

# TEMA 5. DIAGRAMAS DE CLASE

## ANÁLISIS Y DISEÑO DE SISTEMAS DE INFORMACIÓN



Universidad  
Francisco de Vitoria  
**UFV** Madrid

# ÍNDICE

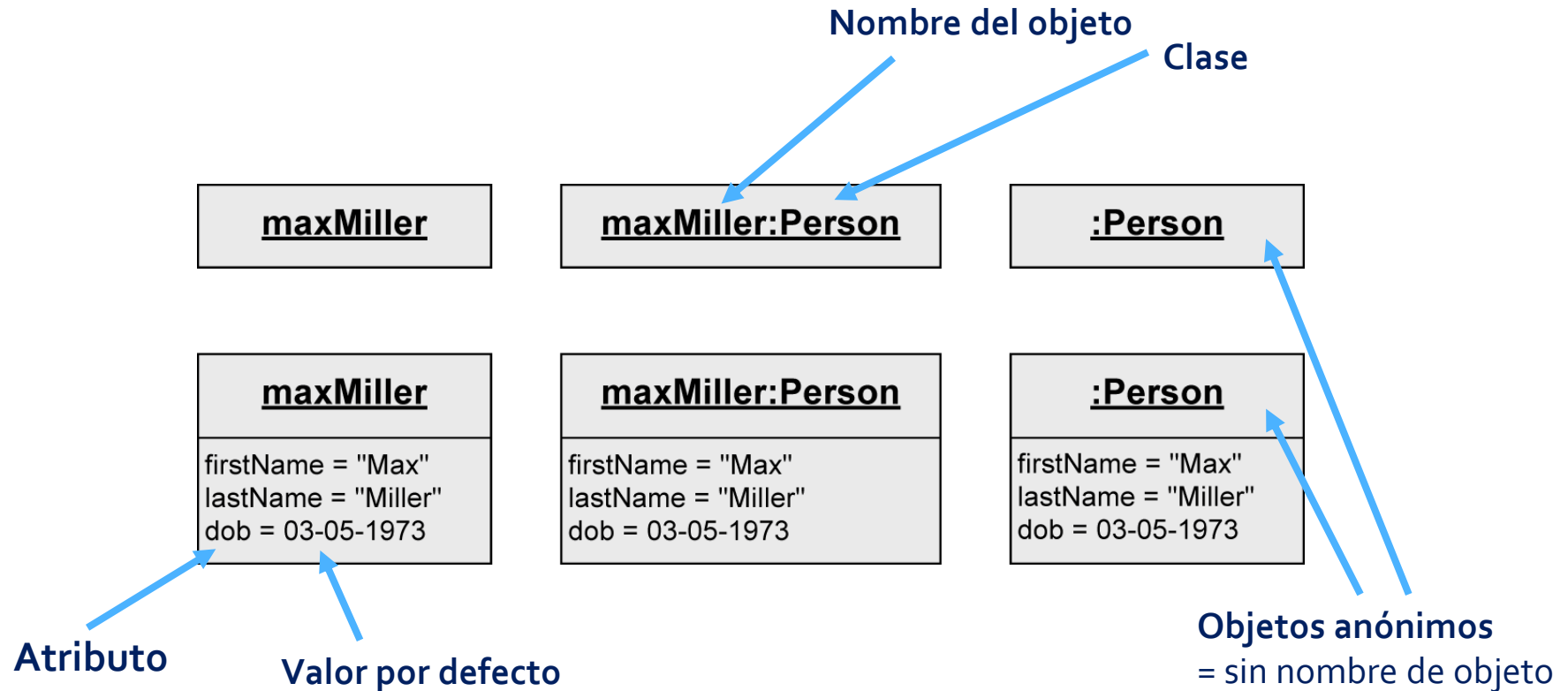
1. Objetos
2. Clases
3. Atributos
4. Operaciones
5. Asociaciones:
  - Asociación binaria
  - Asociación N-aria
  - Clase de asociación
  - Agregación
  - Generalización
6. Crear un diagrama de clases
7. Ejemplo
8. Resumen notación

# 1. OBJETOS

o:C

Son los “actores” del Sistema.

