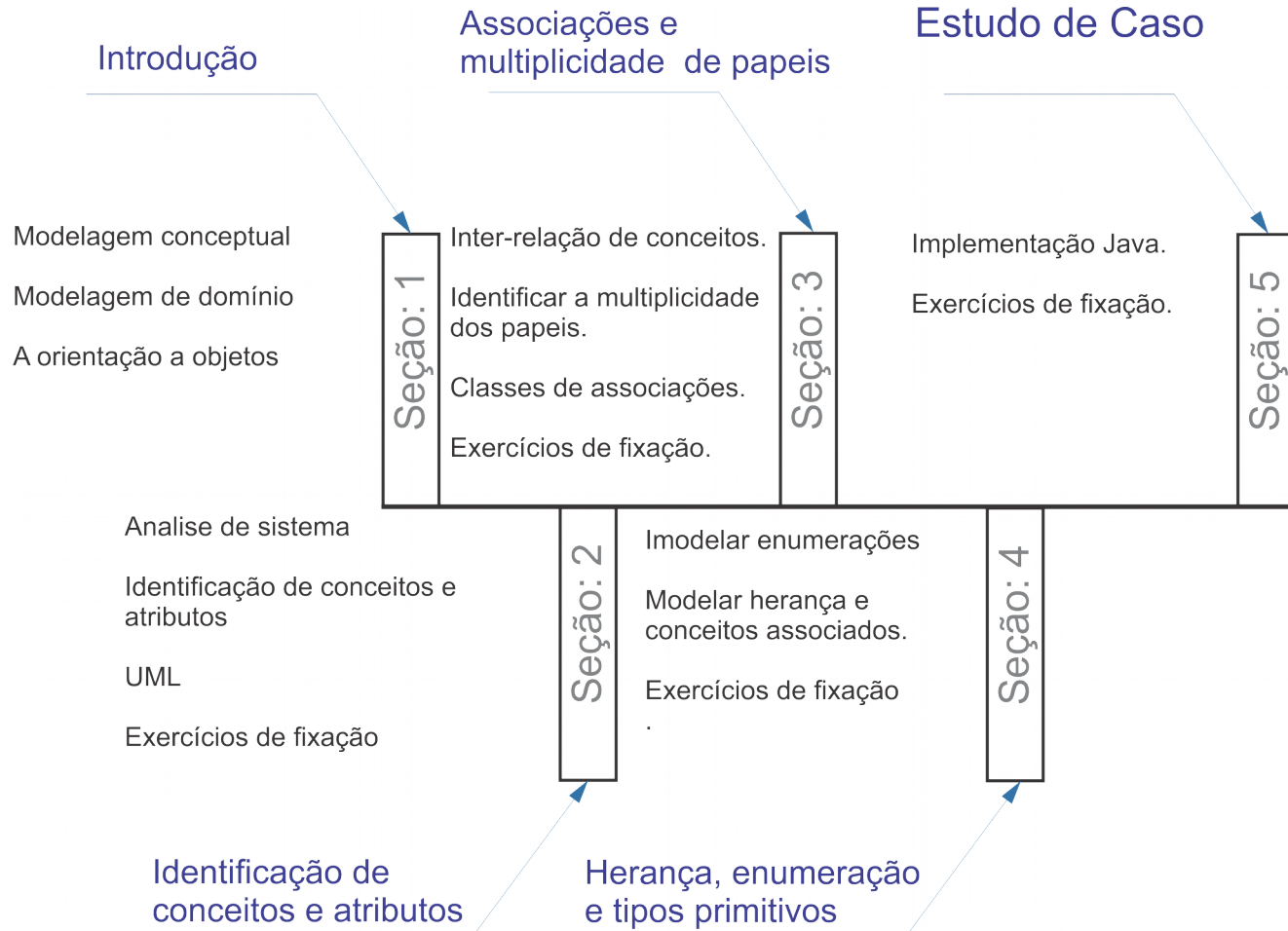


Técnicas de Programação 1

Professor Cristhian Riaño

Tópicos

- Decomposição de problemas usando classes e objetos.
- Reuso com herança e composição de classes.
- Polimorfismo por subtipos/parametrizado.
- Tratamento de exceções.
- Técnicas de desenho e programação OO.
- Tópicos especiais (distribuição, concorrência).
- Técnicas avançadas de modularização.



Tópicos

- Decomposição de problemas usando classes e objetos.
 - Definição de modelo conceitual.
 - Representação UML de conceitos e atributos.
 - Exercícios.

Exercício 1 – Sistema de Controle Cinema

Um cinema pode ter muitas salas, sendo necessário, por tanto, registrar informações a respeito de cada uma, como sua capacidade, ou seja, o numero de assentos disponíveis.

O cinema apresenta muitos filmes. Um filme tem informações, titulo e duração. Assim, sempre que um filme for ser apresentado, deve-se registrá-lo também.

Um mesmo filme pode ser apresentado em diferentes salas e em horários diferentes. Cada apresentação em uma determinada sala e horário é chamada sessão. Um filme sendo apresentado em uma sessão tem um conjunto máximo de ingressos, determinado pela capacidade da sala.

Os clientes do cinema podem comprar ou não ingressos para assistir a uma sessão. O funcionário deve intermediar a compra do ingresso. Um ingresso deve conter informação como o tipo de ingresso (Meio ingresso ou ingresso inteiro). Além disso, um cliente só pode comprar ingressos para sessões ainda não encerradas.

