

---

---

# **Slides 08 - Diagramas de objetos e classes UML**

---

---

— Sebastião Emidio Alves Filho —

---

---

# Linguagem de Modelagem Unificada

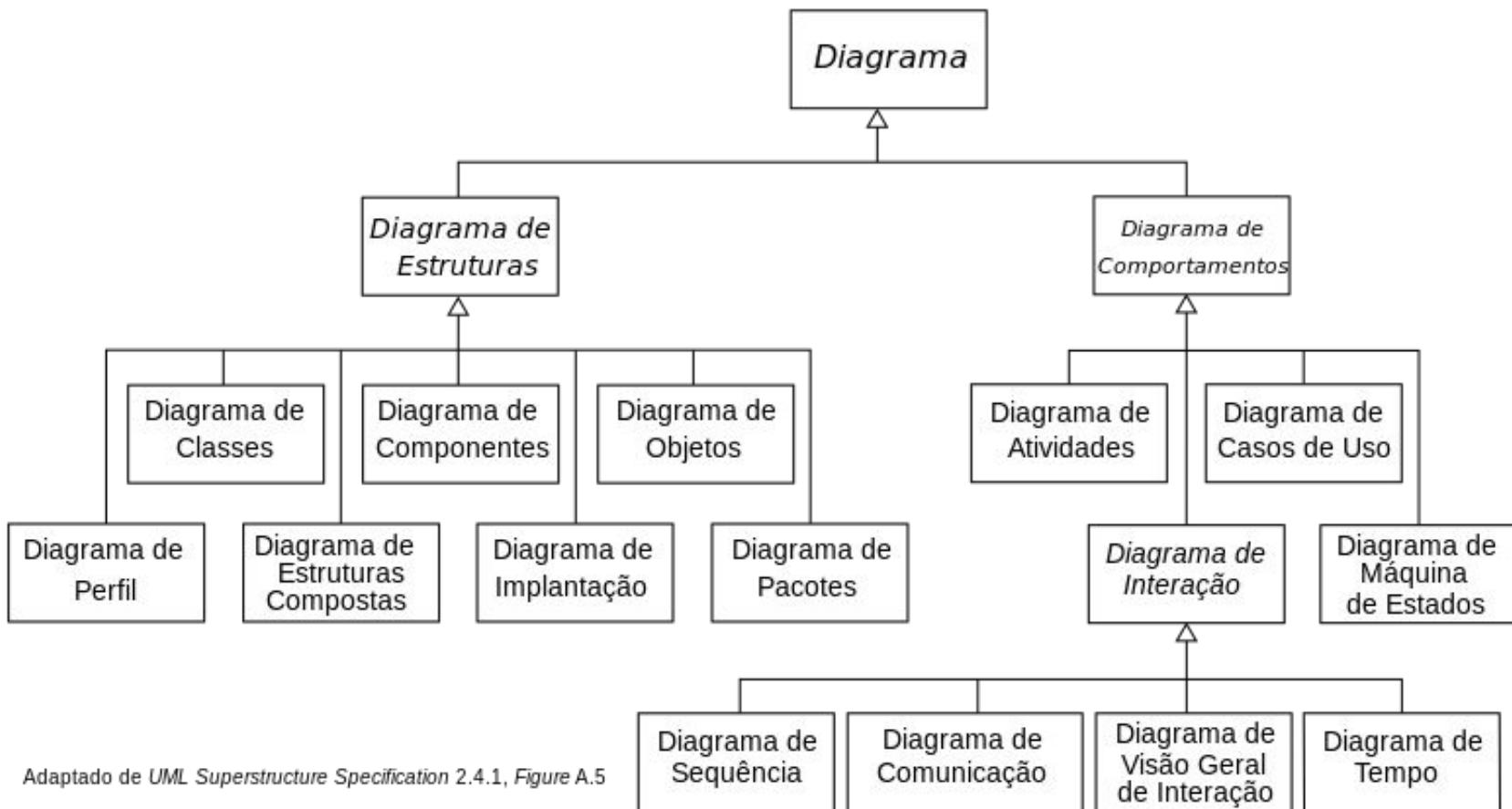
UML é uma linguagem gráfica para modelagem, isto é, para descrever um sistema de forma abstrata

Ajudam a reduzir a verbosidade, ambiguidade e confusão

UML é uma linguagem de modelagem:

- É uma linguagem formal
- É concisa
- É compreensível
- É escalável
- É construída a partir de lições aprendidas
- É um padrão **de fato**

# Os diagramas da UML



Adaptado de UML Superstructure Specification 2.4.1, Figure A.5

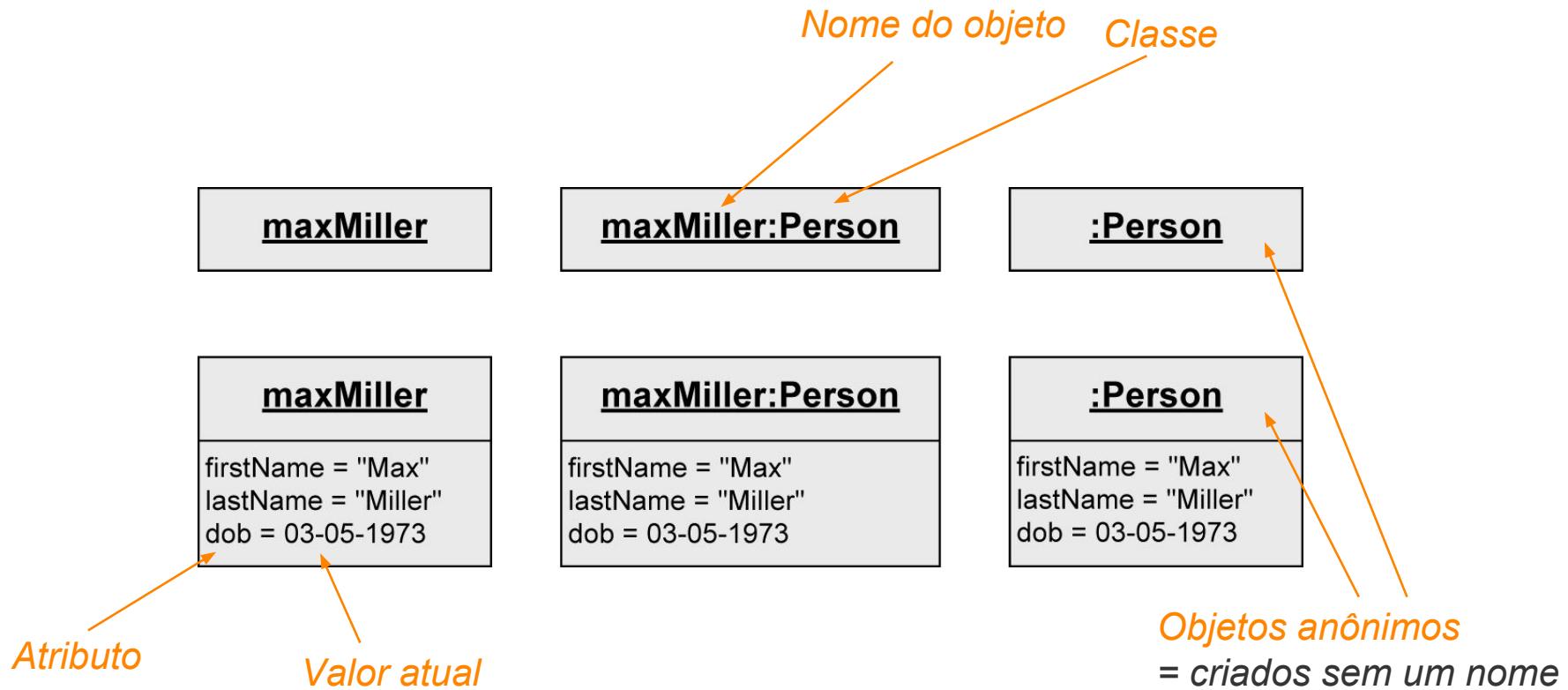
# Os diagramas da UML

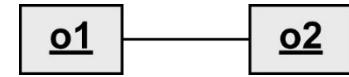
Organizados em 2 grupos:

- **Estruturais:** utilizados para visualizar, especificar, construir e documentar os aspectos estáticos de um sistema.
  - Diagrama de classes, Diagrama de objetos, Diagrama de componentes, Diagrama de instalação ou de implantação, Diagrama de pacotes, Diagrama de estrutura composta e Diagrama de perfil
- **Comportamentais:** Utilizado para visualizar, especificar, construir e documentar aspectos dinâmicos de um devido sistema.
  - Diagrama de caso de uso, Diagrama de transição de estados ou Máquina de estados, Diagrama de atividade e os Diagramas de interação: Diagrama de sequência Diagrama Visão Geral de Interação ou de interação, Diagrama de colaboração ou comunicação e Diagrama de tempo ou temporal
  -