Notaciones:



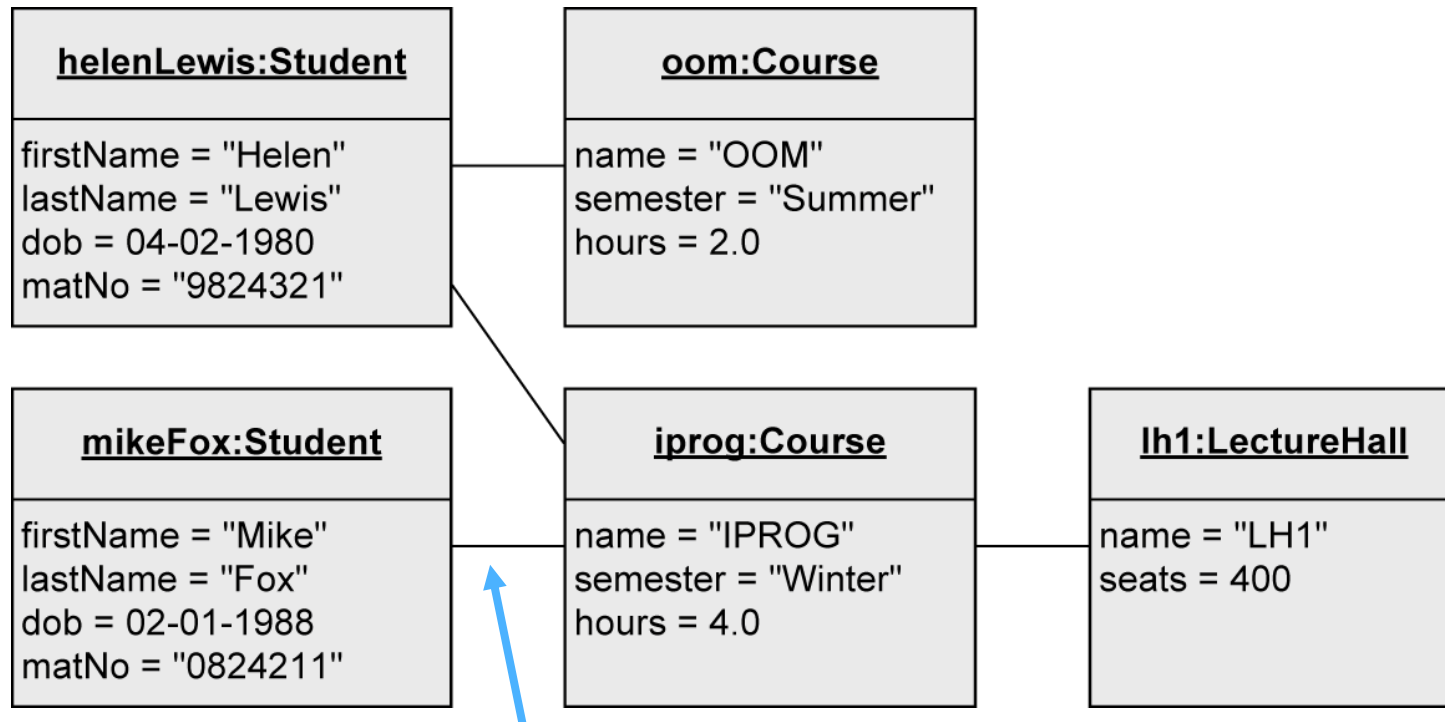
# 1. DIAGRAMA DE OBJETOS

o1

o2

## Objetos del Sistema y sus relaciones

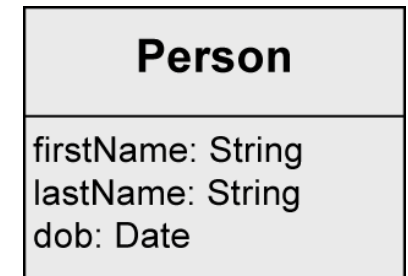
Es una foto concreta de objetos en un momento determinado en el tiempo.



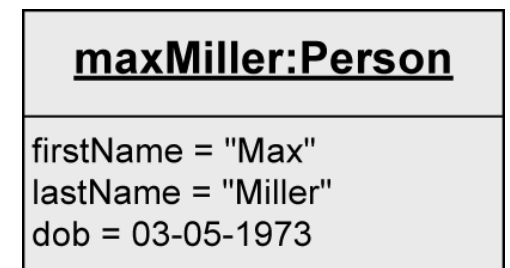
# 1. DE OBJETO A CLASE

- Los elementos de un sistema a menudo tienen características y comportamiento idénticos.
- Una clase es la forma de diseñar un conjunto de objetos similares de un sistema.
- Los objetos son instancias de clases.
- **Atributos:** características estructurales de una clase.
  - Valor diferente para cada instancia (= objeto)
- **Operaciones:** comportamiento de una clase.
  - Idéntico para todos los objetos de una clase → no representados en el diagrama de objeto

