Engenharia de Software

Diagrama de classes

Marta Martinho 2020

Organização da apresentação

- Definições de programação orientada a objetos - POO;
- Associações entre classes;
- Generalização e herança;
- Exercícios exemplo;

Objetos

- No desenvolvimento de software Orientado por Objetos (OO), o mundo real é visto como um conjunto de objetos que interagem entre si:

    ✓ os objetos computacionais são imagens de objetos reais.

- Exemplos de objetos do mundo real:

    ✓ o Sr. João Oliveira

    ✓ a aula de ES no dia 2020/11/04

- Exemplos de objetos computacionais:

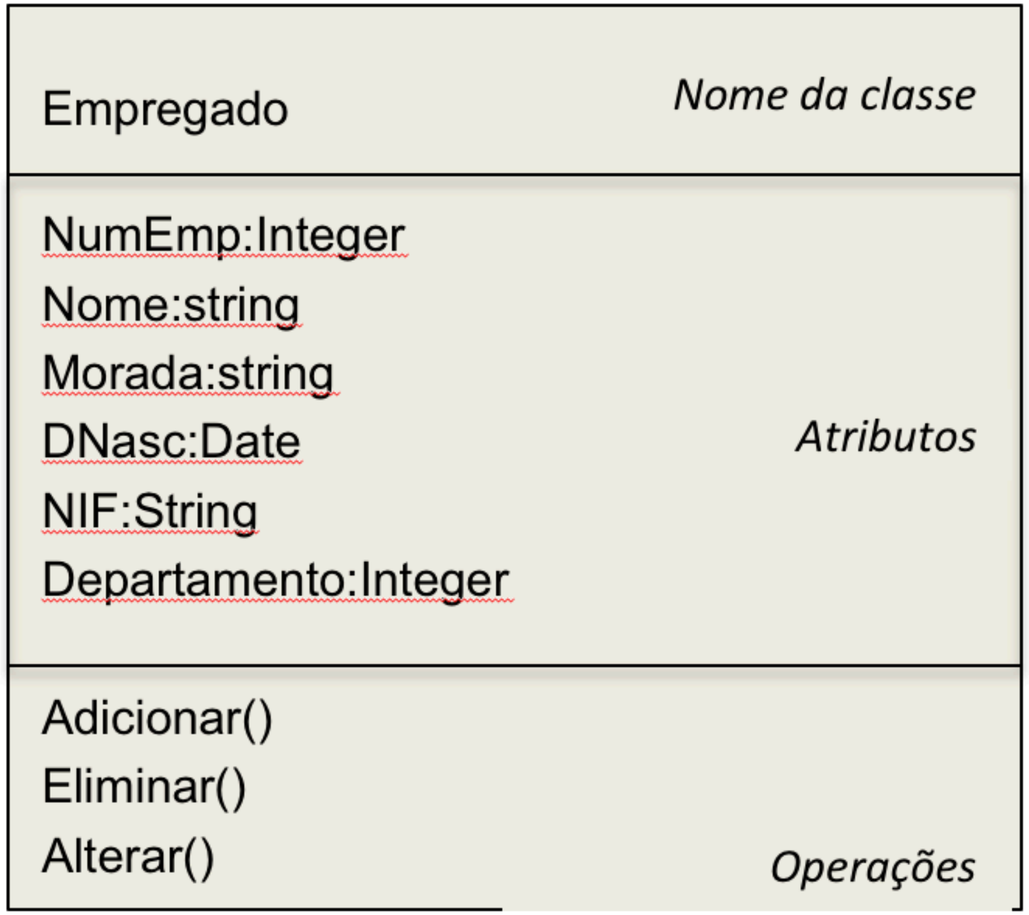
        ✓ o registo que descreve o Sr. João (imagem de objeto do mundo real - e.g. nome, idade, profissao)

Classe

- Classe é uma descrição de um conjunto de objetos que têm **os mesmos atributos,** operações, relacionamentos e semântica
    - ✓ um objeto de uma classe é uma instância da classe
- Uma classe pode representar:
    - ✓ Coisas concretas: Pessoa, Turma, Carro, Imóvel, Fatura, Livro
    - ✓ Papéis que coisas concretas assumem: Aluno, Professor, Piloto
    - ✓ Eventos: Curso, Aula, Acidente
    - ✓ Tipos de dados: Data, Intervalo de Tempo, Número Complexo, Vetor
- A abordagem orientada por objetos leva a arquiteturas mais estáveis

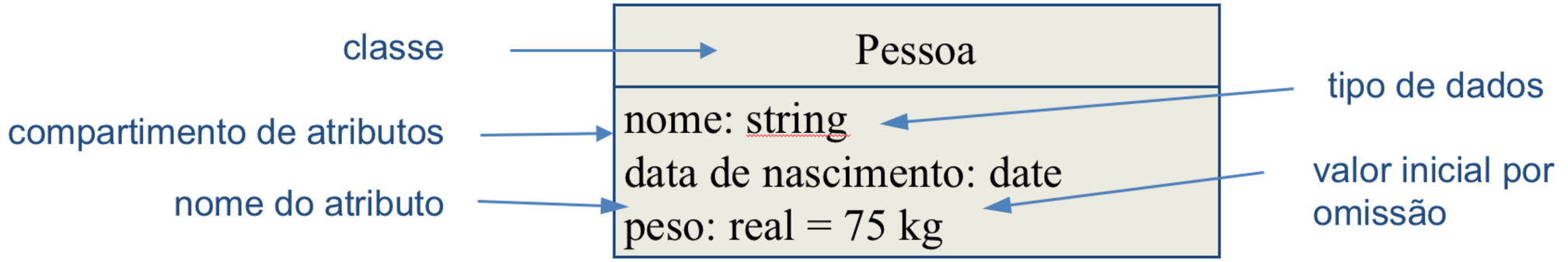
Classe

- Em UML, uma classe é representada por um retângulo com o nome da classe;
- Habitualmente escreve-se o nome da classe no singular:
  - nome de uma instância;
  - com a 1ª letra em maiúscula
- Os atributos são definidos ao nível da classe;
- Os valores do atributos são definidos ao nível do objeto dentro de um domínio
- Podem ser ainda adicionadas operações que se podem realizar na classe - métodos



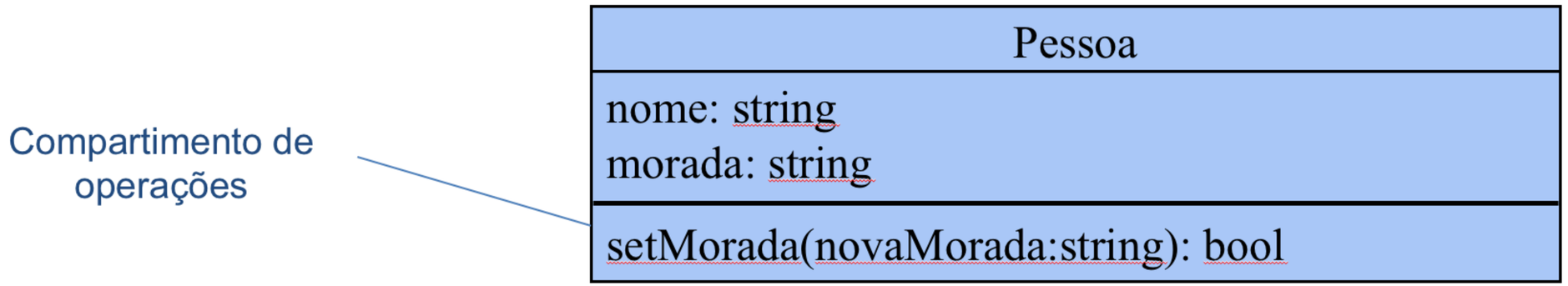
Atributos de instância

- O estado de um objeto é dado pelos valores dos atributos que partilha com a sua classe e pelas ligações que tem com outros objetos
- Os nomes dos tipos não estão pré-definidos em UML, podendo-se usar os da linguagem de implementação alvo
- Exemplo:
  - ✓ uma pessoa, a **classe**, tem os atributos nome, data de nascimento e peso
  - ✓ João, um dos **objetos - instância da classe -**, é uma pessoa com nome “João Silva”, data de nascimento “18/3/1973” e peso “68 Kg”