Exercício 2 – Sistema Registro Artistas Musicais

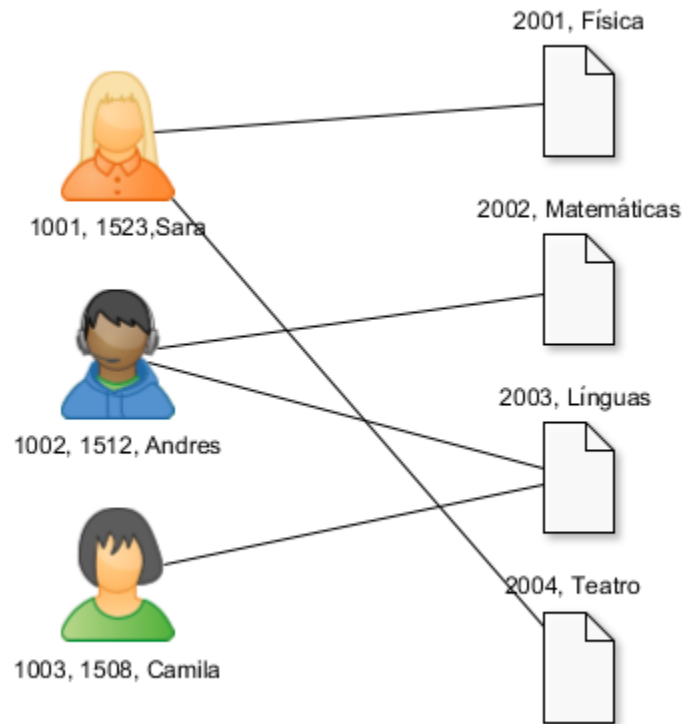
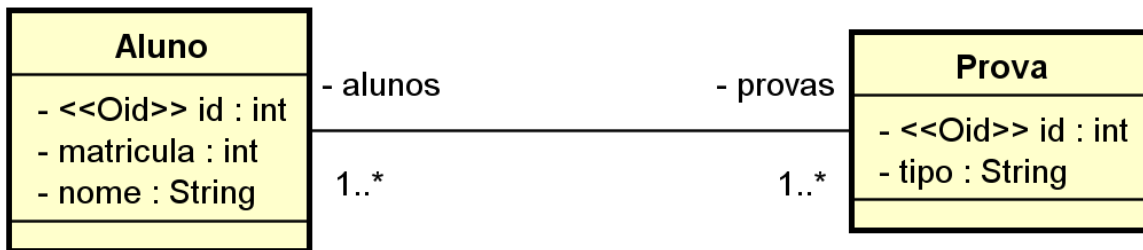
Deseja-se construir um sistema para manter um registro de artistas musicais e seus álbuns. Cada álbum possui várias músicas, as quais poderão ser consultadas pelo sistema. O sistema também deve permitir a busca de artistas por nome ou nacionalidade. O sistema também deve ser capaz de exibir um relatório dos álbuns de um artista, o qual pode ser ordenado por nome, ano, ou duração total do álbum. Um álbum pode ter a participação de vários artistas, sem distinção. Já a música pode possuir um ou mais autores e intérpretes (todos são considerados artistas).

Exercício 3 – Sistema para Campeonatos

Deseja-se construir um sistema para gerenciar as informações de campeonatos de handebol, que ocorrem todo ano. Deseja-se saber nome, data de nascimento, gênero e altura dos jogadores de cada time, bem qual deles é o capitão de cada time. Cada partida do campeonato ocorre em um estádio, que possui nome e endereço. Cada time possui seu estádio-sede e, assim, cada partida possui um time mandante (anfitrião) e o time visitante. O sistema deve ser capaz de listar as partidas já ocorridas e não ocorridas de um campeonato. O sistema deve também ser capaz de listar a tabela do campeonato, ordenando os times por classificação, que é calculada em primeiro lugar por saldo de vitórias e em segundo lugar por saldo de gols.

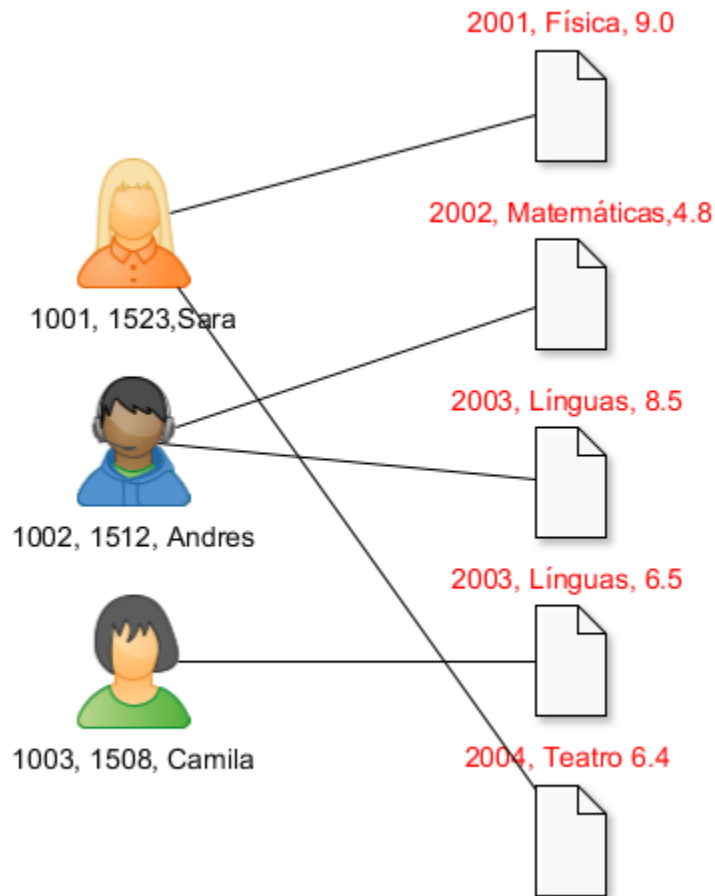
Classe de associação

Desejasse fazer um sistema para manter o cadastro de alunos e provas bem como informação de qual aluno aplicou cada prova.