Clase



Objeto de la clase

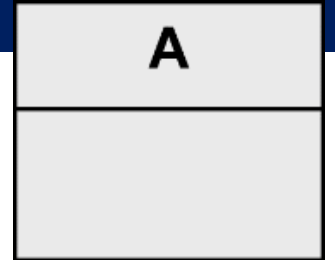
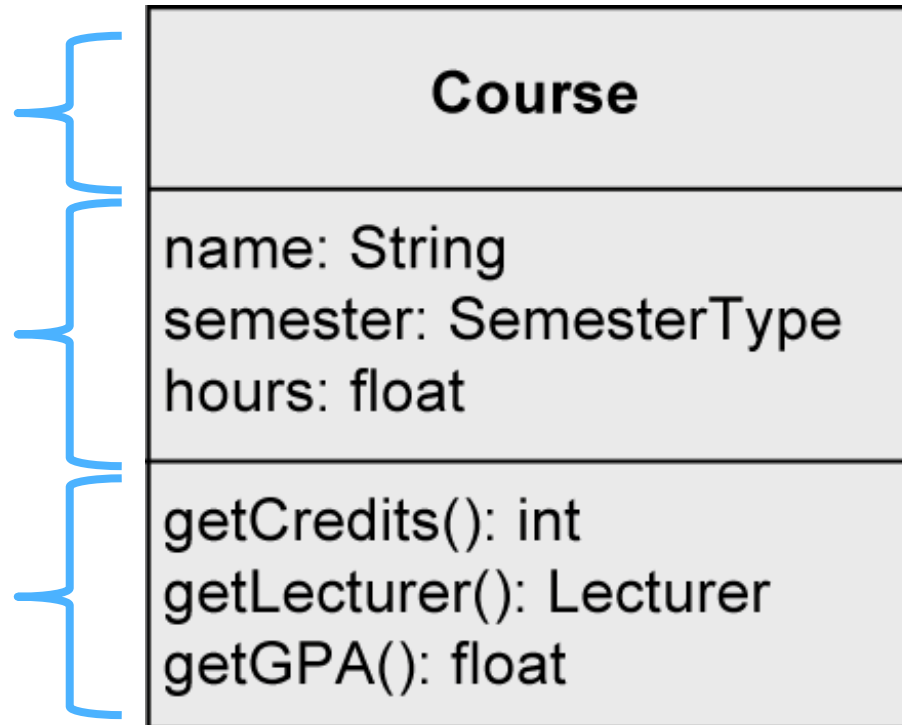