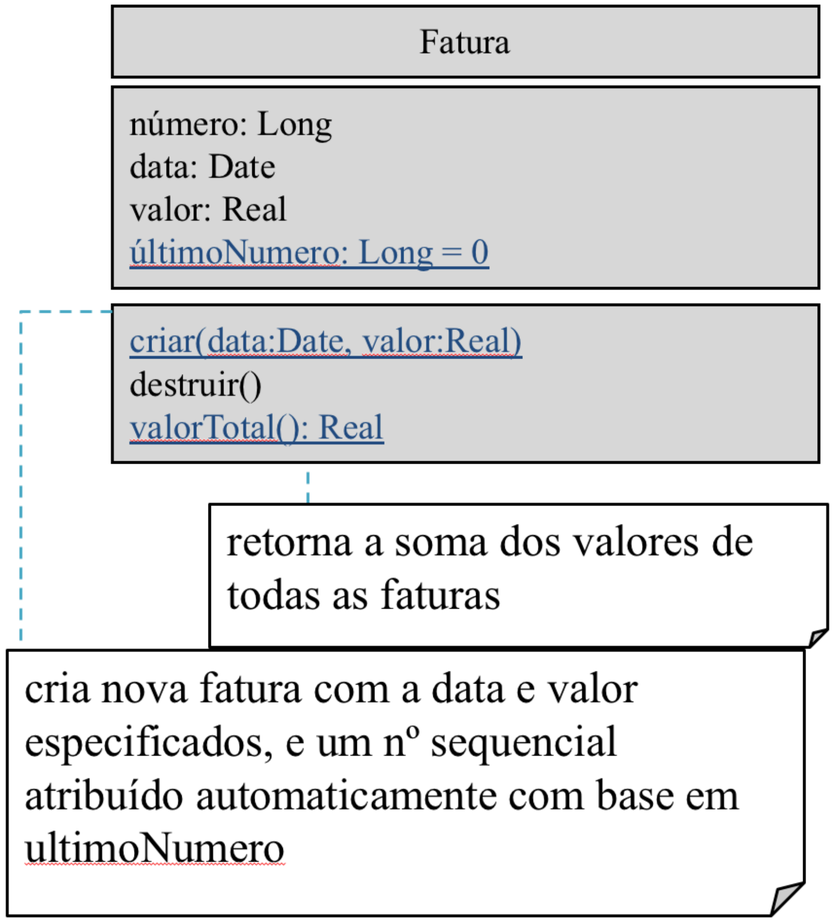
Operações de instância

- Comportamento invocável de objetos é modelado por operações
    - uma operação é algo que se pode pedir a um objeto para este realizar
- Objetos da mesma classe têm as mesmas operações
    - Operações são definidas ao nível da classe, enquanto que a invocação de uma operação é definida ao nível do objeto



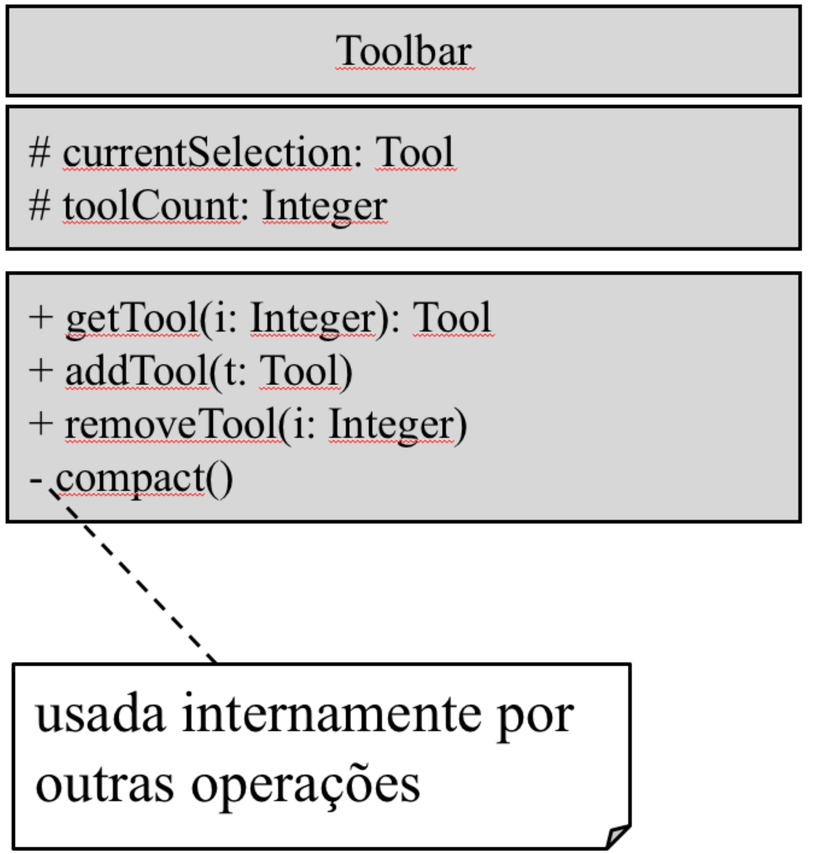
Atributos e operações estáticas

- Atributo estático
  - ✓ tem um único valor para todas as instâncias (objetos) da classe
  - ✓ valor está definido ao nível da classe e não ao nível das instâncias
- Operação estática
  - ✓ não é invocada para um objeto específico da classe
  - ✓ Notação: nome sublinhado
  - ✓ Correspondem a membros estáticos (*static*) em C++, C# e Java



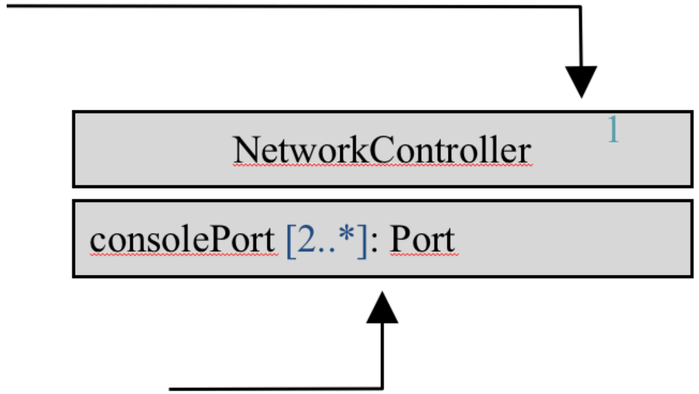
Visibilidade de atributos e operações

- Visibilidade:
  - ✓ + (public) : visível por todos
  - ✓ - (private) : visível só por operações da própria classe
  - ✓ # (protected): visível por operações da própria classe e subclasses desta
- Encapsulamento: esconder todos os detalhes de implementação que não interessam aos clientes da classe
  - ✓ permite alterar representação do estado sem afetar clientes
  - ✓ permite validar alterações de estado