# Objetos

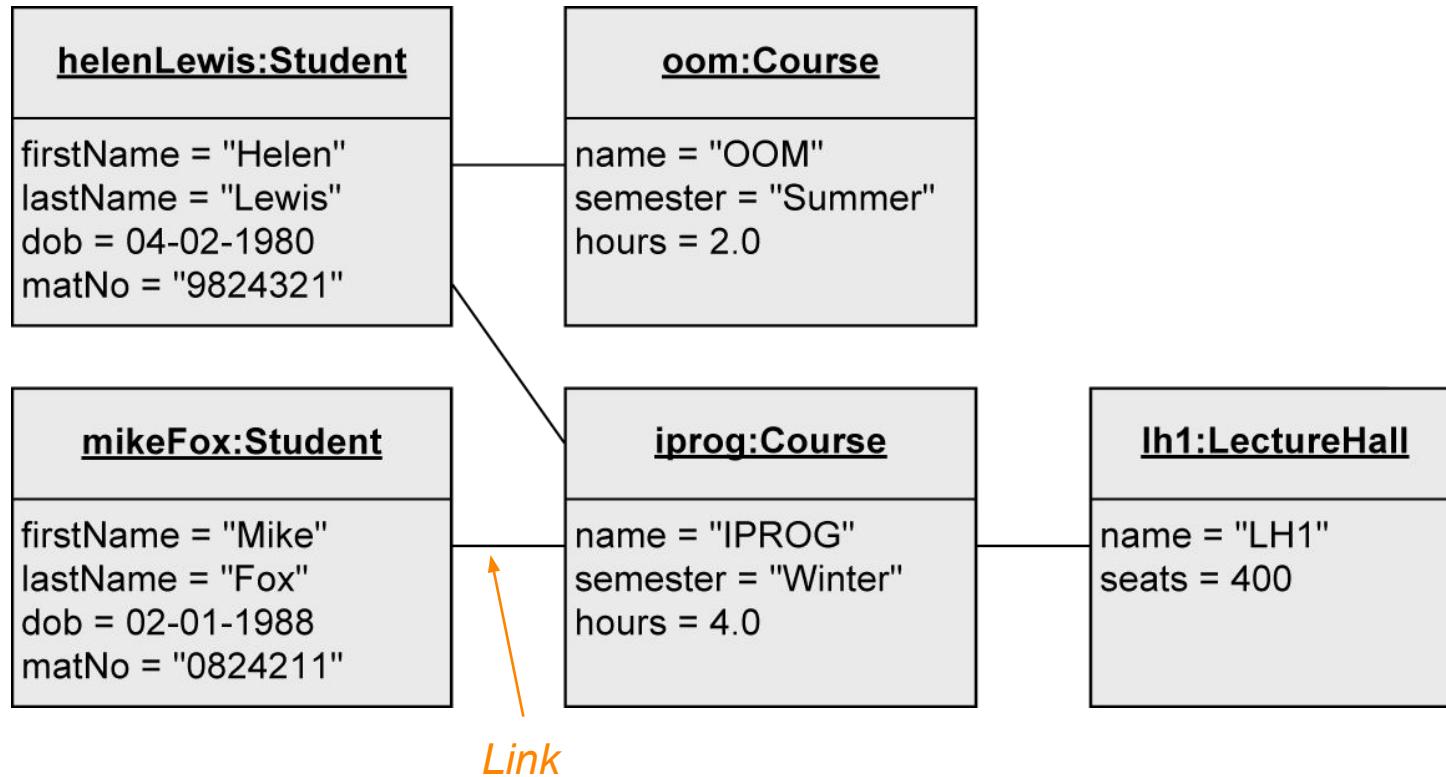
- Componentes individuais de um sistema

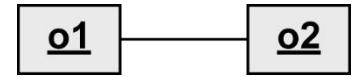




## Diagrama de objetos

- Objetos de um sistema e seus relacionamentos (links)
- Snapshot dos objetos em um momento específico





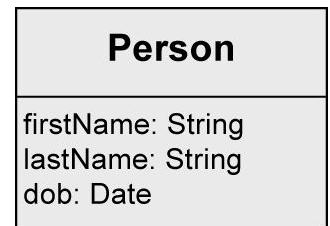
## Diagrama de objetos - Exemplo

- Espectador joao=new Espectador (“João“, [joao@exemplo.com](mailto:joao@exemplo.com));
- Assinante maria=new Assinante(“Maria“, [maria@exemplo.com](mailto:maria@exemplo.com));
  
- Campeonato brasileirao=new Campeonato(“Brasileiro-Serie A“, 20,TipoCampeonato.PONTOS\_CORRIDOS);
  
- maria.criaLiga (new LigaPrivada (“Amigos da UERN“, brasileirao));
  
- maria.getLiga().add(joao);
  
- Palpite chute1=new Palpite(“Corinthians“, 3, “Palmeiras“, 0);
- maria.cadastraPalpite(chute1);
- joao.cadastraPalpite(new Palpite(“Chapecoense“,0,“Flamengo“, 1));

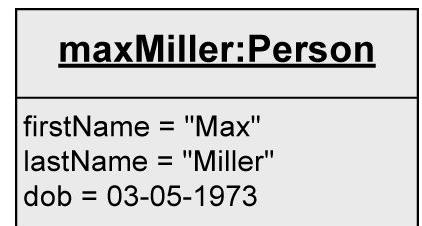
## Do objeto para a classe

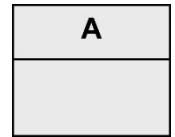
- Indivíduos do sistema tem características e comportamentos idênticos
- Uma classe é uma um modelo para construção de um conjunto de objetos de um sistema
- Atributos: características estruturais de uma classe
  - Valores diferentes para cada instância
- Operações: comportamento da classe
  - Idêntica para todos os objetos da classe  
→ Não vista no diagrama de objetos

*Classe*



*Objeto da classe*



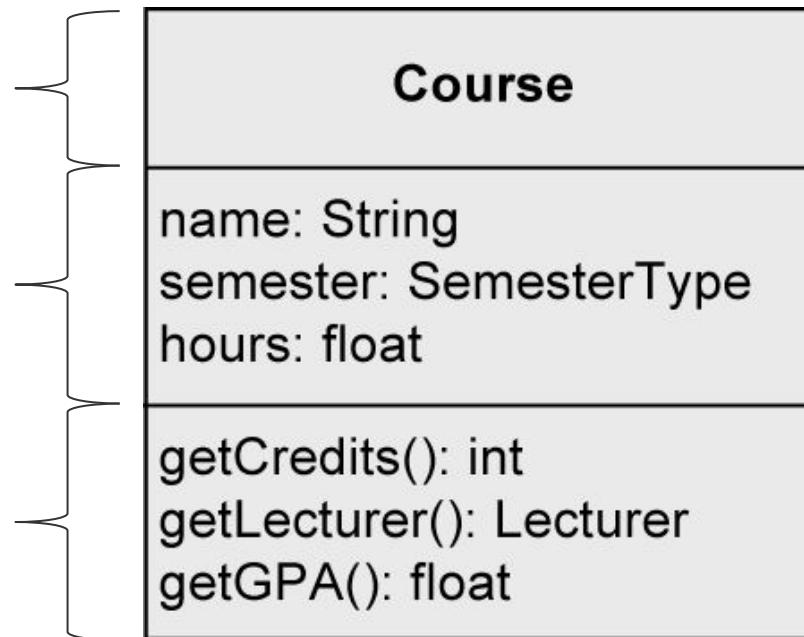


# Classe

*Nome da classe*

*Atributos*

*Operações*

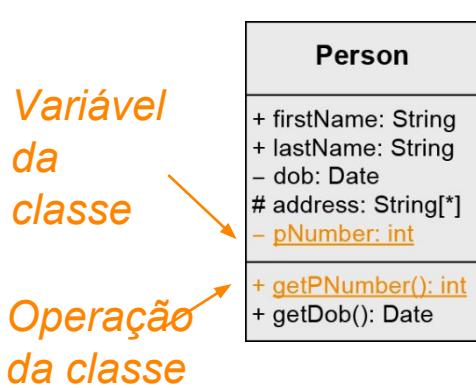


## Exemplo de classes

- Definir os atributos das classes
  - Espectador
  - Campeonato
  - TipoCampeonato
  - LigaPrivada
  - Palpite

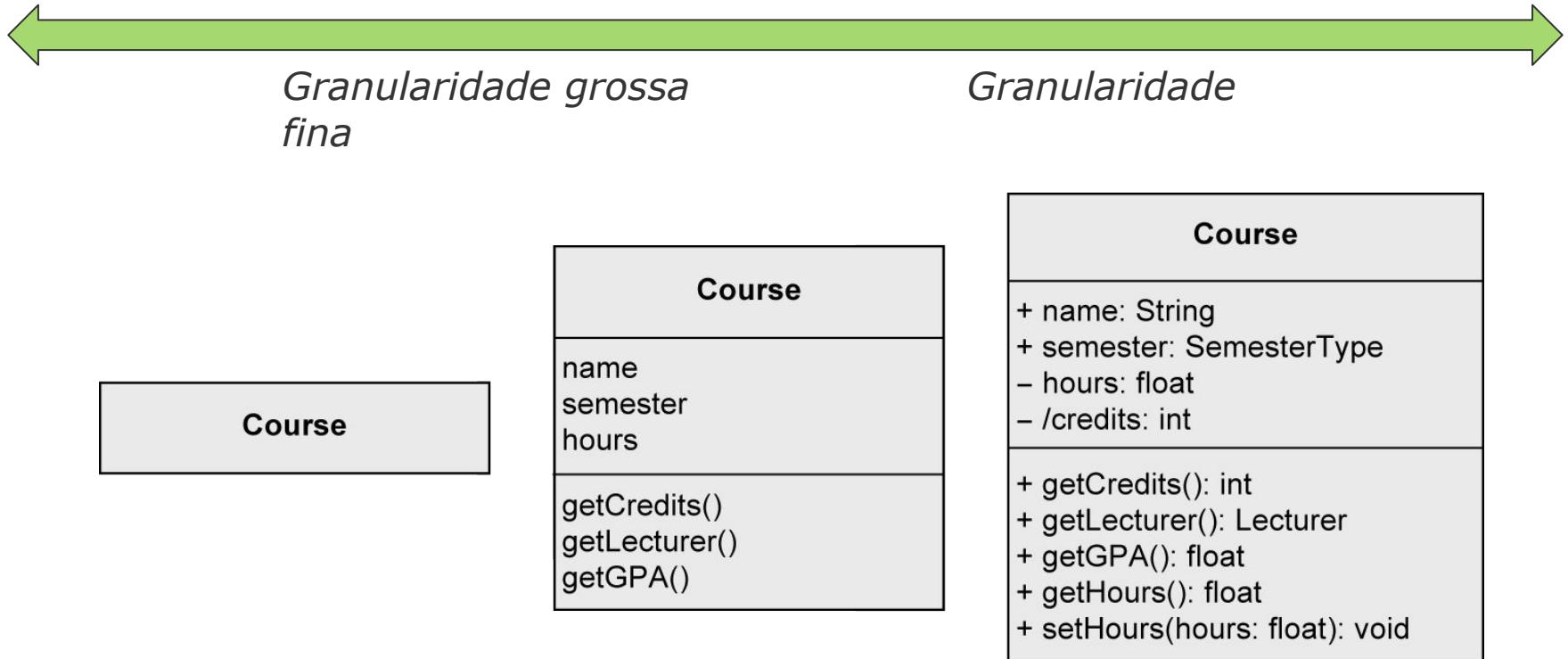
# Variáveis e operações da classe

- Variável de instância: atributos definidos a nível de instância
- Variável de classe
  - Definida apenas uma vez, isto é, compartilhado por todas as instâncias
  - Ex: constantes, contadores
- Operação de classe (estática)
  - Pode ser chamada mesmo se nenhuma instância da classe foi criada
  - Ex: operações matemáticas, criação de instância
- Notação: nome da variável ou operação sublinhado



```
class Person {  
  
    public String firstName;  
    public String lastName;  
    private Date dob;  
    protected String[] address;  
    private static int pNumber;  
    public static int getPNumber() { ... }  
    public Date getDob() { ... }  
}
```