## 2. CLASE

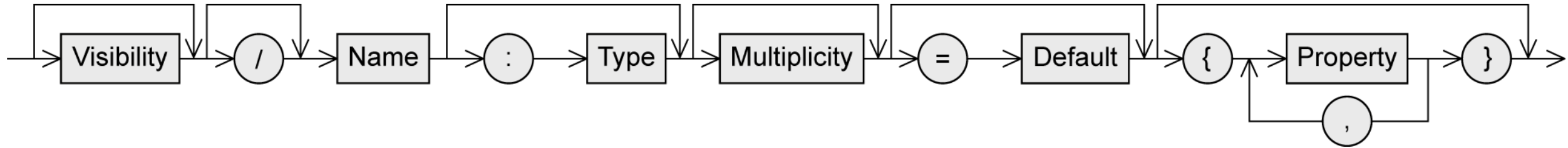
Nombre de la  
clase

Atributos

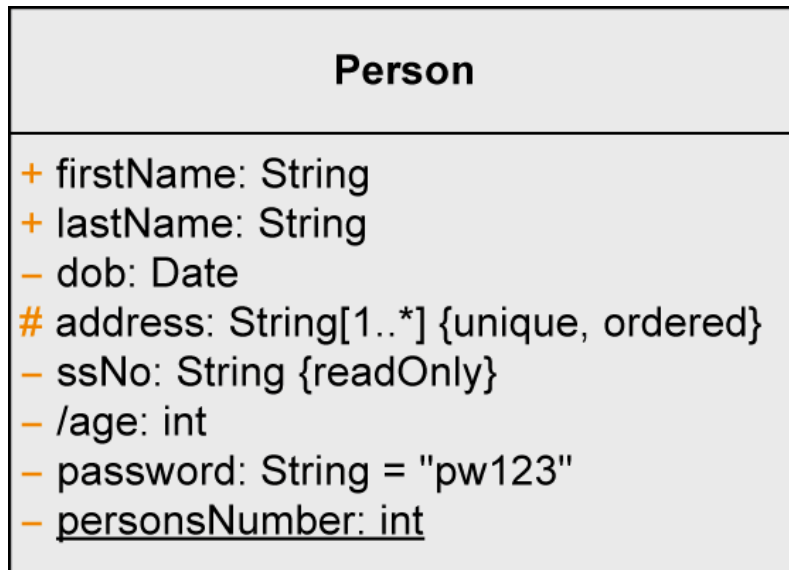
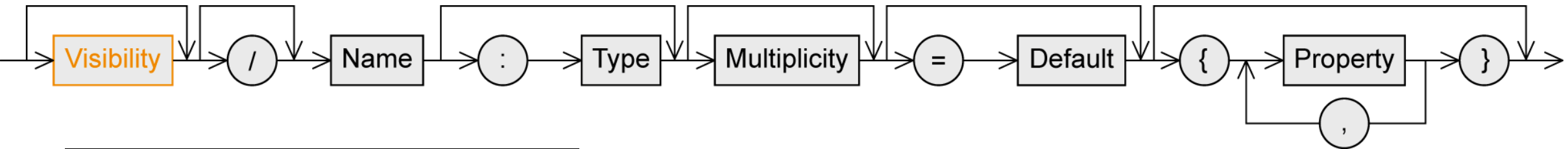
Operaciones



### 3. ATRIBUTO. SINTAXIS DE ATRIBUTO



# 3. SINTAXIS DE ATRIBUTO. VISIBILIDAD

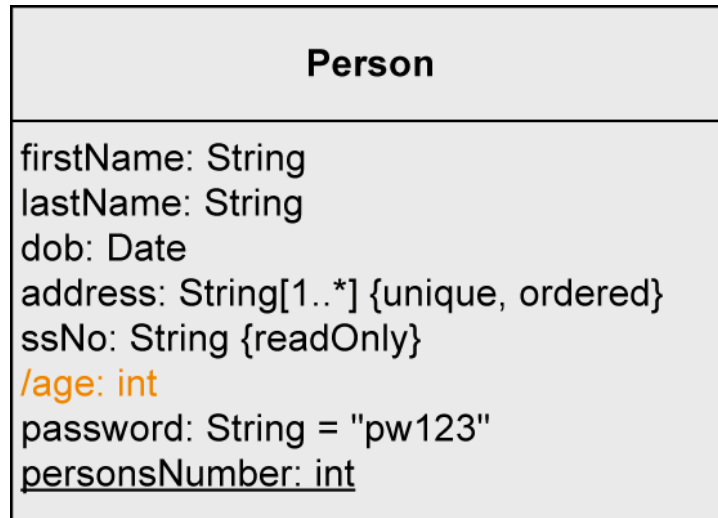
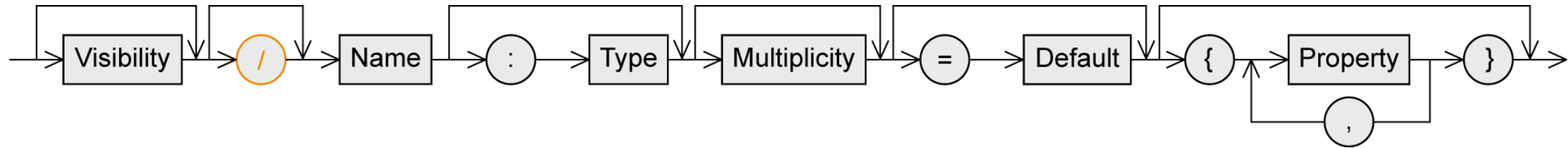


Quién tiene **permiso para acceder al atributo**:

- + ... público: todo el mundo
- ... privado: solo el objeto en sí
- # ... protegido: clase en sí y subclases
- ~ ... paquete: clases que están en el mismo paquete