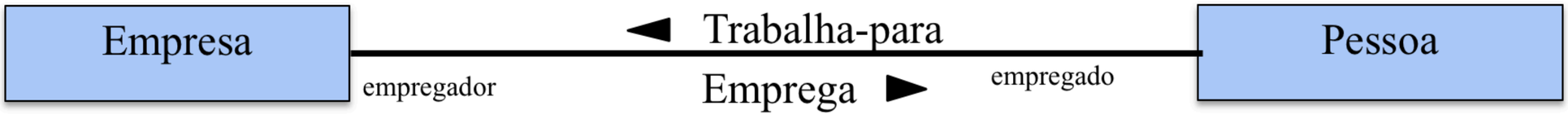
Multiplicidade de classes e atributos

- Multiplicidade de classe:
    - número de instâncias que podem existir
    - por omissão, é 0..*
- Multiplicidade de atributo: número de valores que o atributo pode tomar do tipo especificado
    - por omissão é 1



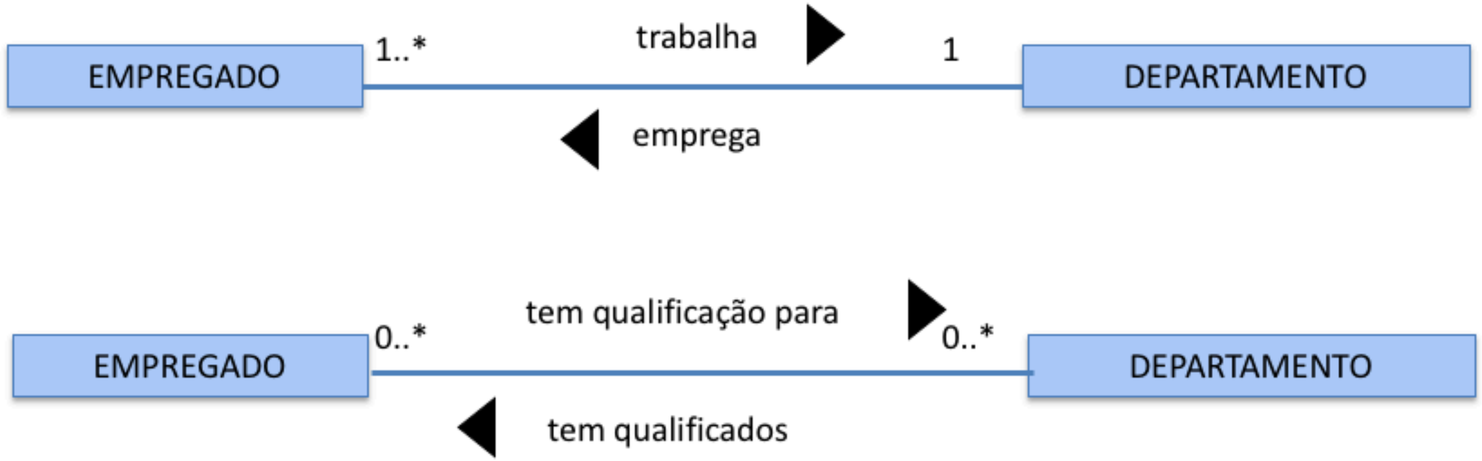
Nome da associação

- A indicação do nome é opcional
- O nome é indicado no meio da linha que une as classes participantes
- Pode-se indicar o sentido em que se lê o nome da associação



Multiplicidade (cardinalidade)

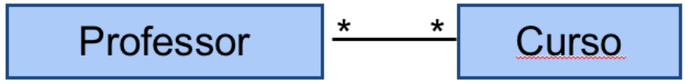
- A indicação do nome é opcional
- O nome é indicado no meio da linha que une as classes participantes
- Pode-se indicar o sentido em que se lê o nome da associação



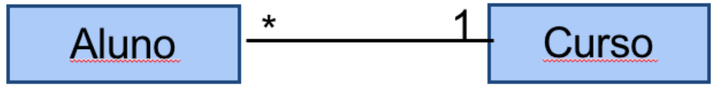
Multiplicidade (cardinalidade)

- Exemplos:
  - ✓ 1 exatamente um
  - ✓ 0..1 zero ou um (zero a 1)
  - ✓ 0..* zero ou mais
  - ✓ * zero ou mais
  - ✓ 1..* um ou mais
  - ✓ 1, 3..5 um ou três a 5