# Especificação das classes: diferentes níveis de detalhes

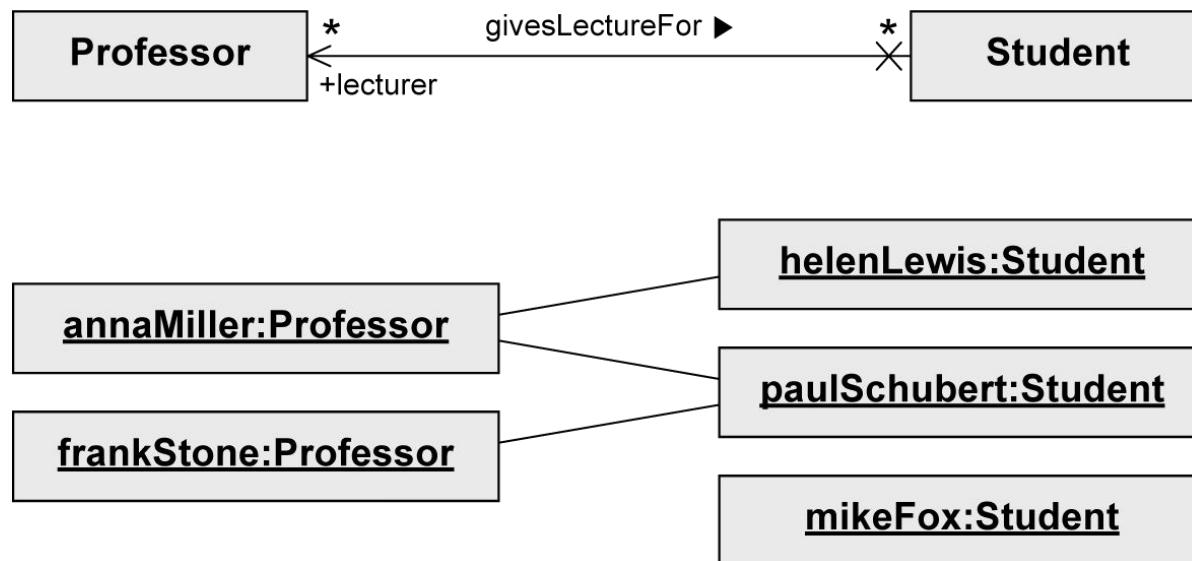


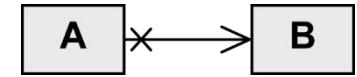
## Exemplo de classes

- Definir os atributos das classes
  - Espectador
  - Campeonato
  - TipoCampeonato
  - LigaPrivada
  - Palpite
- Métodos especiais:
  - Construtores
  - get/set
  - `toString` (java)

# Associação

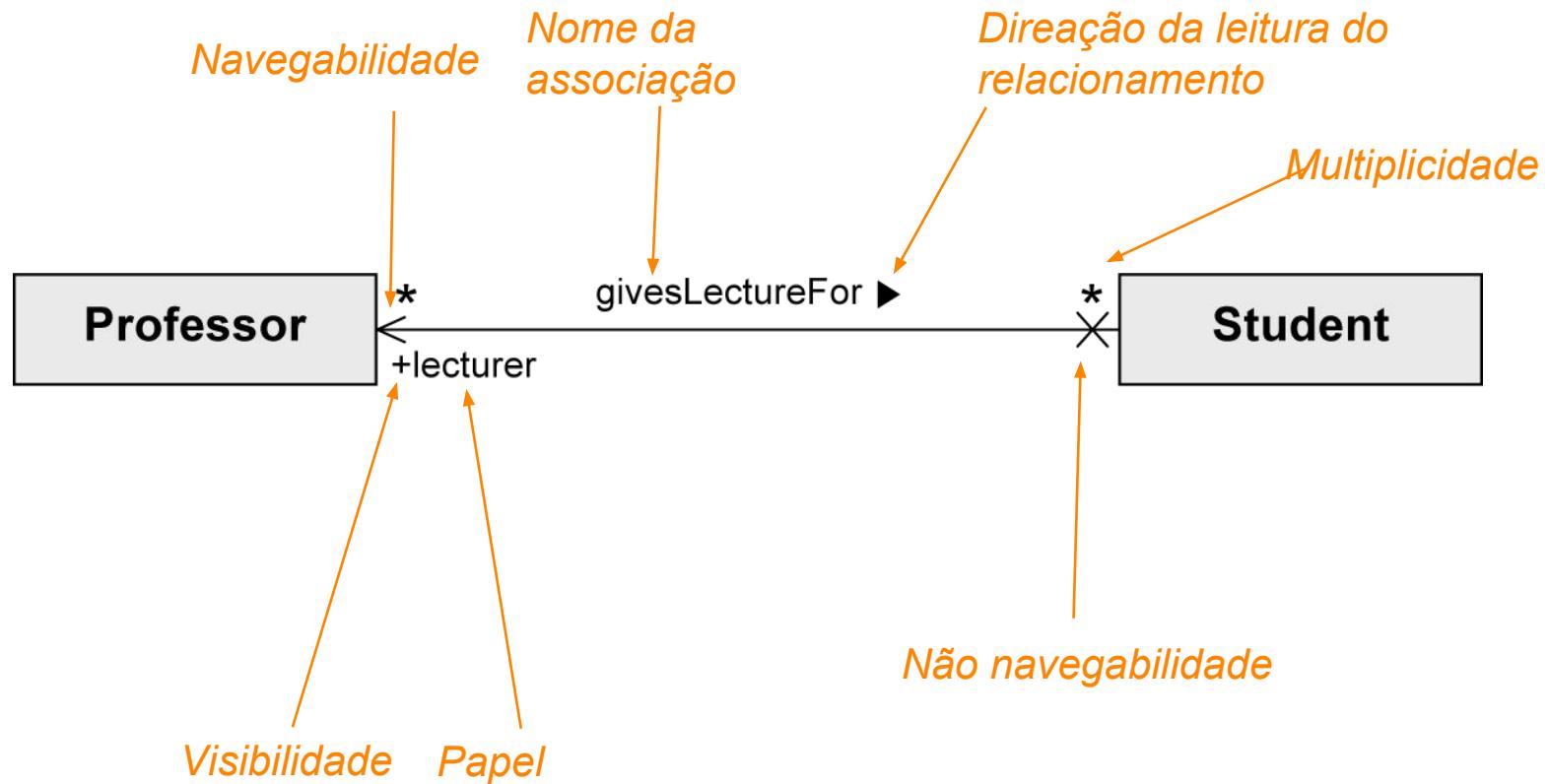
- Modela um possível relacionamento entre instâncias das classes





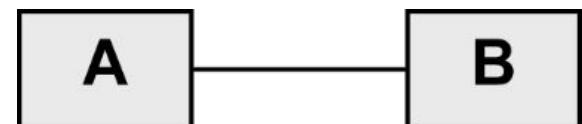
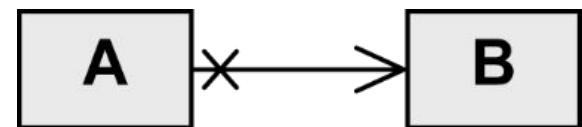
## Associação binária

- Conecta instâncias de duas classes



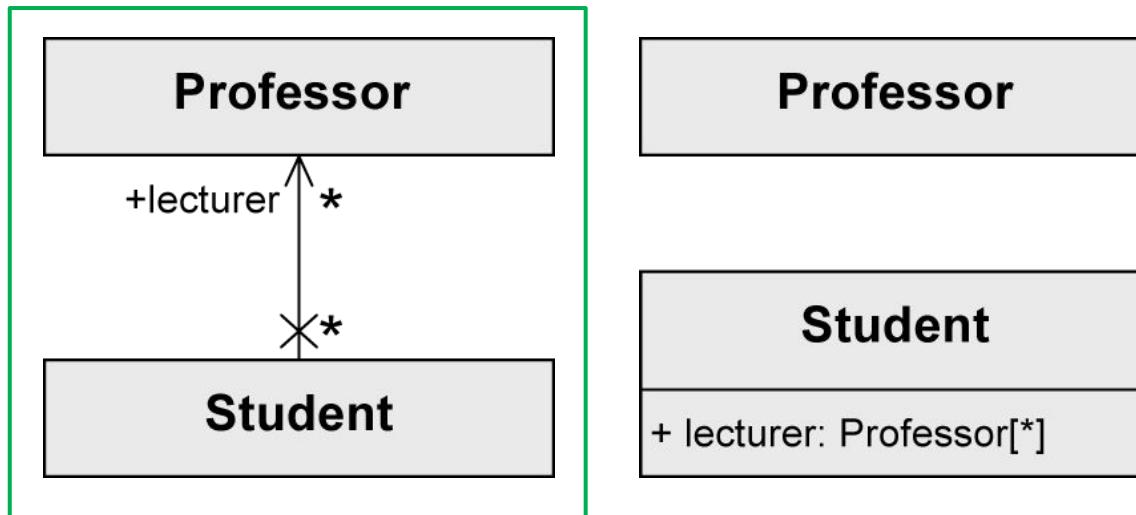
## Associação binária - navegabilidade

- **Navegabilidade:** um objeto conhece a existência e além disso pode acessar seus atributos e operações visíveis
  - Indicado por uma seta aberta
- **Não-navegabilidade**
  - Indicado por uma cruz
- **Exemplo**
  - A pode acessar os atributos e operações visíveis de B
  - B não pode acessar quaisquer artibutos e operações de A
- **Navegabilidade indefinida**
  - Assume-se uma navegabilidade bidirecional



# Associação binária como um atributo

*É a conversão  
mais indicada*



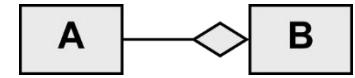
- Código em Java

```
class Professor {...}

class Student{
    public Professor[] lecturer;
    ...
}
```