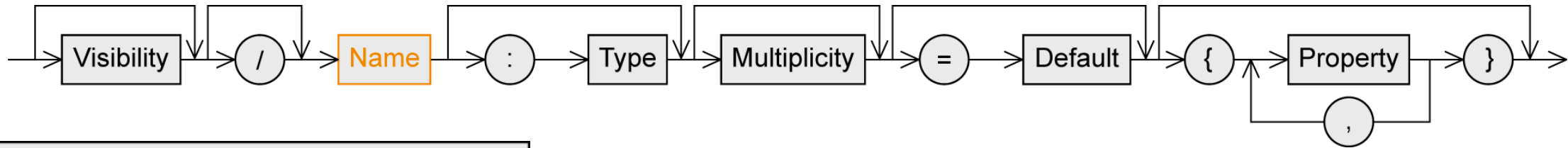


# 3. SINTAXIS DE ATRIBUTO. ATRIBUTO DERIVADO



- El valor del atributo se deriva de otros atributos.
  - **edad** : calculada a partir de la fecha de nacimiento

### 3. SINTAXIS DE ATRIBUTO. NOMBRE

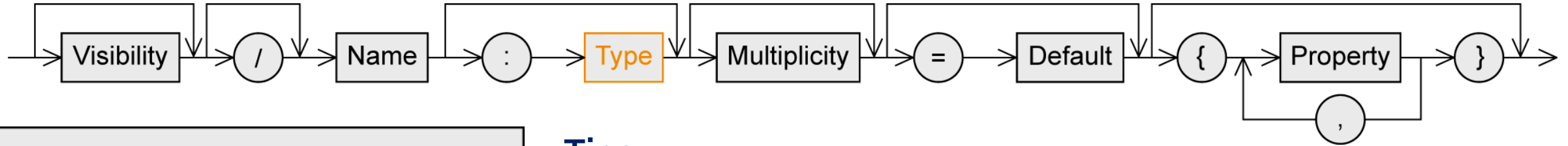


#### Person

**firstName:** String  
**lastName:** String  
**dob:** Date  
**address:** String[1..\*] {unique, ordered}  
**ssNo:** String {readOnly}  
**/age:** int  
**password:** String = "pw123"  
**personsNumber:** int

#### Nombre del atributo

# 3. SINTAXIS DE ATRIBUTO. TIPO



## Person

firstName: **String**  
lastName: **String**  
dob: **Date**  
address: **String**[1..\*] {unique, ordered}  
ssNo: **String** {readOnly}  
/age: **int**  
password: **String** = "pw123"  
personsNumber: **int**

## Tipo

Tipo de dato: valores definidos por el usuario.

Pre-defined: Boolean, Integer, UnlimitedNatural, String

User-defined: «**primitive**»

Composite data type: «**datatype**»

Enumerations: «**enumeration**»