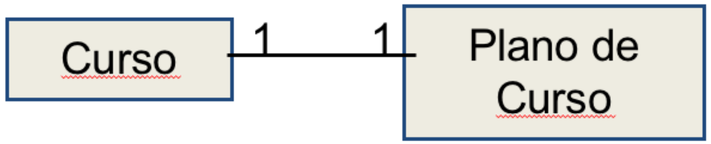
Muitos-para-Muitos



Muitos-para-1



1-para-1



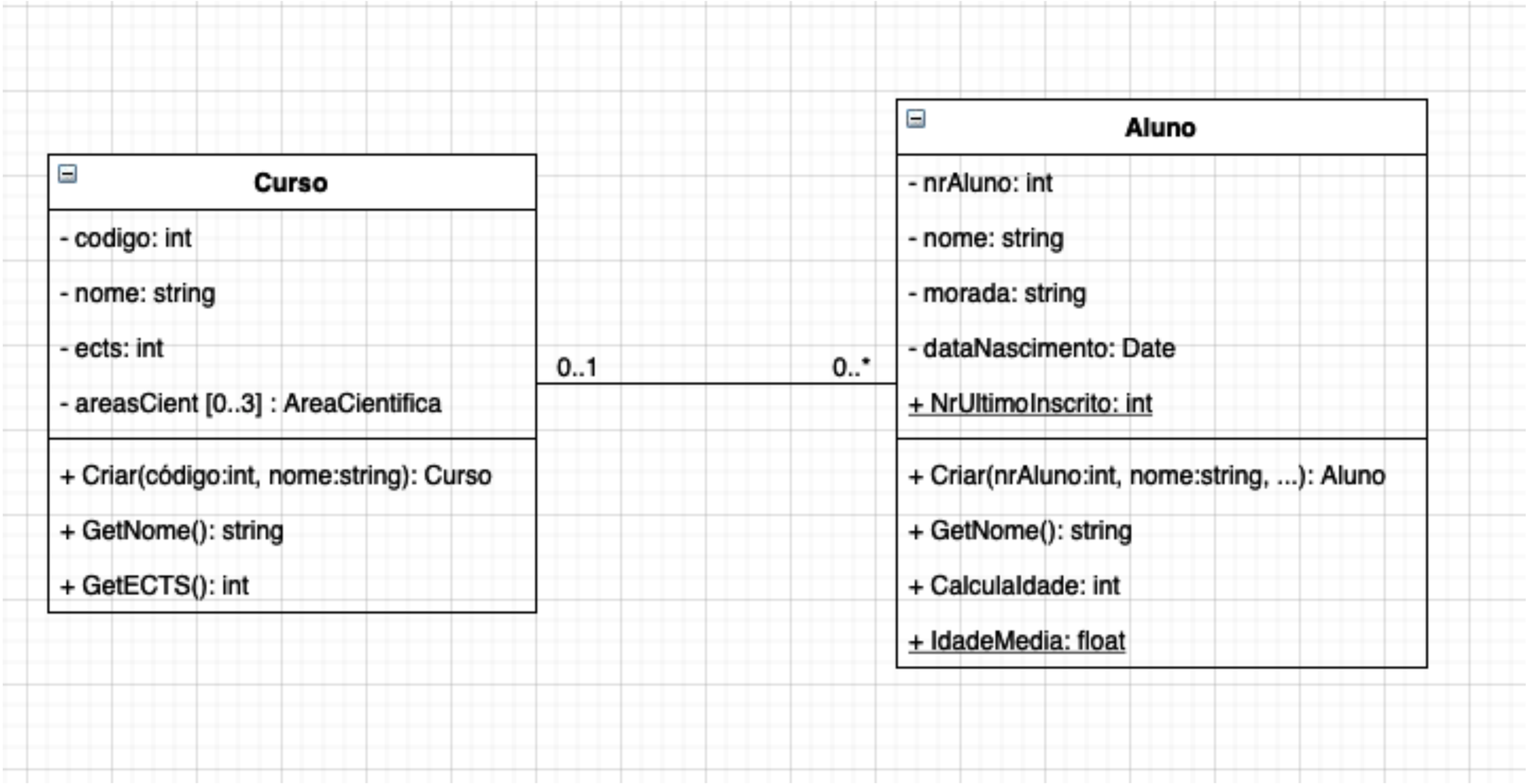
Exercício

Usando a notação apresentada, represente os seguintes objetos:

- Curso
  - ✓ Atributos: Código, Nome, ECTS, AreasCient (até 3)
  - ✓ Métodos: Criar, GetNome, GetECTS
- Aluno
  - ✓ Atributos: NrAluno, Nome, Morada, DataNasc, CartãoCidadão e NrUltInscrito (estático)
  - ✓ Métodos: Criar, GetNome, CalcIdade e IdadeMédia (estático)

Considere a visibilidade e a multiplicidade das classes e atributos.

Exercício

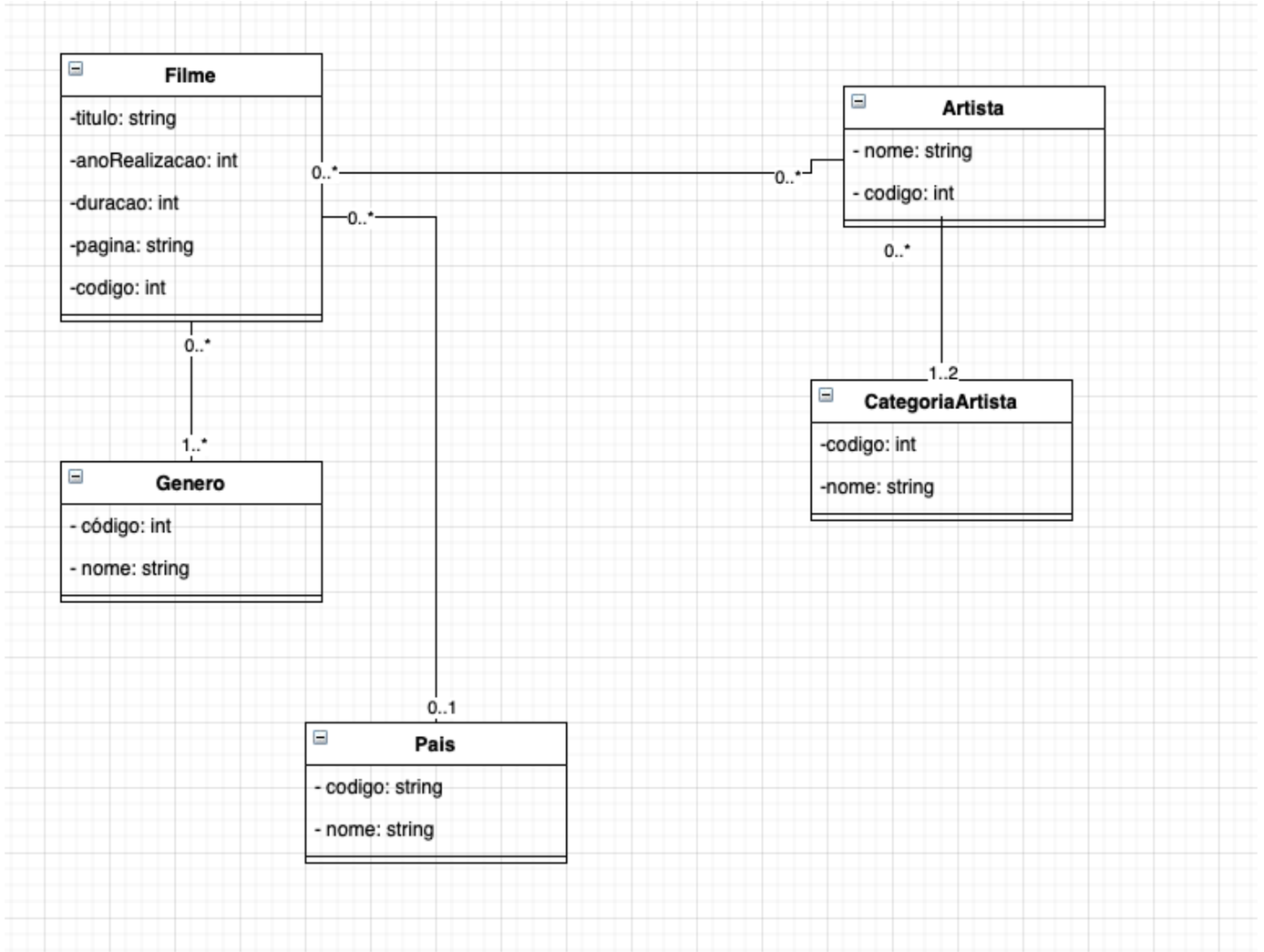

Curso
- codigo: int
- nome: string
- ects: int
- areasCient [0..3] : AreaCientifica
+ Criar(código:int, nome:string): Curso
+ GetNome(): string
+ GetECTS(): int
0..1
0..*
Aluno
- nrAluno: int
- nome: string
- morada: string
- dataNascimento: Date
+ NrUltimoInscrito: int
+ Criar(nrAluno:int, nome:string, ...): Aluno
+ GetNome(): string
+ CalculaIdade: int
+ IdadeMedia: float

Exercícios - Filmes

Um cinéfilo, detentor de uma coleção significativa de filmes, pretende uma aplicação que lhe permita armazenar e consultar os seus filmes. Pretende que os filmes possam ser consultados por título, género, país de origem, ano de realização, realizador ou atores intervenientes. Para além da informação referida, é também necessário saber a duração de cada filme e, caso exista, o endereço da página web dos realizadores e atores.

Desenhe o diagrama de classes.