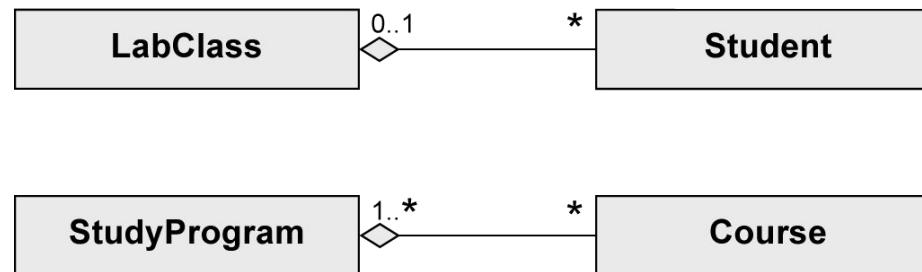
# Agregação

- Tipo especial de associação
- Usado para expressar que uma classe é parte de outra
- Propriedades das agregações:
  - **Transitiva:** se **B** é parte de **A** e **C** é parte de **B**, **C** também é parte de **A**
  - **Assimétrica:** não é possível para **A** ser parte de **B** e **B** ser parte de **A** simultaneamente
  - **Ex: Campeonato, turno, partida**
  -
- **Dois tipos:**
  - Agregação compartilhada
  - Composição



## Agregação compartilhada

- Expressa um elo fraco de pertencimento de partes para um todo  
= Partes existem independente do todo
- Multiplicidade na ponta da agregação pode ser  $> 1$   
= Um elemento pode ser parte de vários outros elementos simultaneamente
- Semelhante a um grafo direcionado acíclico
- Sintaxe: Diamante vazado no fim da agregação
- Exemplos:
  - Aluno é parte de uma turma
  - Disciplina (course) é parte de um curso (StudyProgram)





## Composição

- Existência de uma dependência entre o objeto composto e suas partes
- Uma parte pode ser contida em no máximo um objeto composto em um momento específico  
Multiplicidade é no máximo de 1  
-> O objeto composto forma uma árvore
- Se o objeto composto deixa de existir suas partes também são excluídas
- Sintaxe: Diamante sólido no fim da agregação
- Exemplo: Projetor (Beamer) é parte de um sala de aula (LectureHall) que é parte de um prédio (Building)

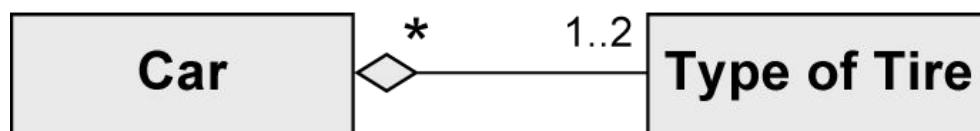
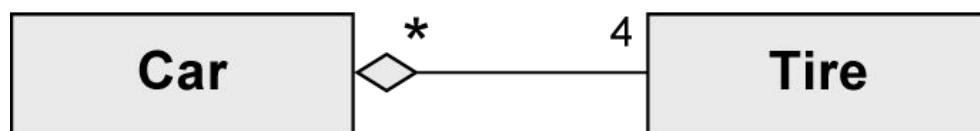
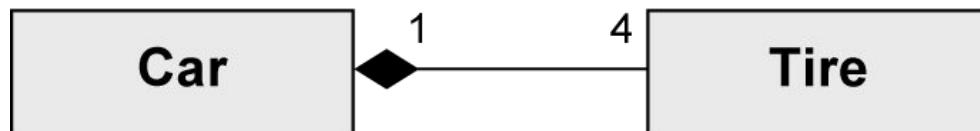
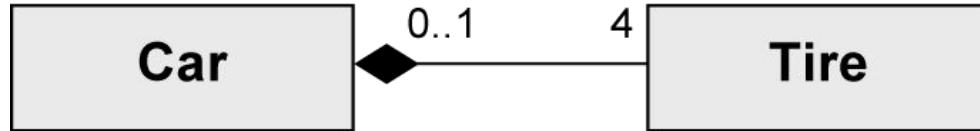


*Se o prédio é apagado do sistema a sala, de aula também é apagada*

*O projetor pode existir sem a sala de aula, mas se ele está atrelado à sala de aula quando esta é apagada ele também é apagado*

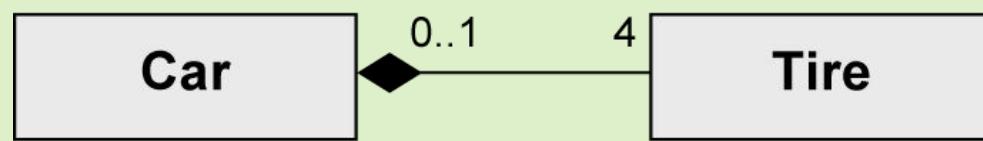
# Agregação (compartilhada) vs composição

- Que modelo se aplica?



# Agregação (compartilhada) vs composição

- Que modelo se aplica?



Um pneu pode existir sem um carro e um pneu pertence a no máximo um carro.

Sim

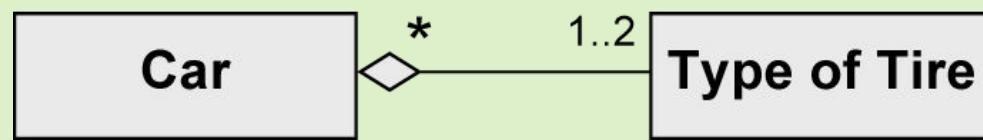


Um pneu não pode existir sem um carro

Não



Um pneu pode pertencer a vários carros

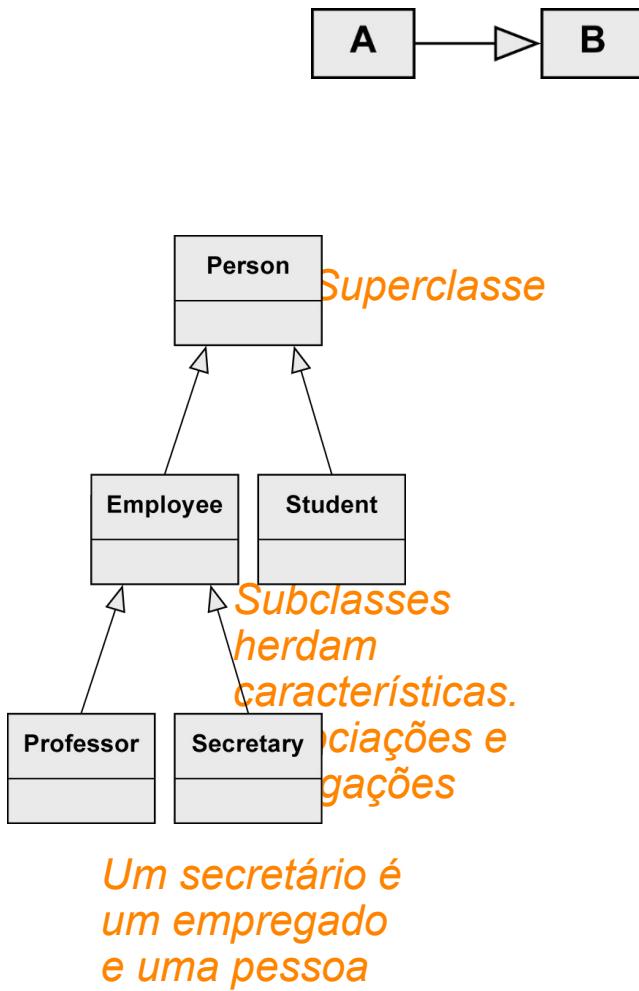


Um carro possui um ou dois tipos diferentes de pneu. Vários carros possuem o mesmo tipo de pneu.

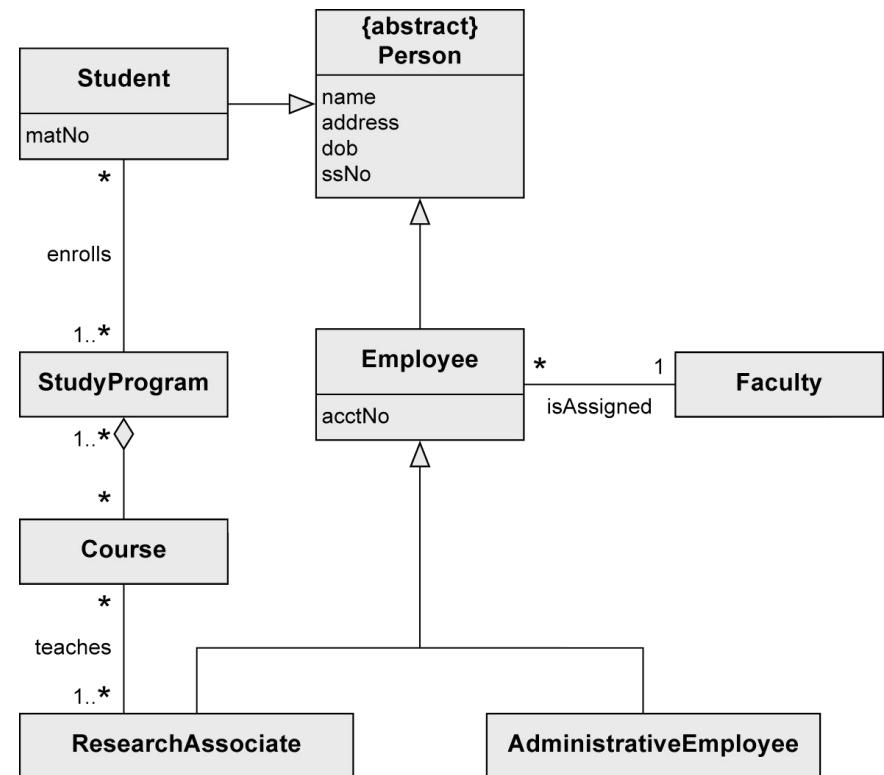
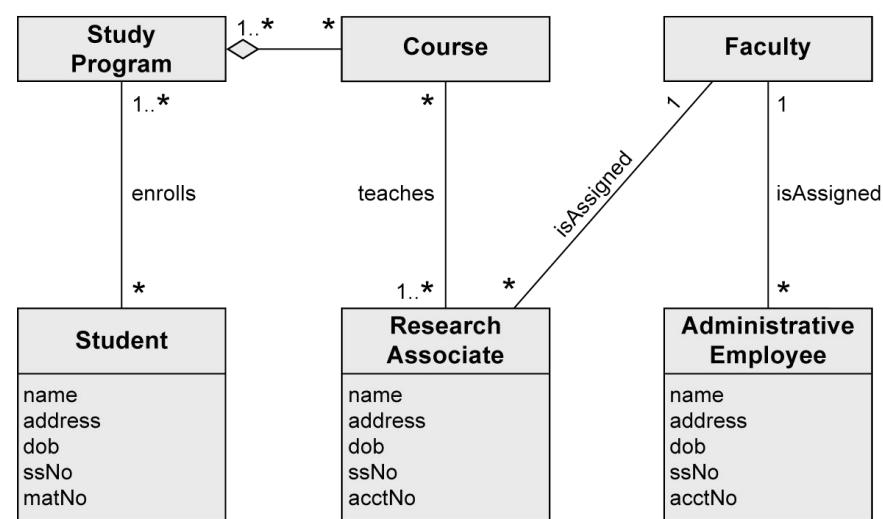
Sim