«**primitive**»  
**Float**

round(): void

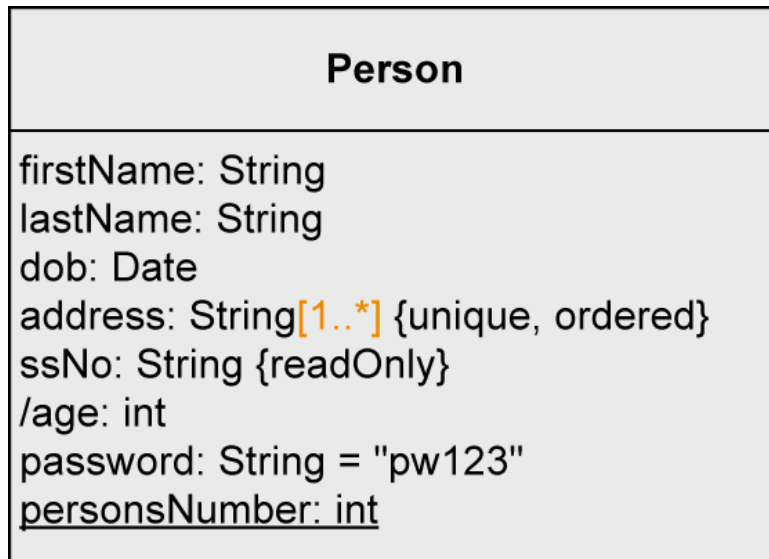
«**datatype**»  
**Date**

day  
month  
year

«**enumeration**»  
**AcademicDegree**

bachelor  
master  
phd

# 3. SINTAXIS DE ATRIBUTO. MULTIPLICIDAD

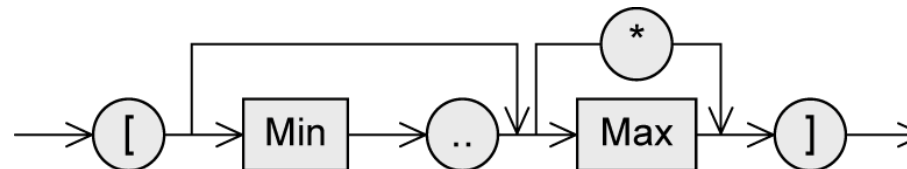


Número de valores que un atributo puede contener

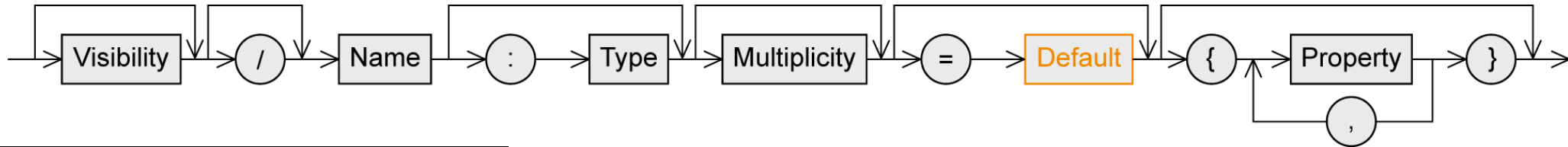
Valor por defecto: 1

Notación: **[min..max]**

**[\*]** or **[0..\*]**



### 3. SINTAXIS DE ATRIBUTO. NOMBRE

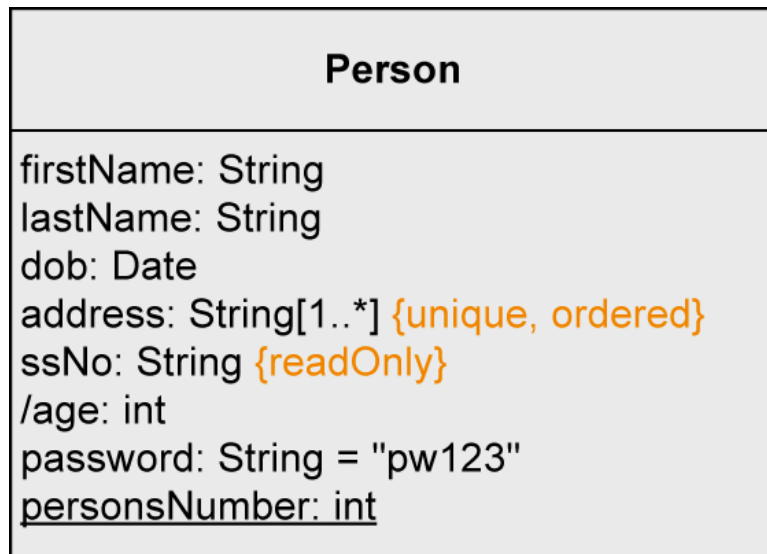
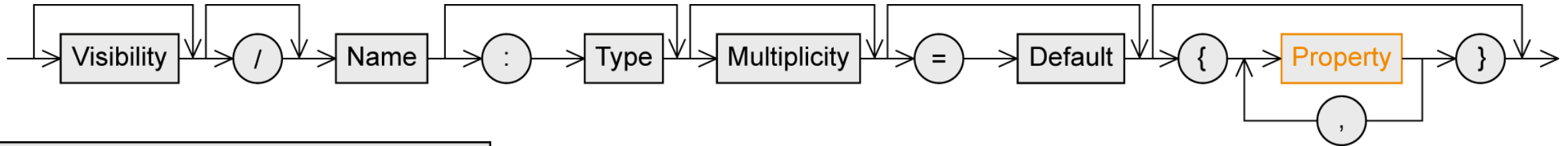


#### Valor por defecto

Se utiliza si el valor no se encuentra configurado explícitamente por el usuario.

Person
firstName: String lastName: String dob: Date address: String[1..*] {unique, ordered} ssNo: String {readOnly} /age: int password: String = "pw123" <u>personsNumber: int</u>