Exercícios - Filmes



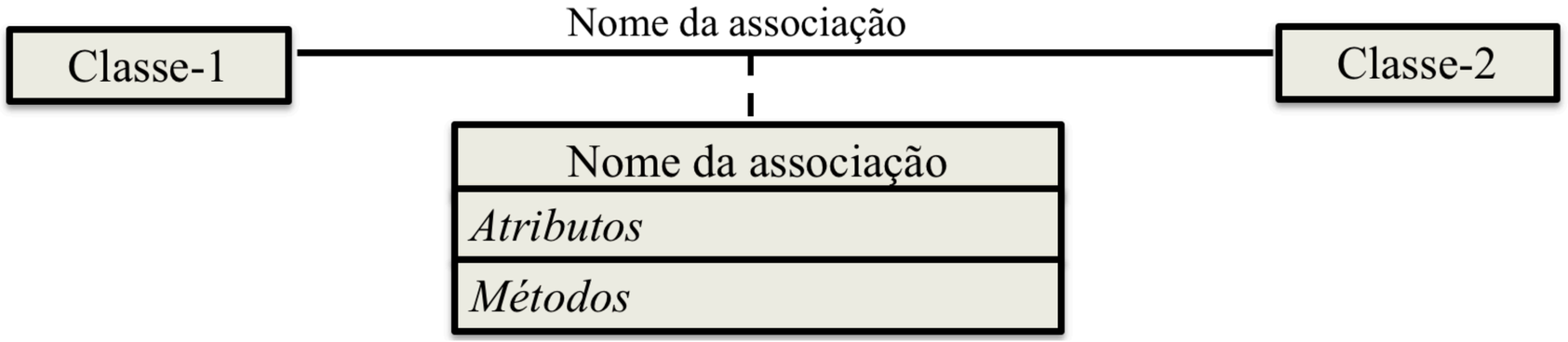
Classe-Associação

- Reúne as propriedades de associação e classe

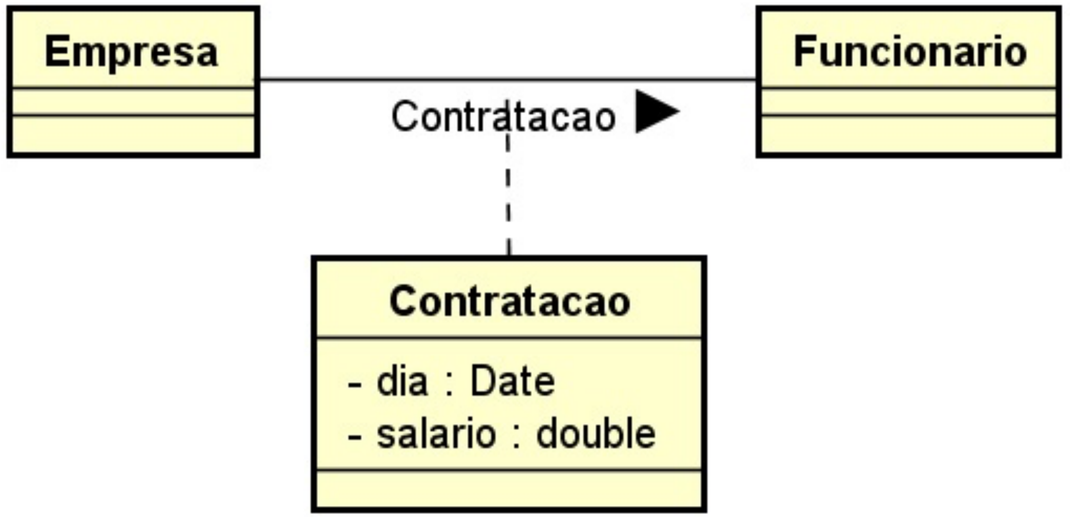
  ✓ o nome pode ser colocado num sítio ou noutro, conforme interessa realçar a natureza de associação ou de classe

  ✓ nome também pode ser colocado nos dois sítios

- Não é possível repetir combinações de objectos das classes participantes na associação



Classe-Associação



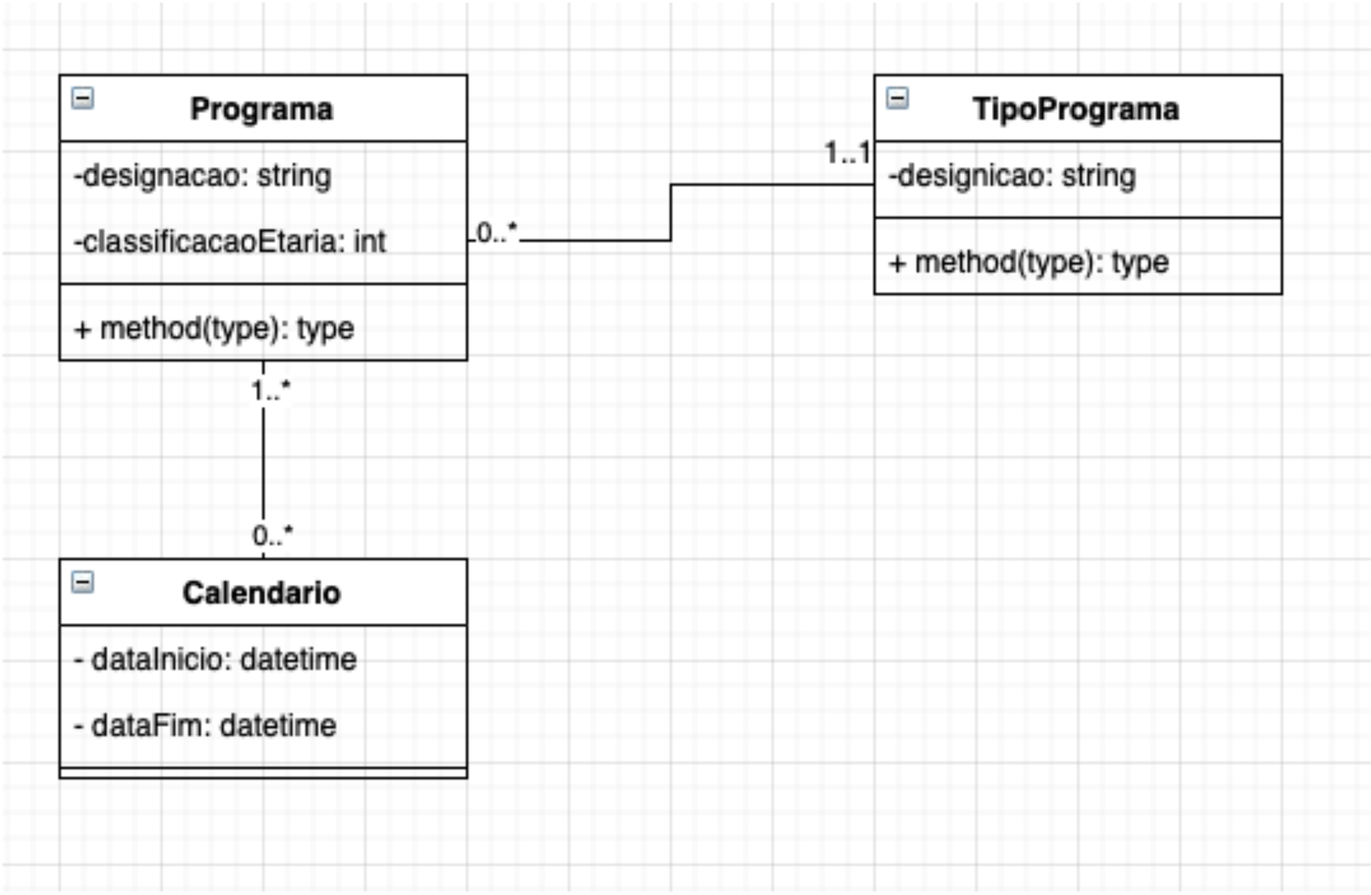
Exercícios - Programação de TV

Uma estação de televisão pretende um sistema de informação simples que a auxilie a armazenar e divulgar a sua programação diária. Quem consultar a programação, por exemplo através de um *browser*, deverá poder visualizar, para cada dia, a sequência de programas, com a indicação da hora de início e da duração de cada programa.