# Generalização

- Características (atributos e operações), associações e agregações que são especificados para a classe mais geral (superclasse) são passados para suas subclasses
- Toda instância da subclasse é simultaneamente uma instância indireta da superclasse
- Subclasse herda todas as características, associações e agregações da superclasse, exceto os membros privados
- Subclasses pode ter características, associações e agregações adicionais
- Generalizações são transitivas (se **B** herda de **A** e **C** herda de **B**, então **C** herda de **A**)

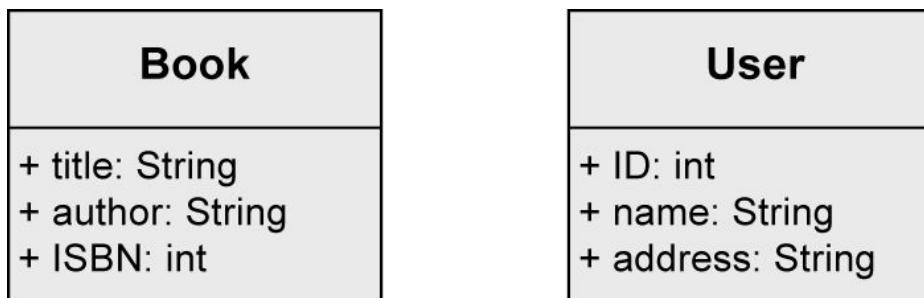


# Exemplo com e sem generalização



# Criando um diagrama de classes

- Não é possível extrair completamente classes, atributos e associações de textos de linguagem natural automaticamente
- Diretrizes
  - Substantivos geralmente indicam classes
  - Adjetivos indicam atributos
  - Verbos indicam operações
- Exemplo: O sistema de gerenciamento da biblioteca armazena usuários com ID único, nome e endereço bem como os livros com seus títulos, autores e número ISBN.



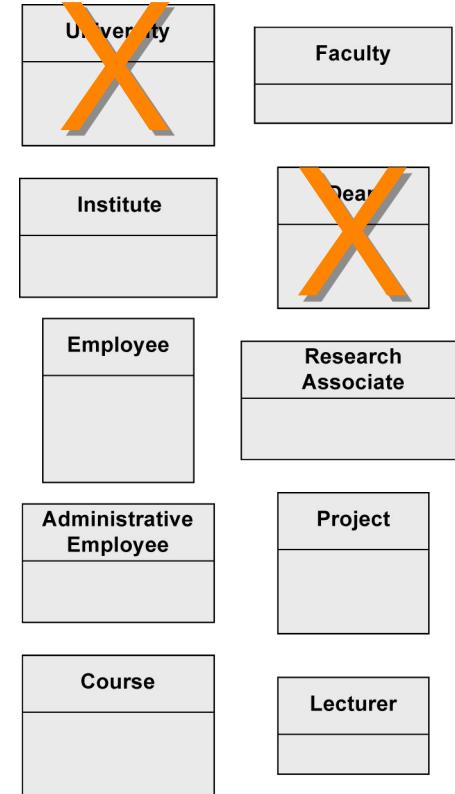
## Exemplo – Sistema de informações de uma universidade

- Uma universidade consiste de múltiplas faculdades que são compostas de vários departamentos. Cada faculdade e cada departamento tem um nome. Um endereço é conhecido para cada departamento.
- Cada faculdade é dirigida por um diretor que é um empregado da universidade.
- O número total de empregados é conhecido. Empregados tem um número de PASEP, um nome e um endereço de e-mail. Há distinção entre pesquisadores e técnicos administrativos.
- Pesquisadores são associados a pelo menos um departamento. O campo de estudo de cada pesquisador é conhecido. Além disso, pesquisadores podem estar envolvidos em projetos com um certo número de horas. Nome, data de início e data de fim do projeto são conhecidos. Alguns pesquisadores também ministram disciplinas. Sendo classificados como professores.
- Disciplinas tem um número de identificação, e uma carga horária semanal em horas.

# Exemplo – Passo 1: Identificando as classes

- Uma universidade consiste de múltiplas faculdades que são compostas de vários departamentos. Cada faculdade e cada departamento tem um nome. Um endereço é conhecido para cada departamento.
- Cada faculdade é dirigida por um diretor que é um empregado da universidade.
- O número total de empregados é conhecido. Empregados tem um número de PASEP, um nome e um endereço de e-mail. Há distinção entre pesquisadores e técnicos administrativos.
- Pesquisadores são associados a pelo menos um departamento. O campo de estudo de cada pesquisador é conhecido. Além disso, pesquisadores podem estar envolvidos em projetos com um certo número de horas. Nome, data de início e data de fim do projeto são conhecidos. Alguns pesquisadores também ministram disciplinas. Sendo classificados como professores.
- Disciplinas tem um número de identificação, e uma carga horária semanal em horas.

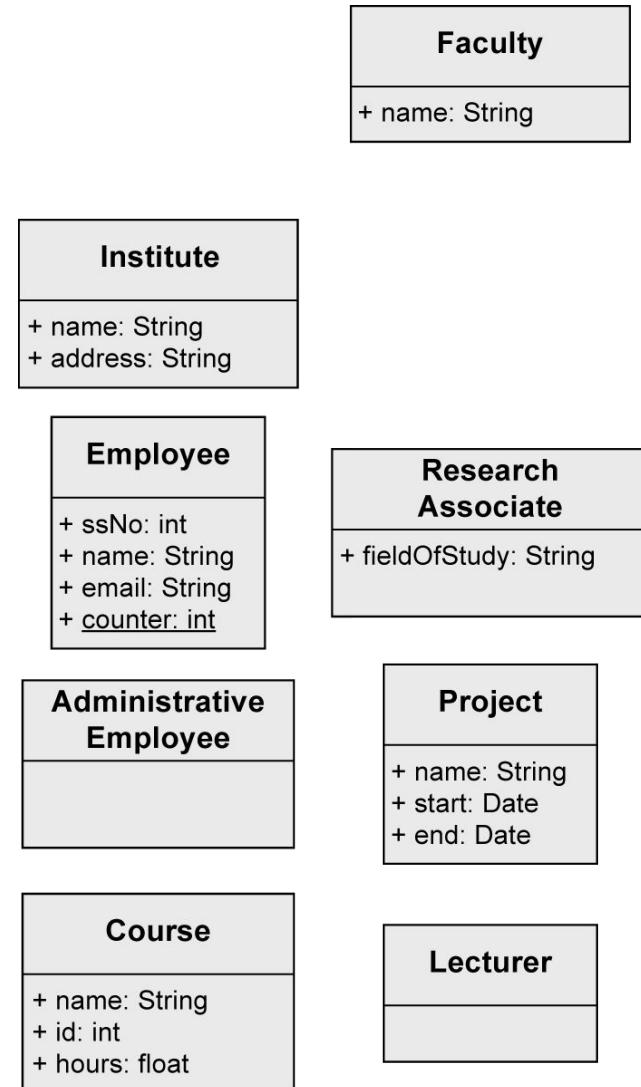
*Vamos modelar o sistema  
“Universidade”*



*Diretor não tem atributos além  
dos outros empregados*

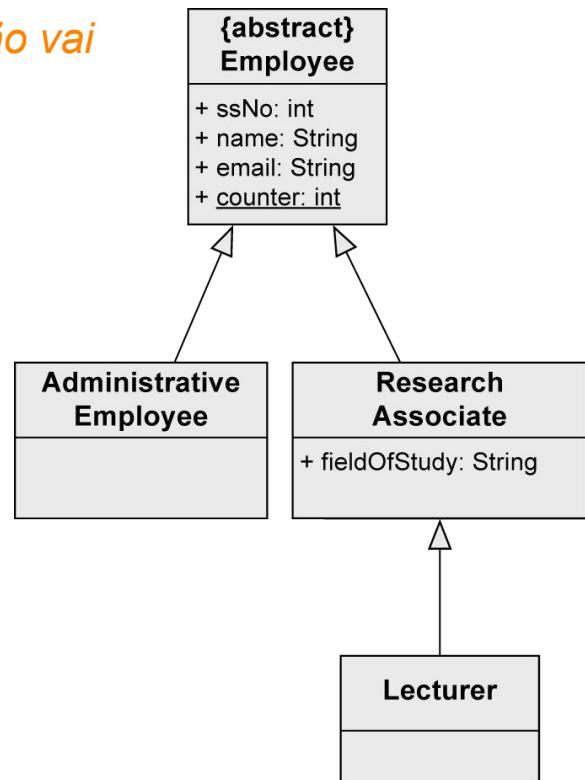
## Exemplo – Passo 2: Identificando os atributos

- Uma universidade consiste de múltiplas faculdades que são compostas de vários departamentos. Cada faculdade e cada departamento tem um nome. Um endereço é conhecido para cada departamento.
- Cada faculdade é dirigida por um diretor que é um empregado da universidade.
- O número total de empregados é conhecido. Empregados tem um número de PASEP, um nome e um endereço de e-mail. Há distinção entre pesquisadores e técnicos administrativos.
- Pesquisadores são associados a pelo menos um departamento. O campo de estudo de cada pesquisador é conhecido. Além disso, pesquisadores podem estar envolvidos em projetos com um certo número de horas. Nome, data de início e data de fim do projeto são conhecidos. Alguns pesquisadores também ministram disciplinas. Sendo classificados como professores.
- Disciplinas tem um número de identificação, e uma carga horária semanal em horas.