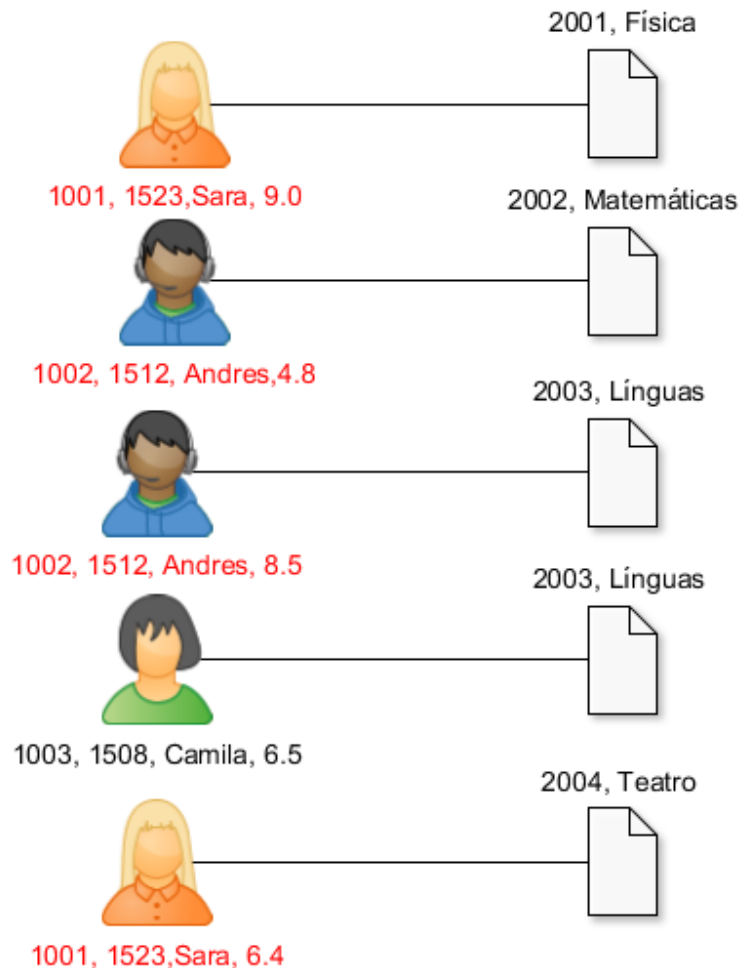
Classe de associação

Além disso, desejo saber também o resultado da prova obtido por cada aluno.



Classe de associação

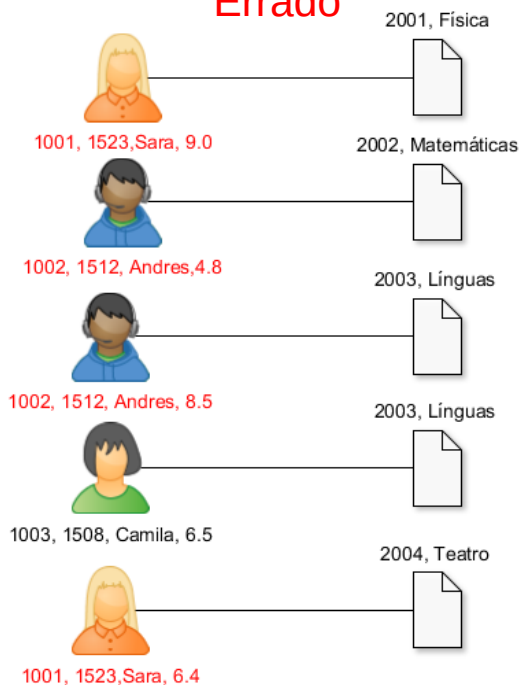
Além disso, desejo saber também o resultado da prova obtido por cada aluno.



Classe de associação

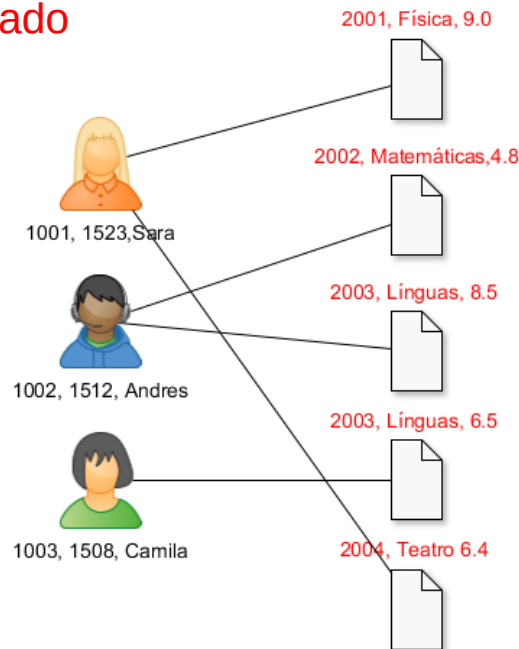
Além disso, desejo saber também o resultado da prova obtido por cada aluno.