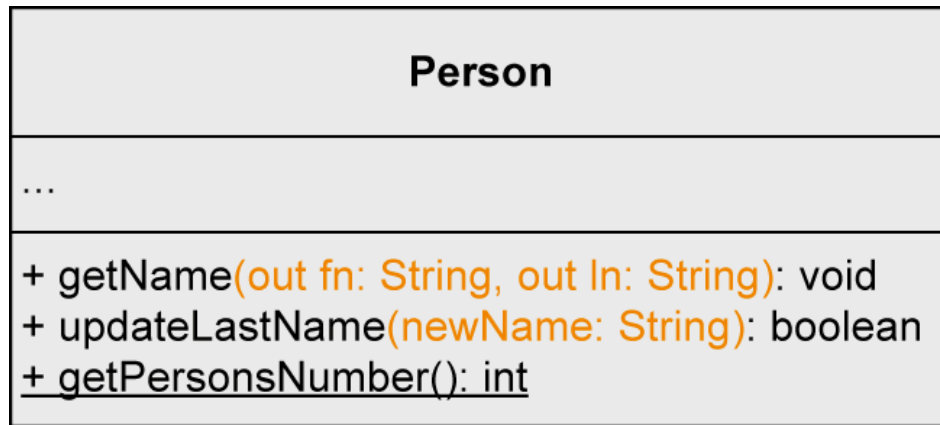
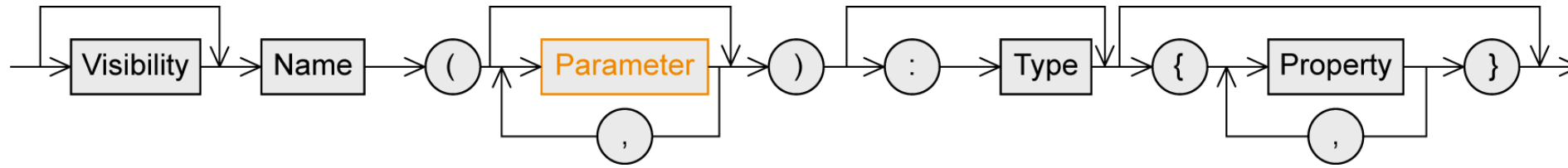
# 3. SINTAXIS DE ATRIBUTO. PROPIEDADES PREDEFINIDAS



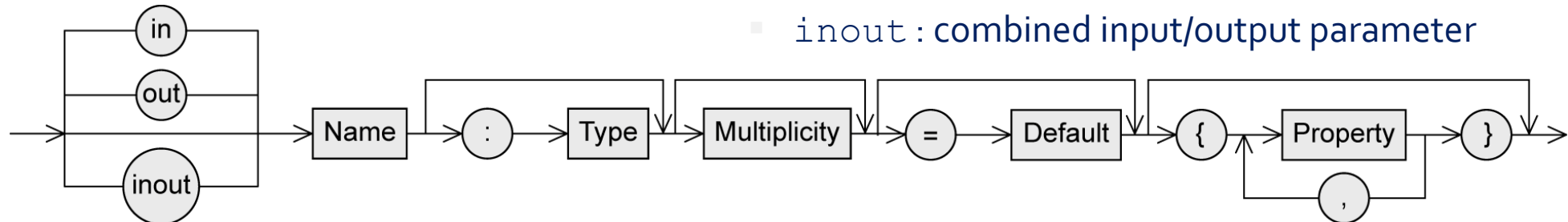
## ■ Propiedades predefinidas

- {readOnly} ... el valor no puede cambiar
- {unique} ... No se permiten duplicados
- {non-unique} ... Los valores pueden estar duplicados
- {ordered} ... determina el orden de los valores
- {unordered} ... no determina el orden de los valores
- Especificación del atributo
  - Set: {unordered, unique}
  - Multi-set: {unordered, non-unique}
  - Ordered set: {ordered, unique}
  - List: {ordered, non-unique}

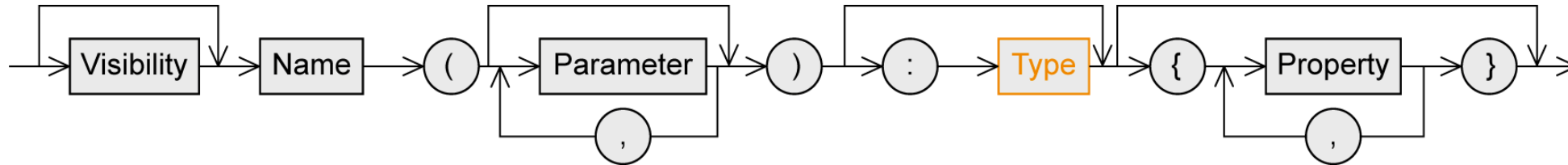
# 4. OPERACIONES. PARÁMETROS



- Notación similar a los atributos
- Dirección del parámetro
  - in ... parámetro de entrada, input
    - When the operation is used, a value is expected from this parameter
  - out ... parámetro de salida, output
    - After the execution of the operation, the parameter has adopted a new value
  - inout : combined input/output parameter



# 4. OPERACIONES. PARÁMETROS



- Tipo de valor de retorno.