Também se pretende que o espectador possa visualizar o tipo de programa (Notícia, Filme, etc.) e a correspondente classificação etária.

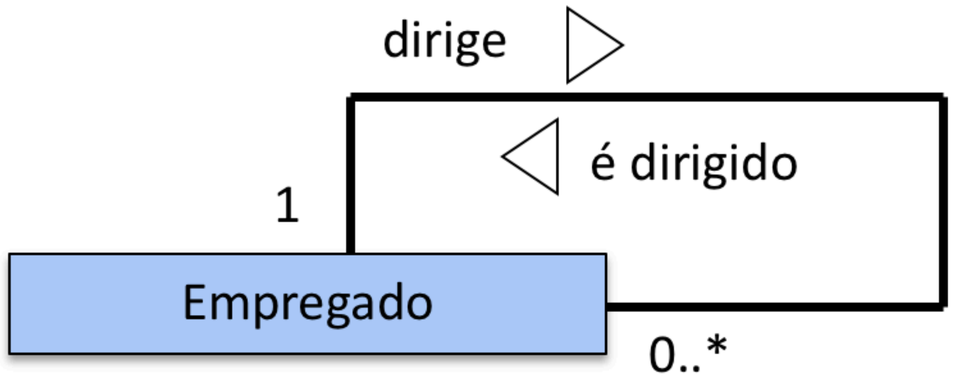
Desenhe o diagrama de classes.

Exercícios - Programação de TV



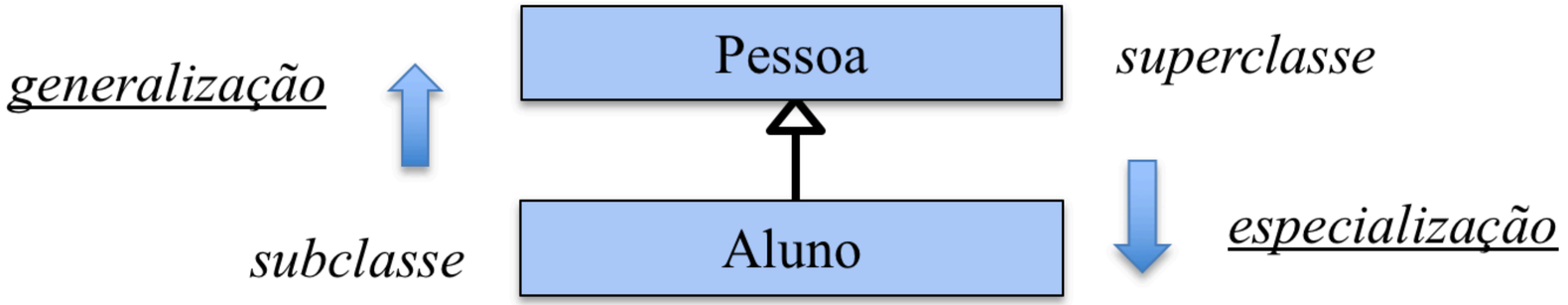
Associação reflexiva

- Pode-se associar uma classe com ela própria
- Normalmente, tem papéis diferentes em cada lado da associação

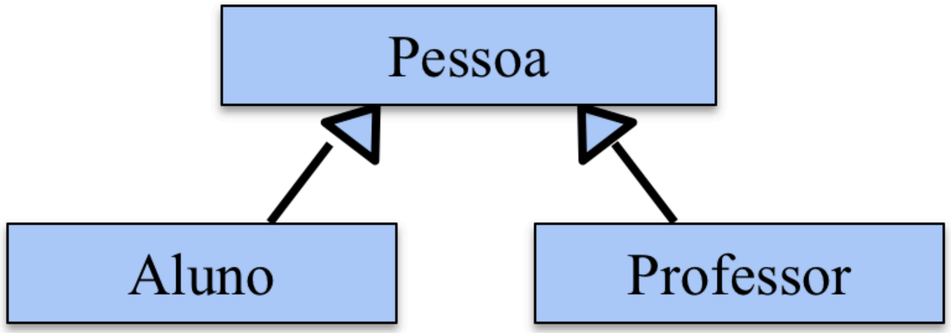


Generalização ou Herança

- Classes podem ser organizadas numa hierarquia onde uma classe (superclasse) é uma generalização de uma ou mais classes (subclasses)
- Uma **subclasse herda os atributos e operações da superclasse** e pode adicionar novos métodos
- Generalização em UML é implementada como herança nas linguagens OO
- O desenho da hierarquia de classes é um processo que se torna complexo quando se deseja evitar duplicação nos diferentes ramos



Generalização ou Herança

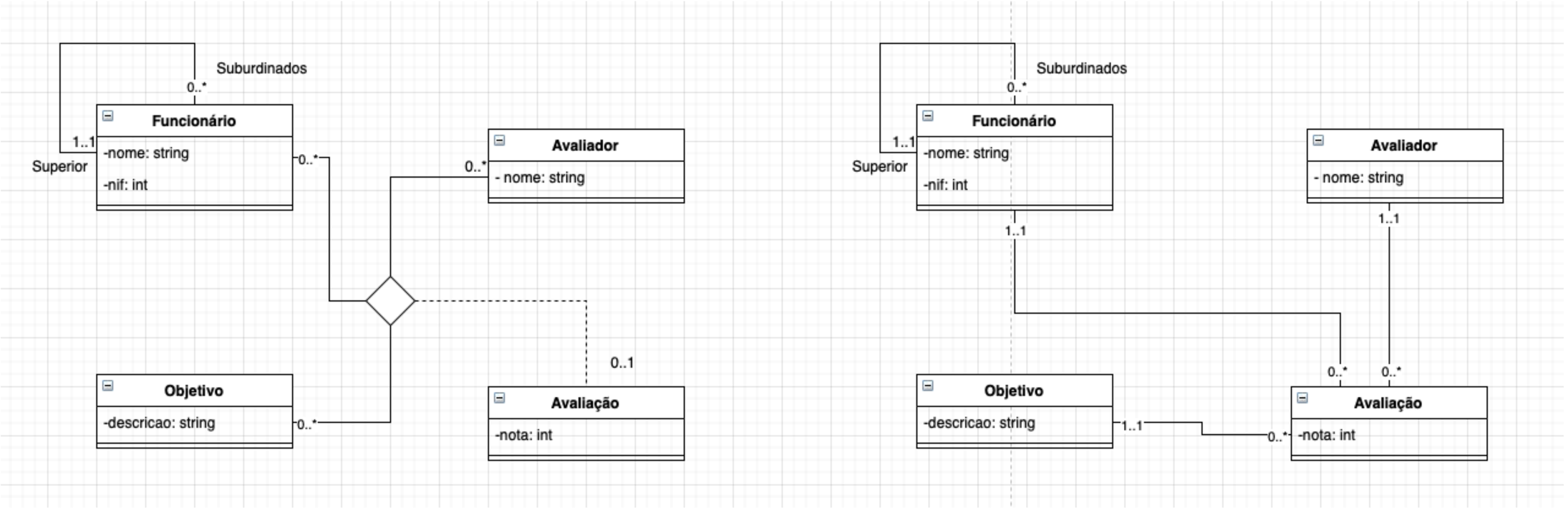


Exercícios - Avaliação de funcionários

Uma organização pretende uma aplicação de recursos humanos que armazene os resultados das avaliações periódicas que são efetuadas aos seus funcionários. Cada funcionário é avaliado por vários avaliadores, sendo que cada avaliador atribui uma nota para cada objetivo previamente atribuído ao funcionário avaliado. Sobre cada funcionário, para além do seu nome e número de contribuinte, é necessário saber qual o seu superior hierárquico. O avaliador é externo à organização.