Errado



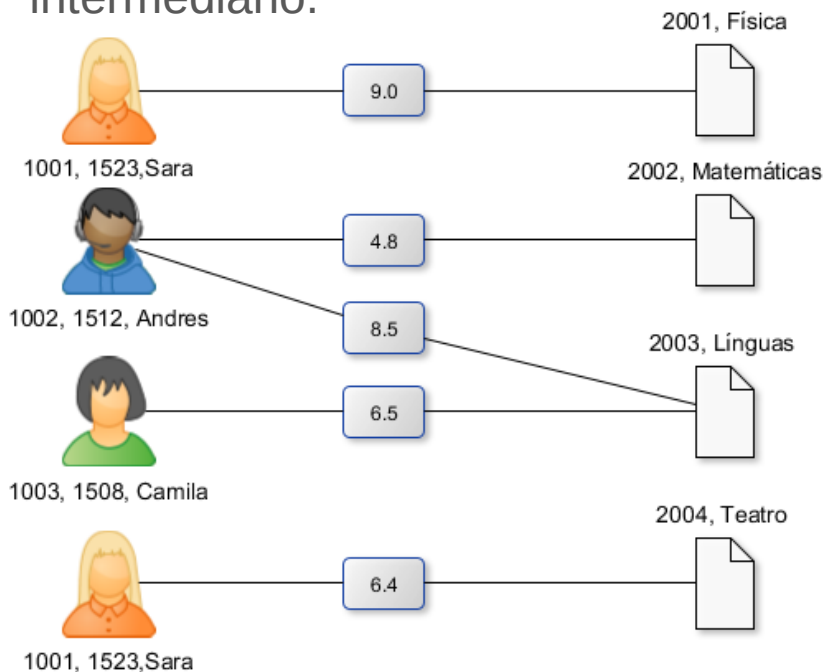
Esse valor não pode ser um atributo do objeto **aluno**, porque um aluno pode fazer varias provas. O valor também não pode ser um atributo da classe **prova**, porque cada prova pode ser feita por mais de um aluno.

Errado

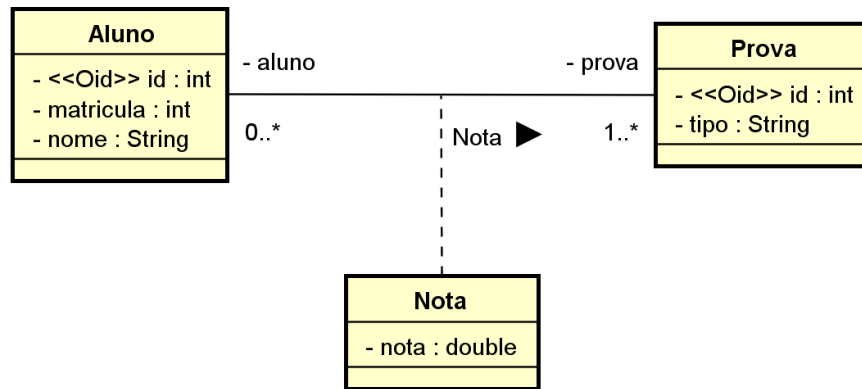


Classe de associação

Conclusão: O valor da nota obtida pelo aluno, é um atributo de relacionamento, neste caso, o nome do personagem deveria ser armazenado em um objeto intermediário.

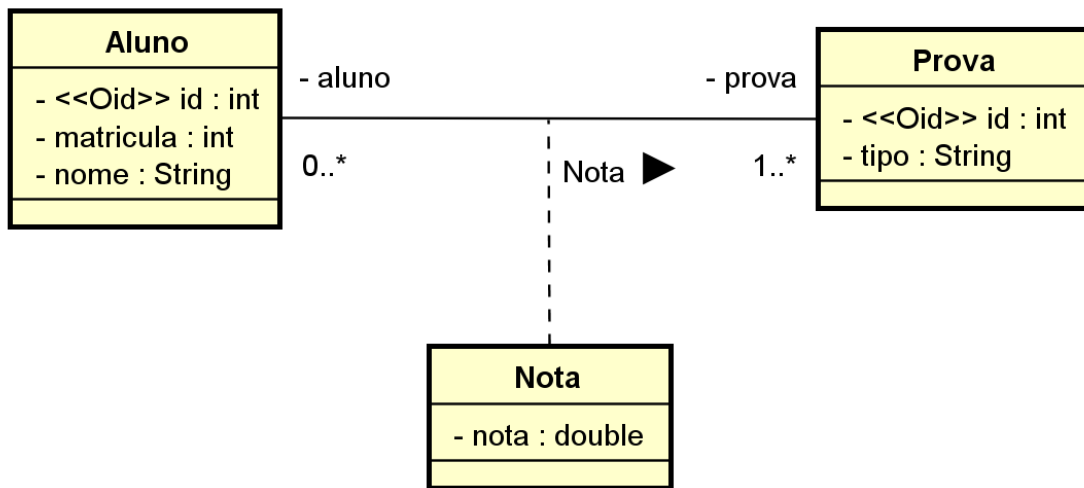


Esses atributos são mais comuns nos relacionamentos do tipo “muitos para muitos”, porque, nos relacionamentos um para muitos, o atributo pode algumas vezes ser colocado no participante insolado.



Classe de associação

- Este modelo indica que para um mesmo aluno e uma mesma prova só pode haver uma nota.
- Cada nota continua associada com apenas um aluno e uma prova.
- Não tem identificador.