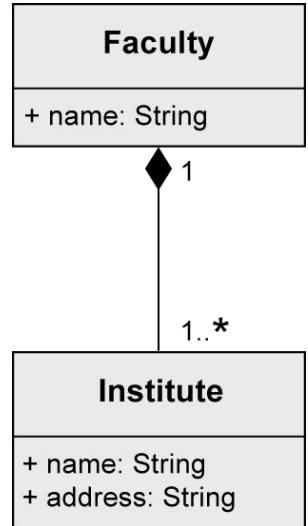
## Exemplo – Passo 3: Identificando relacionamentos (1/6)

- Três tipos de relacionamento
    - Associação
    - Generalização
    - Agregação
  - Indícios de generalização
  - “Há distinção entre pesquisadores e técnicos administrativos”
  - “Alguns pesquisadores também ministram disciplinas. Sendo classificados como professores.”
- Abastrato, isto é, não vai ser Instanciado.*



## Exemplo – Passo 3: Identificando relacionamentos (2/6)

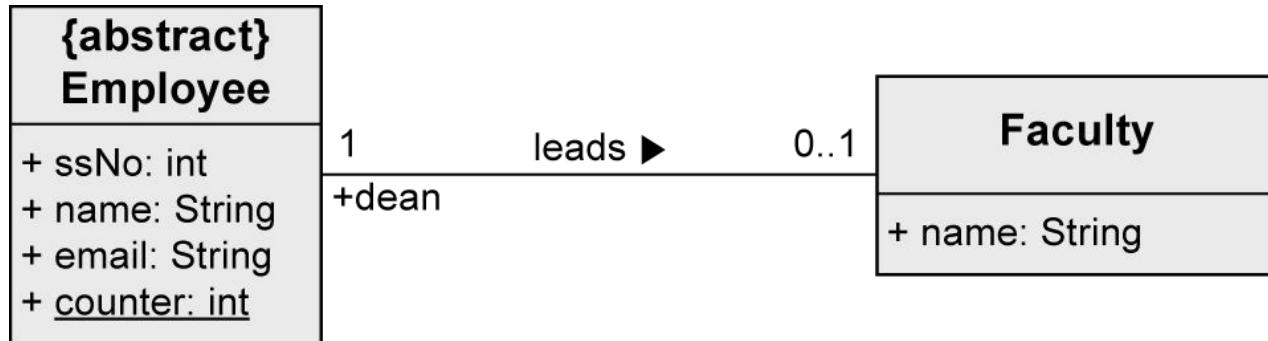
- “Uma universidade consiste de múltiplas faculdades que são compostas de vários departamentos.”



Composição para mostrar a dependência

## Exemplo – Passo 3: Identificando relacionamentos (3/6)

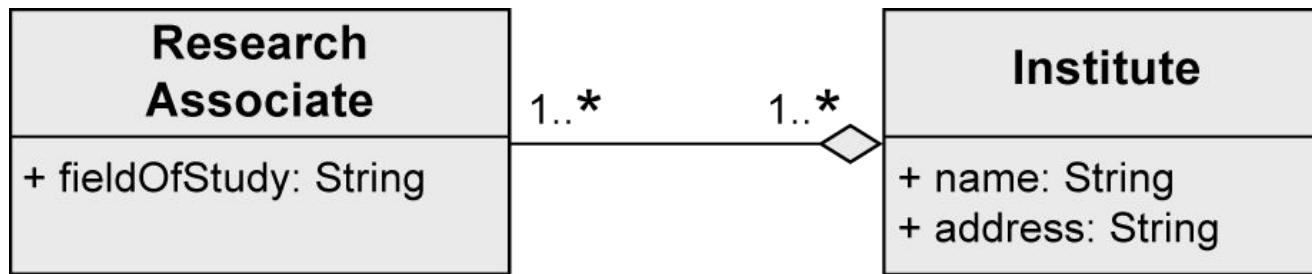
- “Cada faculdade é dirigida por um diretor que é um empregado da universidade.”



*No relacionamento dirige, um empregado assume o papel de diretor*

## Exemplo – Passo 3: Identificando relacionamentos (4/6)

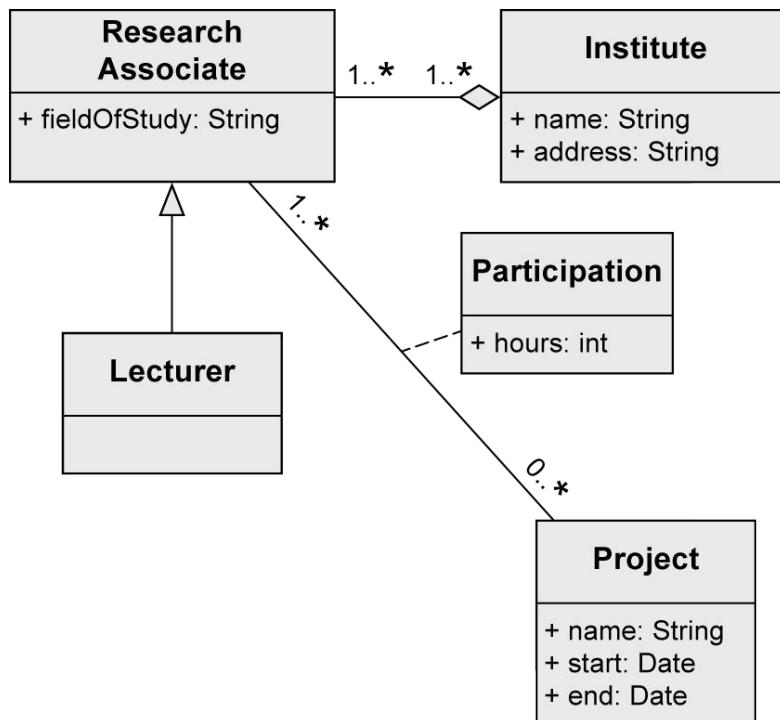
- “Pesquisadores são associados a pelo menos um departamento.”



Agregação compartilhada para indicar que um pesquisador é parte de um departamento mas não há uma dependência existencial

## Exemplo – Passo 3: Identificando relacionamentos (5/6)

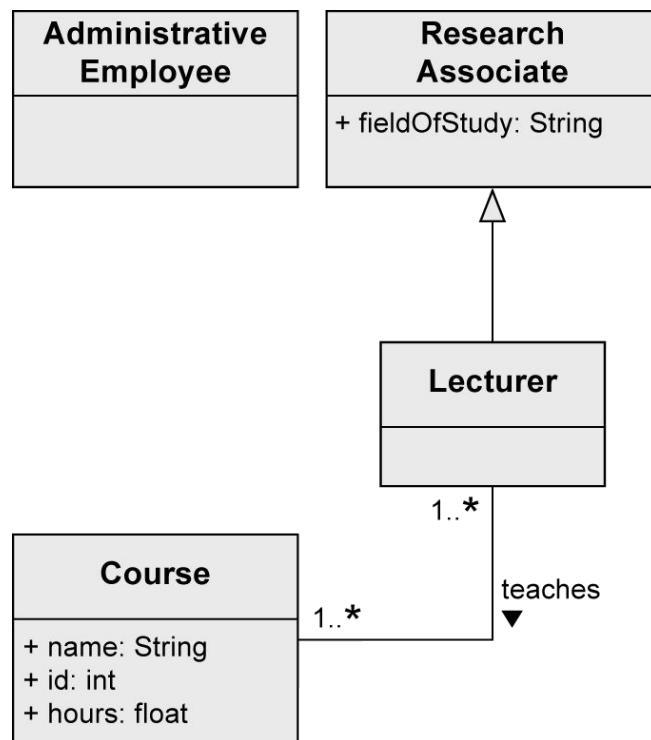
- “Além disso, pesquisadores podem estar envolvidos em projetos com um certo número de horas.”



*Uma classe de associação para armazenar o número de horas de cada pesquisador para cada projeto.*

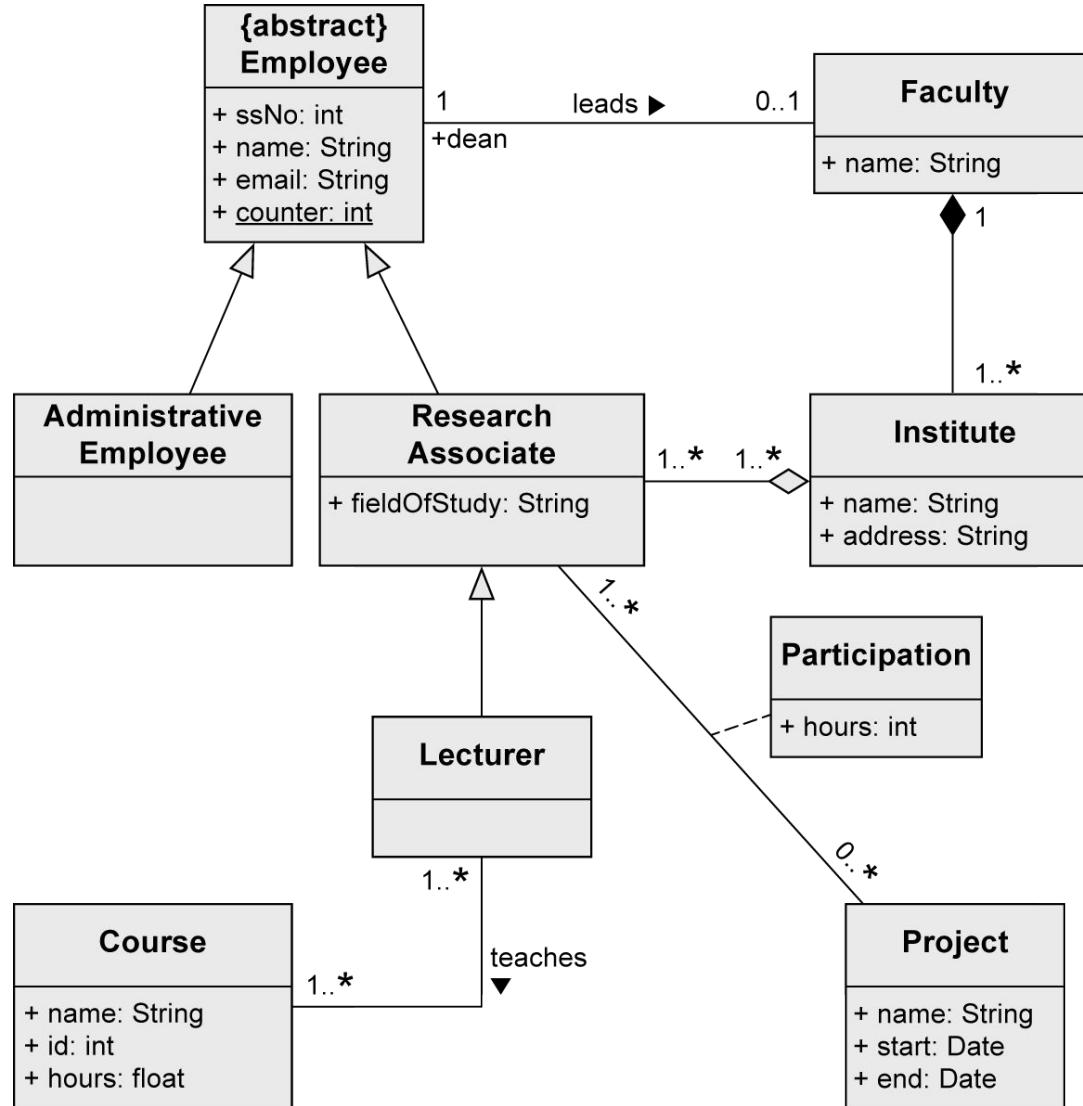
## Exemplo – Passo 3: Identificando relacionamentos (6/6)

- “Alguns pesquisadores também ministram disciplinas. Sendo classificados como professores.”