Desenhe o diagrama de classes.

Exercícios - Avaliação de funcionários



Podem aplicar herança entre Funcionário e Avaliador

Tipos de relacionamentos entre classes

- A atribuição do tipo de relacionamento entre classes depende das regras de negócio do sistema.

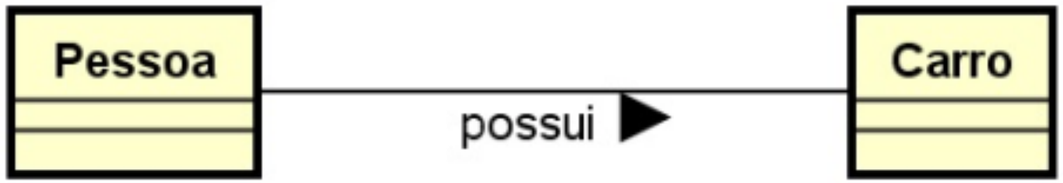
  ✓ Simples

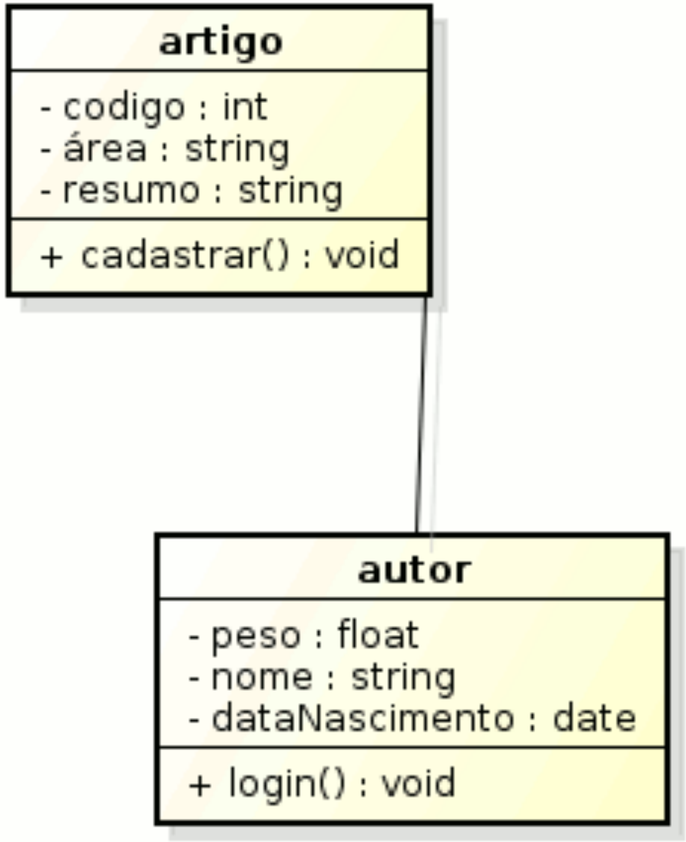
  ✓ Agregação

  ✓ Composição

Relacionamento simples

- Quando não existe uma relação “Pai e Filho” estamos perante uma associação simples, i.e., as classes não compartilham atributos entre si;
- No exemplo verifica-se que classe autor não compartilha atributos da classe artigo e vice-versa

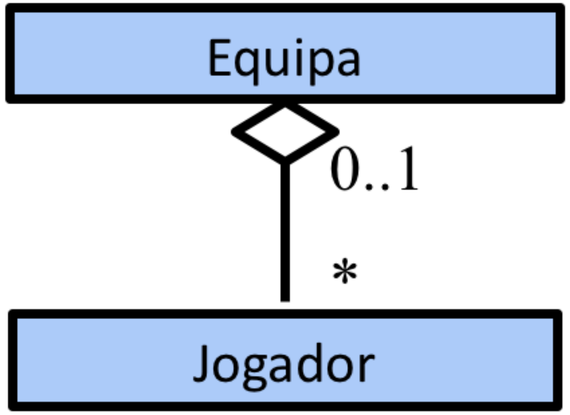




Relacionamento de agregação

- É uma associação com o significado “contém ...”, “é constituído por ...” ou “faz parte de ...”
    - Relação de inclusão entre as instâncias das classes
    - Hierarquias de objetos
- Representa um vinculo fraco entre duas classes, i.e., a classe filha continua a fazer sentido existir mesmo se a classe pai deixar de existir.

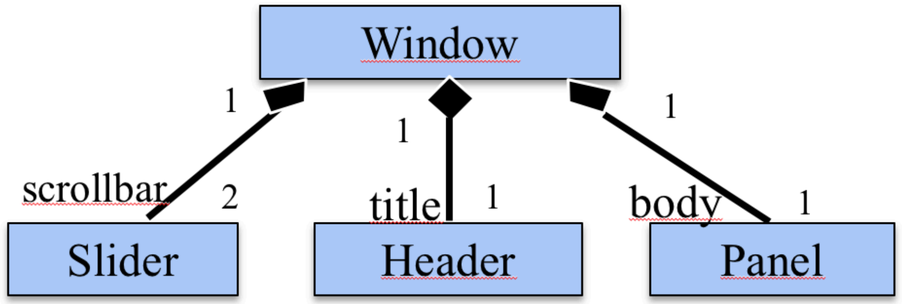
- Uma equipa contém *0 ou mais* jogadores
- Um jogador faz parte de *uma* equipa (num dado momento), mas também pode estar desempregado



Relacionamento de composição

- Forma mais forte de agregação aplicável quando:
  - Existe um forte grau de pertença das partes ao todo
  - Cada parte só pode fazer parte de um todo (i.e., a multiplicidade do lado do todo não excede 1)
  - As partes nascem e morrem dentro de um todo
  - A eliminação do todo propaga-se para as partes, em cascata
- Representa um vínculo forte entre duas classes, i.e., a classe filha só faz sentido existir se a classe pai também existir. Se a classe pai for apagada, automaticamente a filha deixará de existir.

- Notação: losango cheio (◆) ou notação baseada em caixas
- E.g., uma empresa é constituída por departamentos



Identificar tipos de relações entre classes

- Se não existe nenhuma dependência entre as classes
    - Então a relação é uma associação simples
- Senão.
- Se apagar a classe pai a classe filho deixa de fazer sentido existir
    - Então é uma relação de composição
- Senão
    - A relação é de agregação

Exercício - Edição colaborativa de documentos

Numa dada organização é frequente os seus funcionários colaborarem para a elaboração e edição de documentos. A organização pretende um sistema que armazene a estrutura dos documentos, com a indicação de quem contribui para a sua elaboração. Pretende-se que um documento seja decomposto em secções, que, por sua vez , podem sucessivamente ser decompostas em subsecções. Para cada secção ou subsecção é necessário registar qual o seu autor. Através da aplicação deverá também ser possível aos autores adicionarem comentários às várias secções ou subsecções. Aos autores dos documentos pode ser atribuído um grupo de forma a poderem colaborar em documentos de acordo com a sua competência.

Sobre os autores é suficiente conhecer o nome e endereço de correio eletrónico, e sobre as secções é necessário armazenar o seu título e o seu posicionamento dentro do documento.

Desenhe o respetivo diagrama de classes.