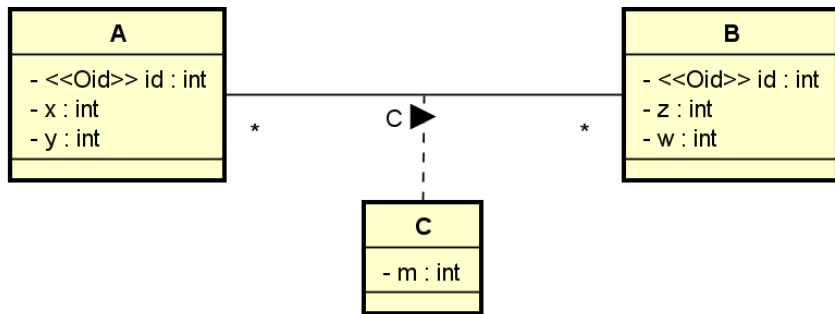


Classe de associação

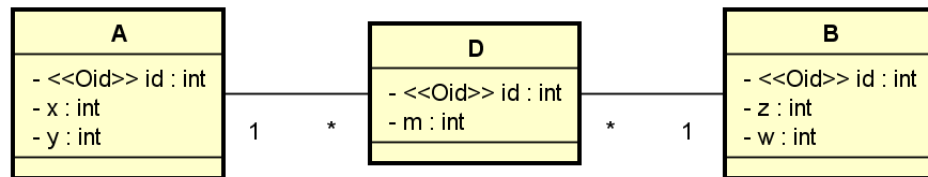


Quando tem associações muitos-para-muitos verificar:

- Tem atributos não associação?
- O mesmo “A” pode estar associado com o mesmo “B”, mais de uma vez?



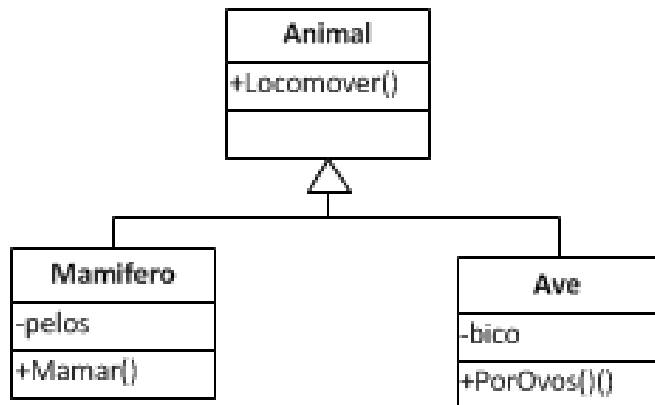
Não



Sim

Herança

A herança permite o reaproveitamento de atributos e de métodos, otimizando o tempo de desenvolvimento. Herança se relaciona com os conceitos de superclasses e subclasses. As subclasses a serem derivadas a partir de uma superclasse, herdam suas características, ou seja, seu atributos e métodos.



Herança

Em uma herança os atributos e os relacionamentos pertencentes às superclasses serão “herdados” por todas as subclasses. Cada subclasse pode possuir atributos adicionais e participar de outros relacionamentos.

Um objeto obtido de uma subclasse sempre pode substituir um objeto de tipo superclasse o seja tudo o que podemos deduzir a respeito de uma superclasse também será válido para suas subclasses.

.

Exemplo Herança

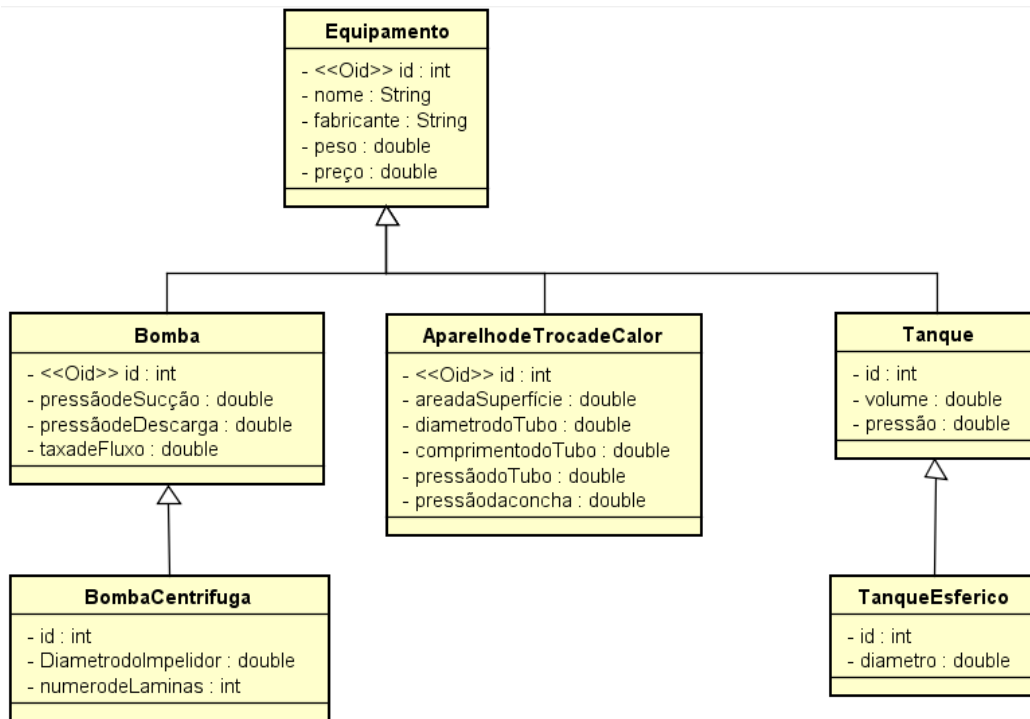
Modelar a seguinte situação: Cada peça de equipamento tem fabricante, peso e preço. As bombas têm, além disso, pressão de sucção e taxa de fluxo. Os tanques também têm volume e pressão. Queremos definir características do equipamento apenas uma vez e, depois, seria feita a adição dos detalhes para bombas, tanques e outros tipos de equipamentos.