*Professores herdam todas as Características, associações e Agregações de pesquisadores. Mas tem uma associação De ensinar disciplinas.*

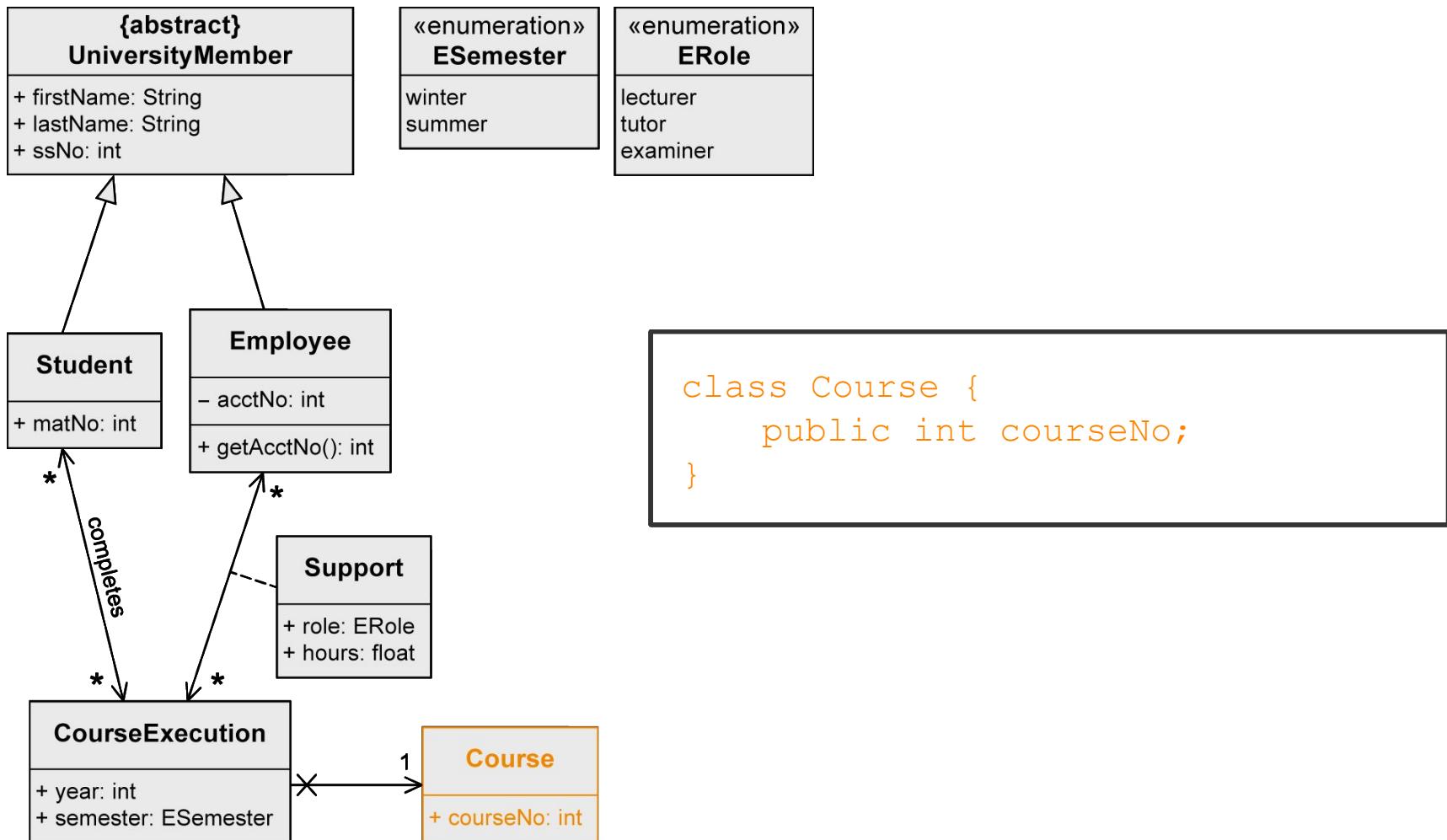
# Exemplo – Diagrama de classes completo



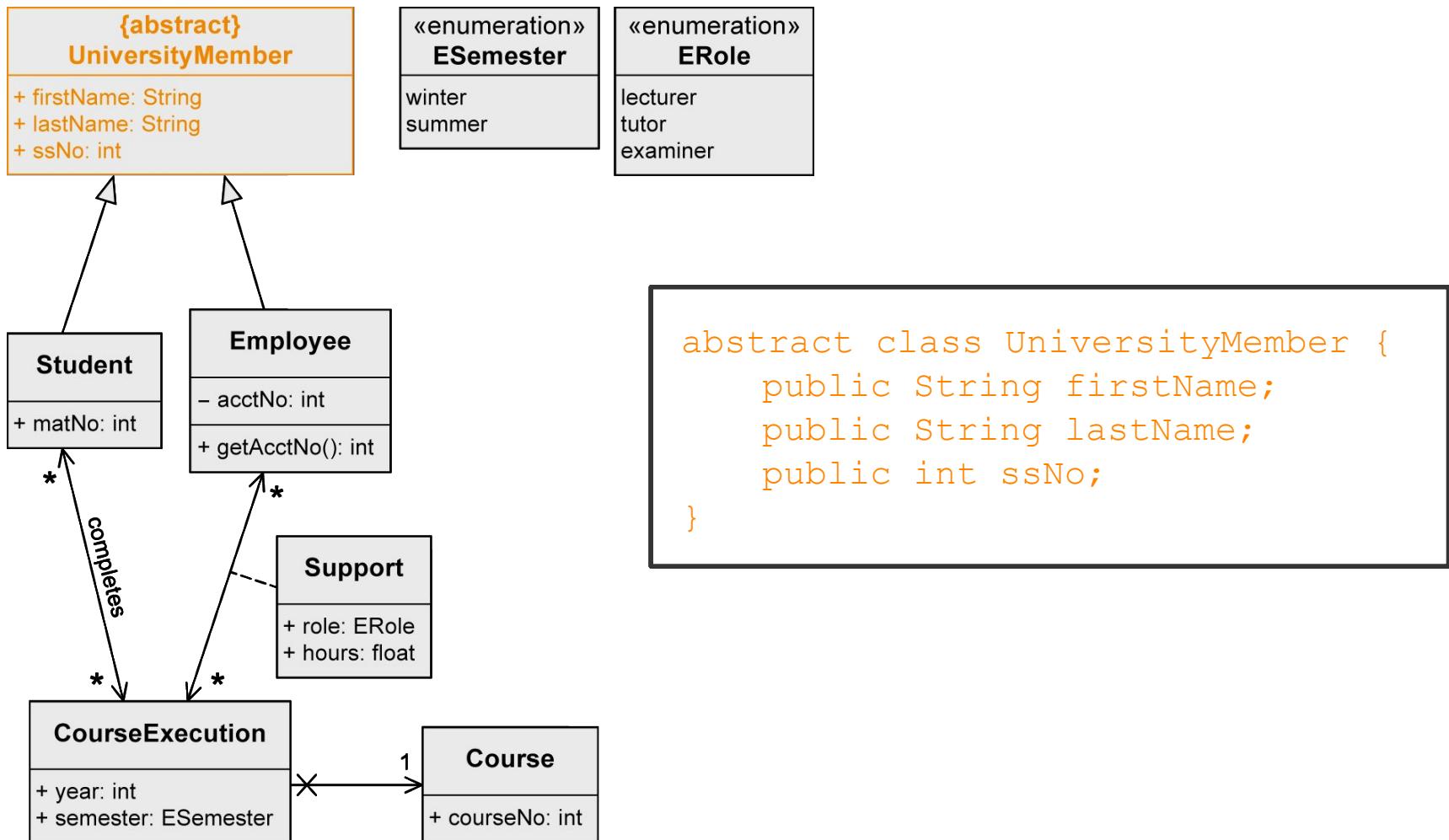
## Geração de código

- Diagramas de classe geralmente são criados com a intenção de implementar os elementos modelados em uma linguagem de programação orientada a objetos
- Frequentemente, a tradução é semi-automática e requer uma intervenção manual mínima