Exercício - Biblioteca

Considere o caso de uma biblioteca, na qual sócios podem requisitar publicações para ler em casa. Desenvolva o diagrama de classes respetivo de forma incremental, incluindo restrições de integridade.

Uma publicação pode ter vários exemplares, pertencer a uma editora e ser do tipo livro ou do tipo revista.

Um livro pode ter um ou mais autores.

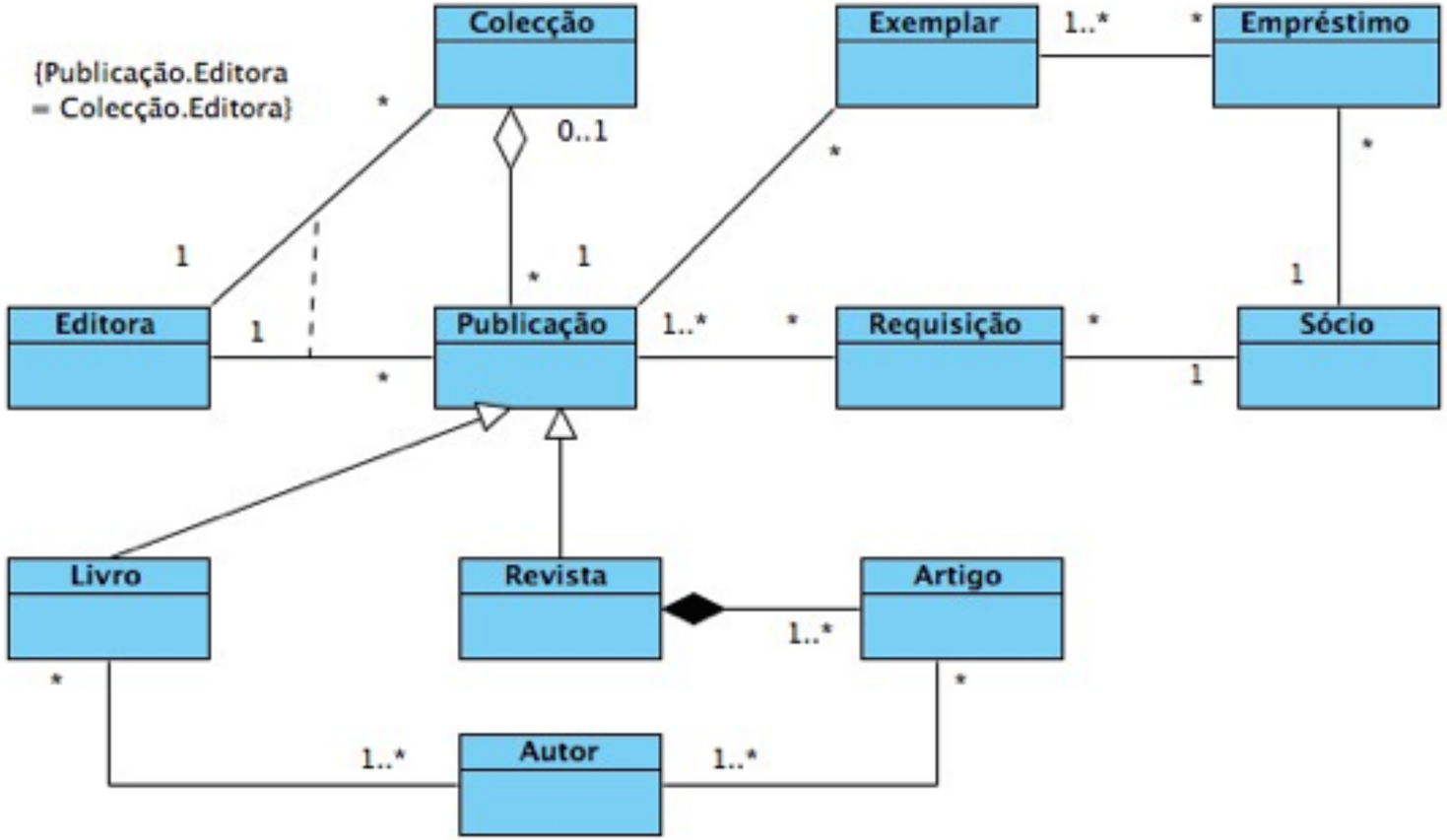
Uma revista contém artigos, sendo os autores definidos artigo a artigo.

As publicações podem ser agrupadas em coleções, sendo, nesse caso, definida a editora ao nível da coleção.

O sócio pode realizar a requisição de várias publicações. A requisição de uma publicação é concretizada através do empréstimo de um exemplar.

Desenhe o respetivo diagrama de classes.

Exercício - Biblioteca



Exercício - Agência de modelos

Uma agência de modelos pretende implementar um sistema de informação para gerir toda a informação dos seus colaboradores e dos trabalhos por eles efetuados. Os colaboradores dividem-se em modelos, fotógrafos e agentes.

Dos colaboradores, a aGência pretende saber o nome, a morada, o contacto e o número de identificação fiscal (NIF). Dos modelos, é necessário ainda saber o género, nacionalidade, altura e um conjunto três medidas. Os modelos podem ser representados por um dos agentes, mas também podem ter uma atividade independente. Os agentes são funcionários da agência cuja função é acompanhar o trabalho dos modelos que representam e zelar pelos seus interesses. É importante saber o ano em que cada agente iniciou a sua atividade na agência. Por forma a manter histórico sobre a atividade de cada agente, é necessário guardar as datas em que começou e terminou o seu trabalho com cada um dos modelos que representou desde que iniciou a sua atividade na agência. É importante saber o motivo pelo qual deixou de representar um determinado modelo.

Desenhe o respetivo diagrama de classes.

Exercício - Gestão de condomínios

Uma administração de um condomínio pretende um sistema de informação que a auxilie a controlar as contas correntes dos seus condóminos. A aplicação deverá permitir saber, para cada condómino (andar), que valores foram pagos (mensalidades ordinárias e pagamentos extraordinários) e que valores ainda estão em dívida.

O valor que cada condómino paga mensalmente (mensalidade ordinária) depende da área da sua casa, daí que não exista um valor igual para todas as casas. Esses valores podem sofrer alterações de ano para ano e, ocasionalmente, alguns condóminos têm dívidas referentes a anos anteriores.

As despesas extraordinárias (por exemplo, as despesas decorrentes de obras no prédio) são imputadas a cada condómino, normalmente em proporção da área da sua casa. É necessário registar quando cada condómino terá que pagar, caso contrário não será possível saber se o pagamento foi ou não efetuado. As despesas correntes do prédio não são pagas diretamente pelos condóminos mas sim pela administração..

Desenhe o respetivo diagrama de classes.

Exercício - Estatísticas de futebol

Uma Um jornal desportivo pretende um sistema de informação que lhe permita armazenar e disponibilizar online informação sobre um campeonato de futebol nacional. Para além da informação habitual sobre os jogos das várias jornadas (que equipas se confrontaram e em que data e qual o resultado dos jogos), o jornal pretende igualmente disponibilizar informação sobre estatísticas do jogo, nomeadamente que jogadores jogaram, qual foi considerado o melhor jogador e, para cada jogador, quantos golos marcou, cartões que recebeu e quanto tempo jogou.

Dado que a meio de um campeonato os jogadores podem mudar de clube, é importante poder saber, num determinado momento, quais os jogadores que compõem uma equipa.

Desenhe o respetivo diagrama de classes.

Engenharia de Software

Diagrama de classes

Marta Martinho 2020