Herança

Questionamento inicial para considerar aplicar herança.

1. Os conceitos têm uma estrutura comum?
 - Nome, fabricante, peso e preço.
2. Existe um conceito genérico que consiga generalizar os tipos diferentes de conceitos?
 - Equipamento: Bomba, tanques, Aparelho de Troca de Calor, etc.

Herança



Extensão:

- Pode adicionar atributos, mas não pode remover atributos.
- A herança é uma associação de classes e não de objetos . O seja só um objeto é criado, mas não um objeto para classe.
- Recomenda-se evitar herança parcial, preferencialmente aplicar a herança total.

Uso inadequado de Herança

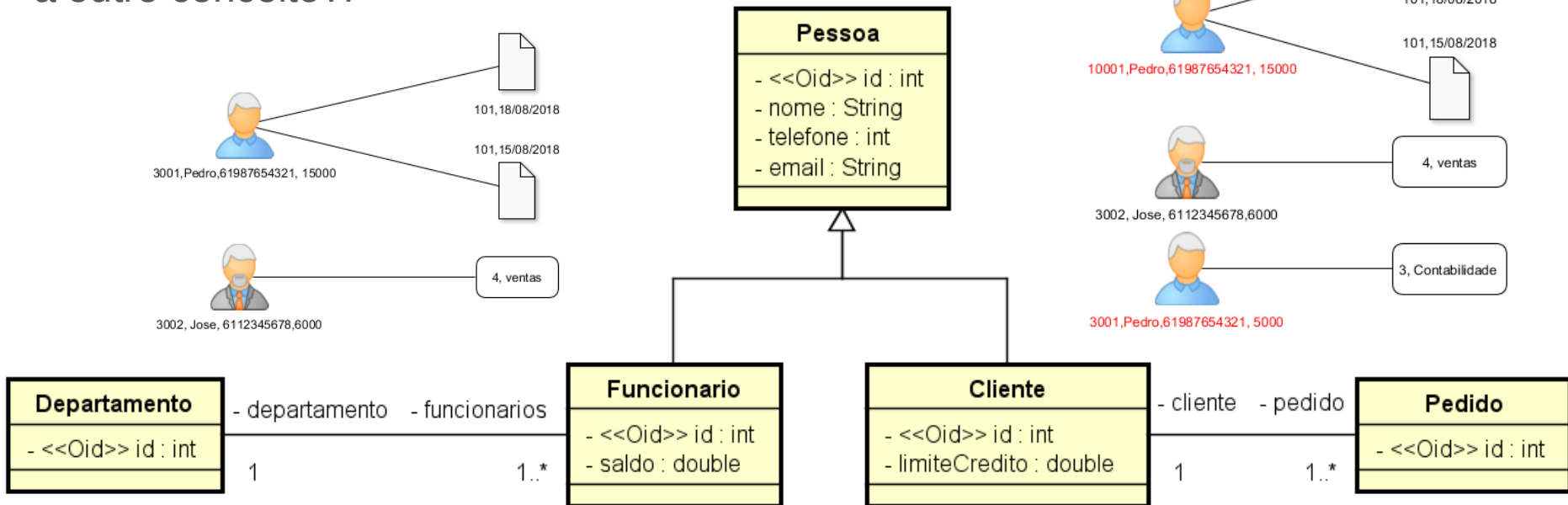
As seguintes questões podem indicar o uso errado de herança:

- Para o conceito subclasse fazer sentido, precisa de uma associação a outro conceito?.
- Uma Superclasse pode ser **conceito 1** e também **conceito 2** ?.

Uso inadequado de Herança

Para o conceito subclasse fazer sentido, precisa de uma associação a outro conceito?

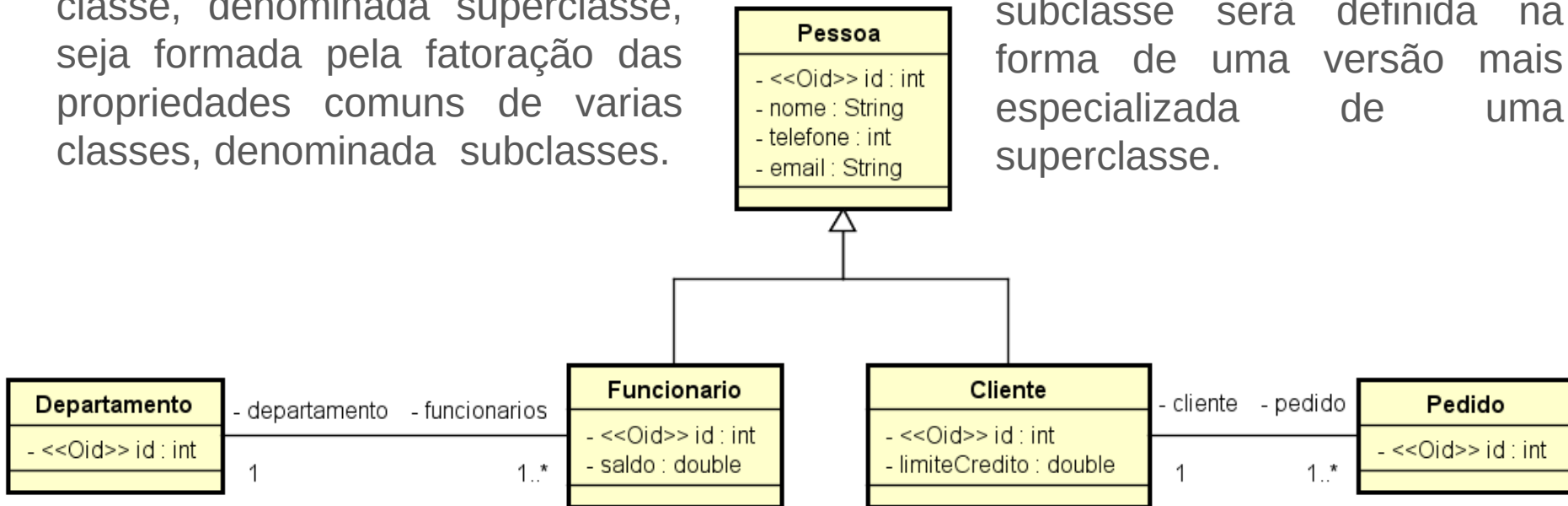
Uma Superclasse pode ser conceito_1 e também conceito_2 ?