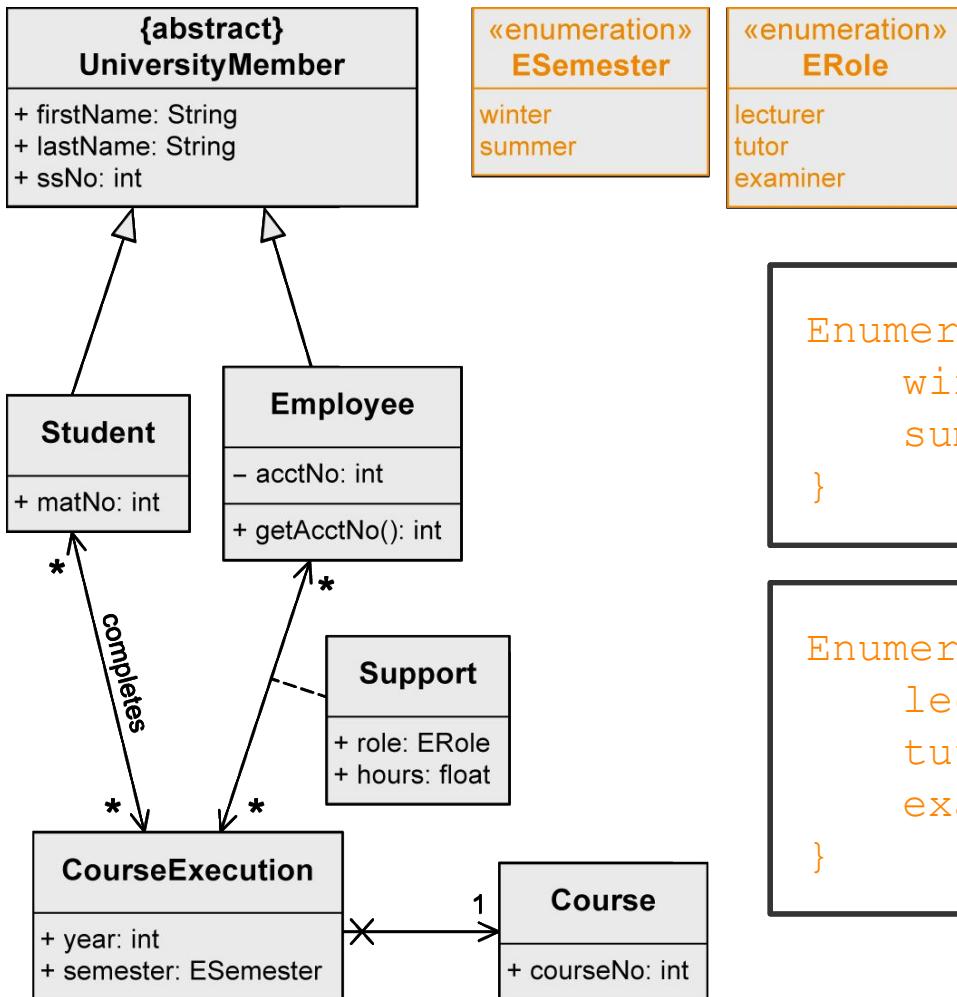
## Exemplo: geração de código (1/6)



## Exemplo: geração de código (2/6)



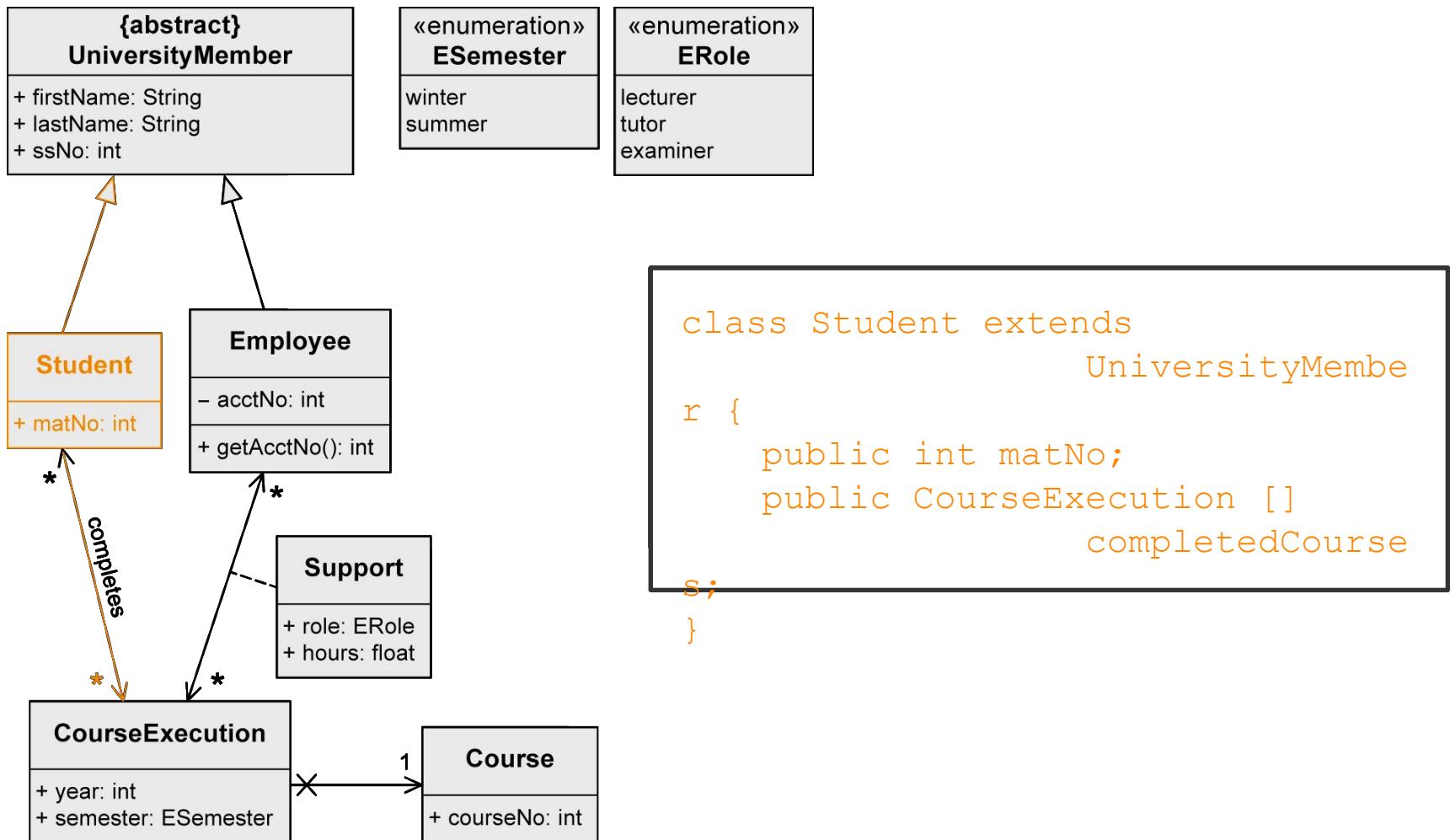
## Exemplo: geração de código (3/6)



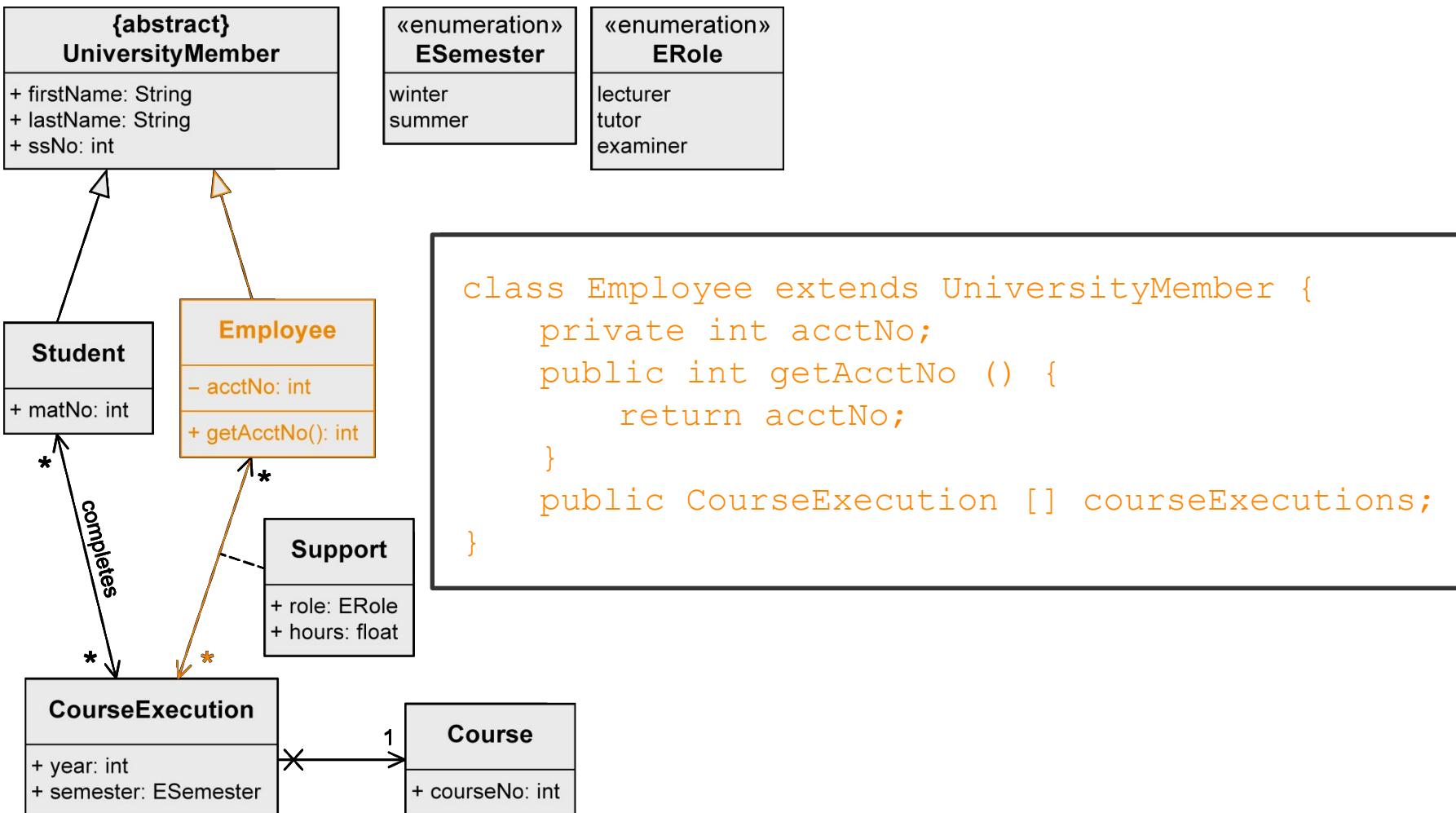
```
Enumeration ESemester {  
    winter,  
    summer  
}
```

```
Enumeration ERole {  
    lecturer,  
    tutor,  
    examiner  
}
```

## Exemplo: geração de código (4/6)



## Exemplo: geração de código (5/6)



## Exemplo: geração de código (6/6)

