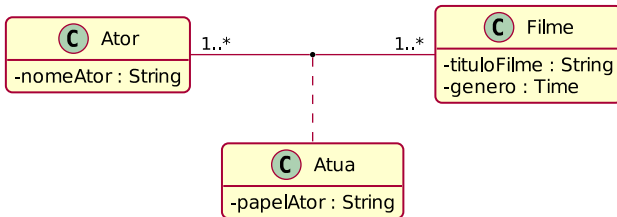
- Fazer o exercício que está no Moodle: Cinema e Vetor de Null

Classe Associativa



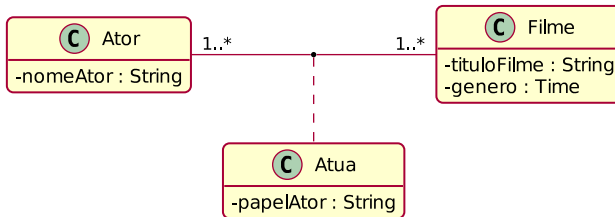
Classes Associativas

- São necessárias quando existem atributos relacionados à associação que não podem ser armazenados por nenhuma das classes envolvidas.
- São produzidas quando ocorrem associações com multiplicidade muitos (*) em todas as extremidades.



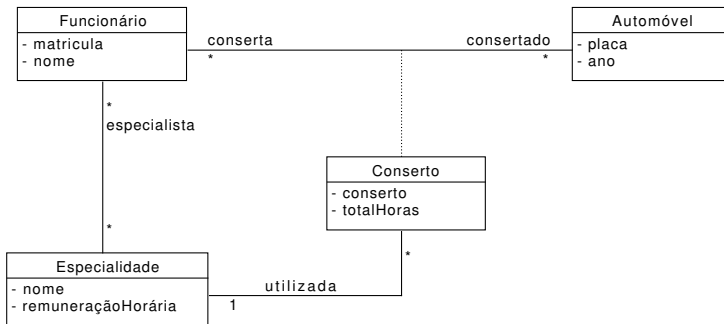
Classes Associativas

- São necessárias quando existem atributos relacionados à associação que não podem ser armazenados por nenhuma das classes envolvidas.
- São produzidas quando ocorrem associações com multiplicidade muitos (*) em todas as extremidades.



Um ator pode atuar em muitos filmes e um filme pode ter vários atores atuando nele. Existe a necessidade de saber qual o papel interpretado por um ator em um determinado filme, mas onde armazenar essa informação?

Classes Associativas



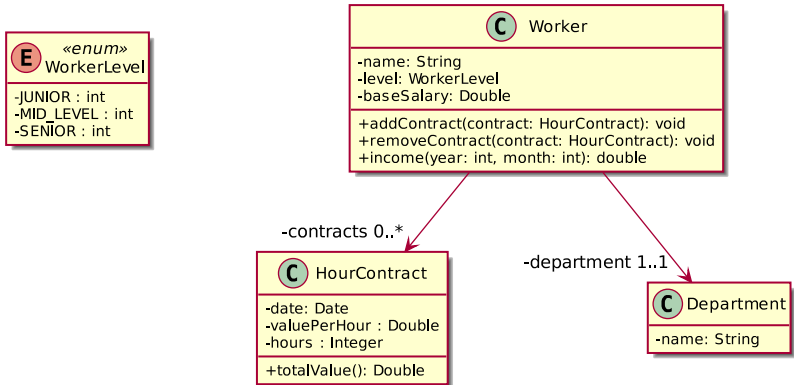
- Há funcionários com várias especialidades. Consertos em automóveis são realizados por funcionários, mas é necessário saber que especialidade foi utilizada pelo funcionário em certo conserto. Para isso, uma classe associativa é criada entre as classes **Funcionário** e **Automóvel**. Além disso, essa mesma classe associativa está associada à classe **Especialidade** para permitir conhecer qual a especialidade utilizada em um conserto.

Exercícios



Exercício 2

Ler os dados de um trabalhador com N contratos (N fornecido pelo usuário). Depois, solicitar do usuário um mês e mostrar qual foi o salário do funcionário nesse mês, conforme exemplo da próxima página.



Exercício 2 (cont.)

Entre o nome do departamento: **Design**

Entre os dados do trabalhador:

Nome: **Alex**

Nível: **MID_LEVEL**

Base salary: **1200.00**

Quantos contratos para este trabalhador? **3**

Entre os dados do contrato 1:

Data (DD/MM//YYYY): **20/08/2018**

Valor por hora: 50.00

Duração (horas): **20**

Entre os dados do contrato 2:

Data (DD/MM//YYYY): **13/06/2018**

Valor por hora: 30.00

Duração (horas): **18**

Entre os dados do contrato 3:

Data (DD/MM//YYYY): **25/08/2018**

Valor por hora: 80.00

Duração (horas): **10**

Entre o mês e ano para calcular o ganho (MM/YYYY): **08/2018**

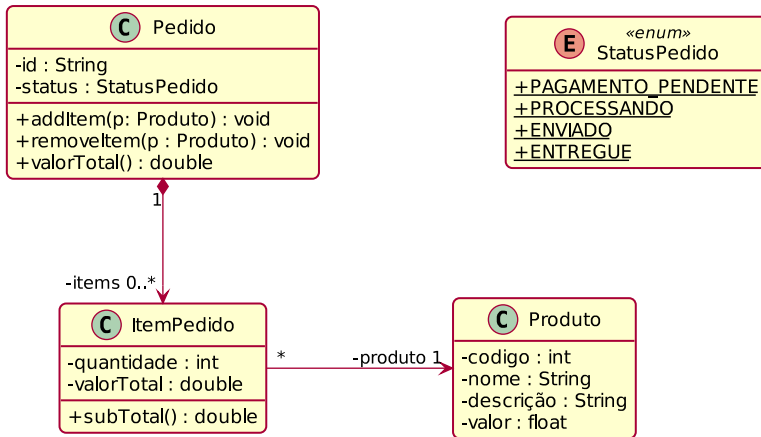
Nome: Alex

Departamento: Design

Ganho em 08/2018: 3000.00

Exercício 3

Implemente as classes correspondentes ao diagrama de classes abaixo.



FIM