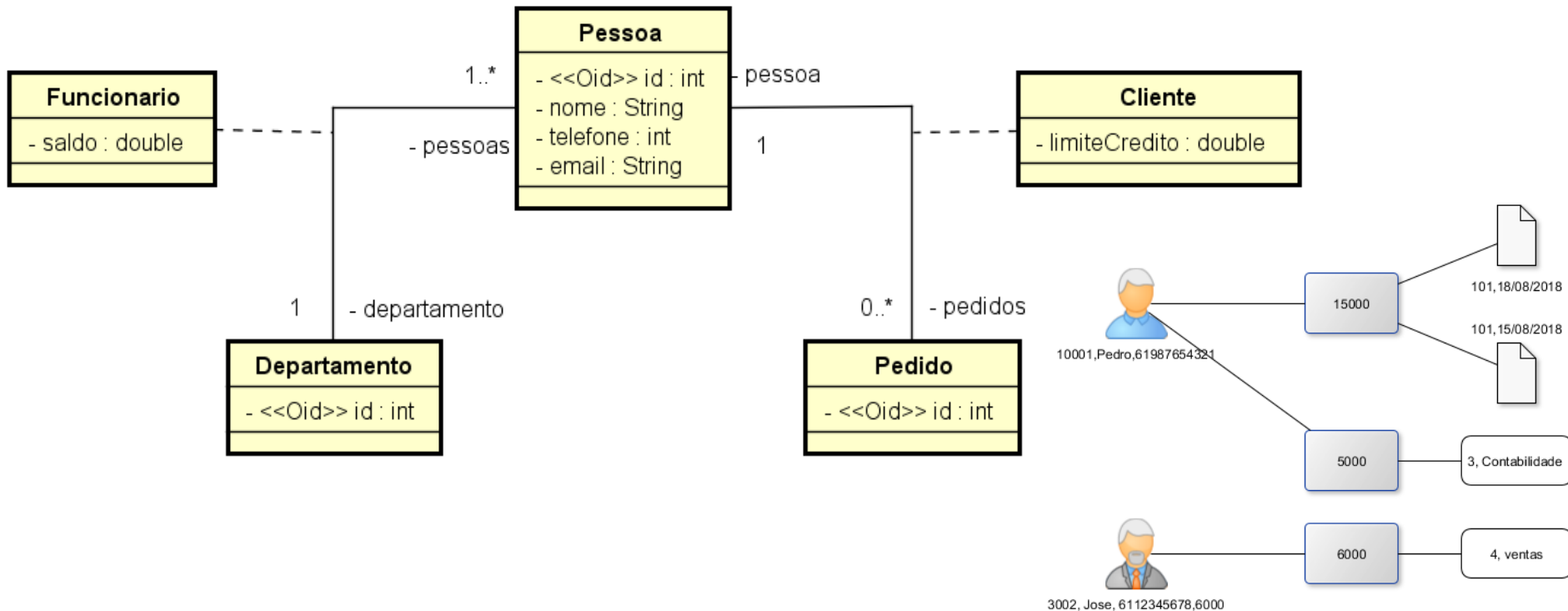
Generalização e especialização

A *generalização* permite que uma classe, denominada superclasse, seja formada pela faturação das propriedades comuns de varias classes, denominada subclasses.

Especialização representa o caso inverso, onde uma nova subclasse será definida na forma de uma versão mais especializada de uma superclasse.



Solução



Exercício 1 – Sistema de Controle Cinema

Um cinema pode ter muitas salas, sendo necessário, por tanto, registrar informações a respeito de cada uma, como sua capacidade, ou seja, o numero de assentos disponíveis.

O cinema apresenta muitos filmes. Um filme tem informações, titulo e duração. Assim, sempre que um filme for ser apresentado, deve-se registrá-lo também.

Um mesmo filme pode ser apresentado em diferentes salas e em horários diferentes. Cada apresentação em uma determinada sala e horário é chamada sessão. Um filme sendo apresentado em uma sessão tem um conjunto máximo de ingressos, determinado pela capacidade da sala.

Os clientes do cinema podem comprar ou não ingressos para assistir a uma sessão. O funcionário deve intermediar a compra do ingresso. Um ingresso deve conter informação como o tipo de ingresso (Meio ingresso ou ingresso inteiro). Além disso, um cliente só pode comprar ingressos para sessões ainda não encerradas.

Exercício 2 – Locadora

Sistema Locadora de jogos digitais e consolas:

Cada jogo e consola possui seu preço diário de locação, sendo que um mesmo jogo pode ter preços de locação diferentes para cada plataforma (Xbox, PS3, PS4, PC, etc.). O cliente (nome, identidade, cpf e-mail, telefone) especifica o jogo, plataforma e dias (pode alocar vários jogos de diferentes plataformas por vários dias). A data e hora da locação são armazenadas.

Exercício 3 – Locadora

Uma locadora de carros deseja fazer um sistema para armazenar as informações das locações que os clientes fazem. A locadora possui varias agencias (código agencia, e localidade). É necessário registrar tanto a data, hora e agência para cada alocação como para sua devolução e dados do cliente. A locação pode ser tanto diária (precisa de dias previstos para devolução) como por período (aplica porcentagem de desconto dado no valor da diária). A locadora armazena dados dos carros (modelo, placa, cor, ano e data de aquisição) e classifica em uma categoria para definir seu valor de diária de locação.

Bibliografia

- J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy y W. Lorensen, Modelagem e projetos baseados em objetos, vol. 8, Campus Rio de Janeiro, 1994.
- G. T. A. Guedes, UML 2-Uma abordagem prática, Novatec Editora, 2018.
- H. DEITEL y P. J. Deitel, C++ como programar. 5a edição, PRENTICE HALL BRASIL, 2006.
- D. Coleman, P. A. S. Bodoff, C. D. H. Gilchrist y F. H. P. Jeremaes, Desenvolvimento orientado a objetos: o método fusion, Campus, 1996.
- M. Weisfeld, The object-oriented thought process, Pearson Education, 2008.

Bibliografia

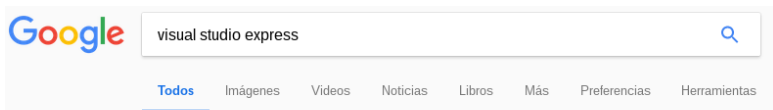
- J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy y W. Lorensen, Modelagem e projetos baseados em objetos, vol. 8, Campus Rio de Janeiro, 1994.
- G. T. A. Guedes, UML 2-Uma abordagem prática, Novatec Editora, 2018.
- H. DEITEL y P. J. Deitel, C++ como programar. 5a edição, PRENTICE HALL BRASIL, 2006.
- D. Coleman, P. A. S. Bodoff, C. D. H. Gilchrist y F. H. P. Jeremaes, Desenvolvimento orientado a objetos: o método fusion, Campus, 1996.
- M. Weisfeld, The object-oriented thought process, Pearson Education, 2008.

Instalação Astah Community

1. Pesquisar em google “Astah community”, conferir o URL_
<http://astah.net/editions/community>
2. Escolher para descarga e instalar “[free edition for students](#)”.

Instalação C # - Windows

1. Pesquisar na web “*Visual Studio Express*“, Baixe o “*Visual Studio Community*” para ter um IDE extensível e totalmente funcional.



Cerca de 118,000,000 resultados (0.61 segundos)

Visual Studio Express | Now Visual Studio Community

<https://visualstudio.microsoft.com/es/vs/express/> Traducir esta página

20 jul. 2018 - Try out Visual Studio Community for a fully-featured IDE; An updated alternative to Visual Studio Express. Download Visual Studio Community ...

MICROSOFT VISUAL STUDIO 2017 EXPRESS FOR WINDOWS ...

<https://visualstudio.microsoft.com/es/vs/older-downloads/> Traducir esta página

2 mar. 2018 - Microsoft Visual Studio - Windows Dev Center - Developer Network - TechNet - Microsoft developer program - Channel 9 - Office Dev Center ...

Download Older Visual Studio Software | Visual Studio - Visual Studio

<https://visualstudio.microsoft.com/es/vs/older-downloads/> Traducir esta página

20 jul. 2018 - Visual Studio Team Foundation Server Express 2015. Visual Studio Team ... Visual Studio Express 2015 for Windows Desktop. Web Installer.

Microsoft Visual Studio Express - Wikipedia, la enciclopedia libre

https://es.wikipedia.org/wiki/Microsoft_Visual_Studio_Express

Microsoft Visual Studio Express Edition es un programa de desarrollo en entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas ...

Características - Productos

Microsoft Visual Studio Express - Wikipedia

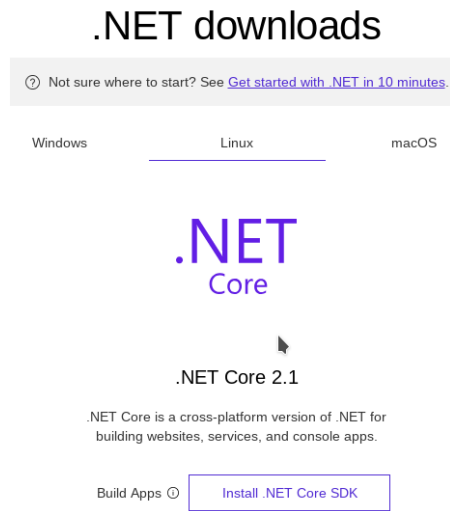
Visual Studio Express

Baixe o Visual Studio Community para ter um IDE extensível e totalmente funcional; uma alternativa atualizada do Visual Studio Express.

Baixar o Community 2017 

Instalação C # - Linux

1. Pesquisar em google “.net core SDK”Pesquisar em google “.net core SDK”,
conferir o URL <https://www.microsoft.com/net/download>.
2. Escolher para descargar o instalador que se encaixar no seu sistema, sem esquecer que seja o SDK



```
sudo apt-get install apt-transport-https  
sudo apt-get update  
sudo apt-get install dotnet-sdk-2.1
```

Instalação C # - Linux

3. Vamos a buscar em Google, “VS Code”, conferir o URL <https://code.visualstudio.com/> e descargar segundo nossa plataforma.