Person
...
getName(out fn: String, out ln: String): void updateLastName(newName: String): boolean <u>getPersonsNumber(): int</u>



# 4. VARIABLE DE CLASE Y OPERACIÓN DE CLASE

Variable de instancia (= atributo de instancia): atributos definidos a nivel de instancia

Variable de clase (= atributo de clase, atributo estático)

Definido solo una vez por clase, es decir, lo comparten todas las instancias de la clase.

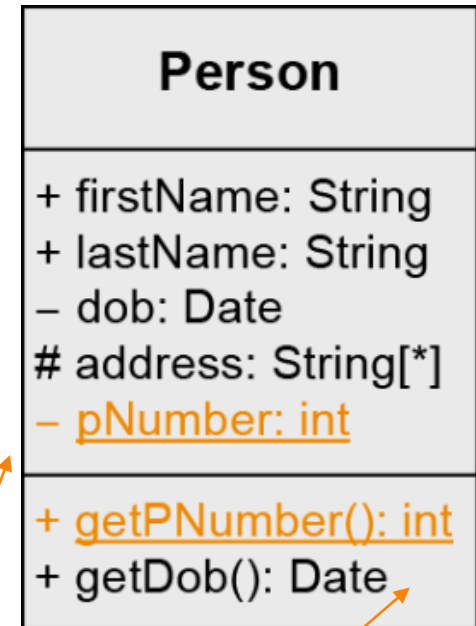
P.ej: contadores para el número de instancias de una clase, constantes, etc.

Operación de clase (= operación estática)

Se puede usar si no se creó una instancia de la clase correspondiente.

P.ej: constructores, operaciones de recuento, matemáticas. funciones (sin (x)), etc.

Notación: subrayar el nombre de la variable de clase / operación de clase



Variable  
de clase

Operación  
de clase

# 4. VARIABLE DE CLASE Y OPERACIÓN DE CLASE

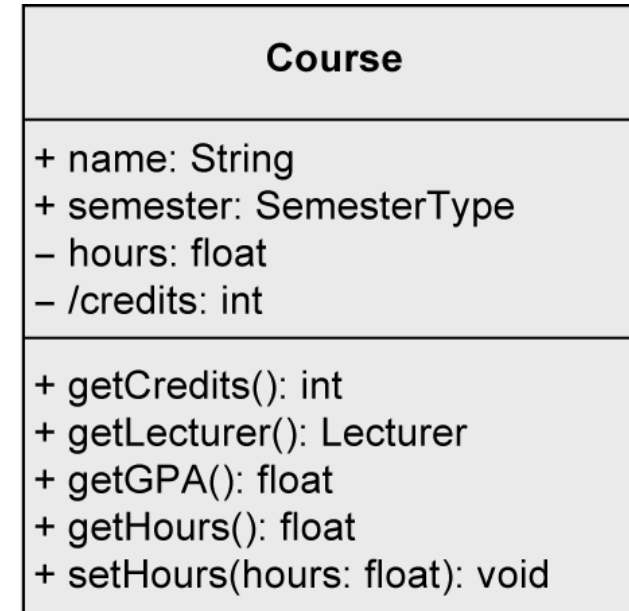
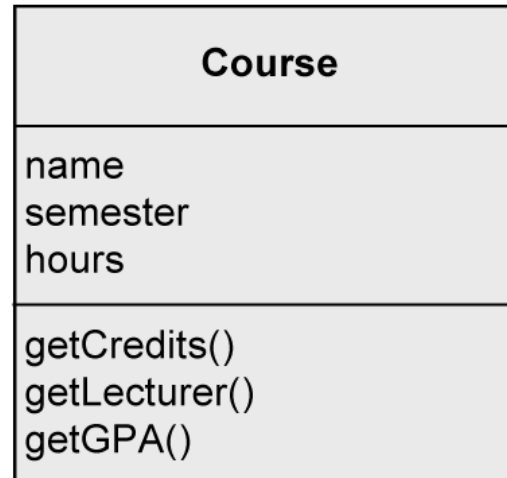
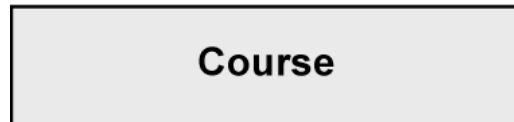
Variable  
de clase

Operación  
de clase

Person
+ firstName: String + lastName: String - dob: Date # address: String[*] <u>- pNumber: int</u>
+ <u>getPNumber(): int</u> + getDob(): Date

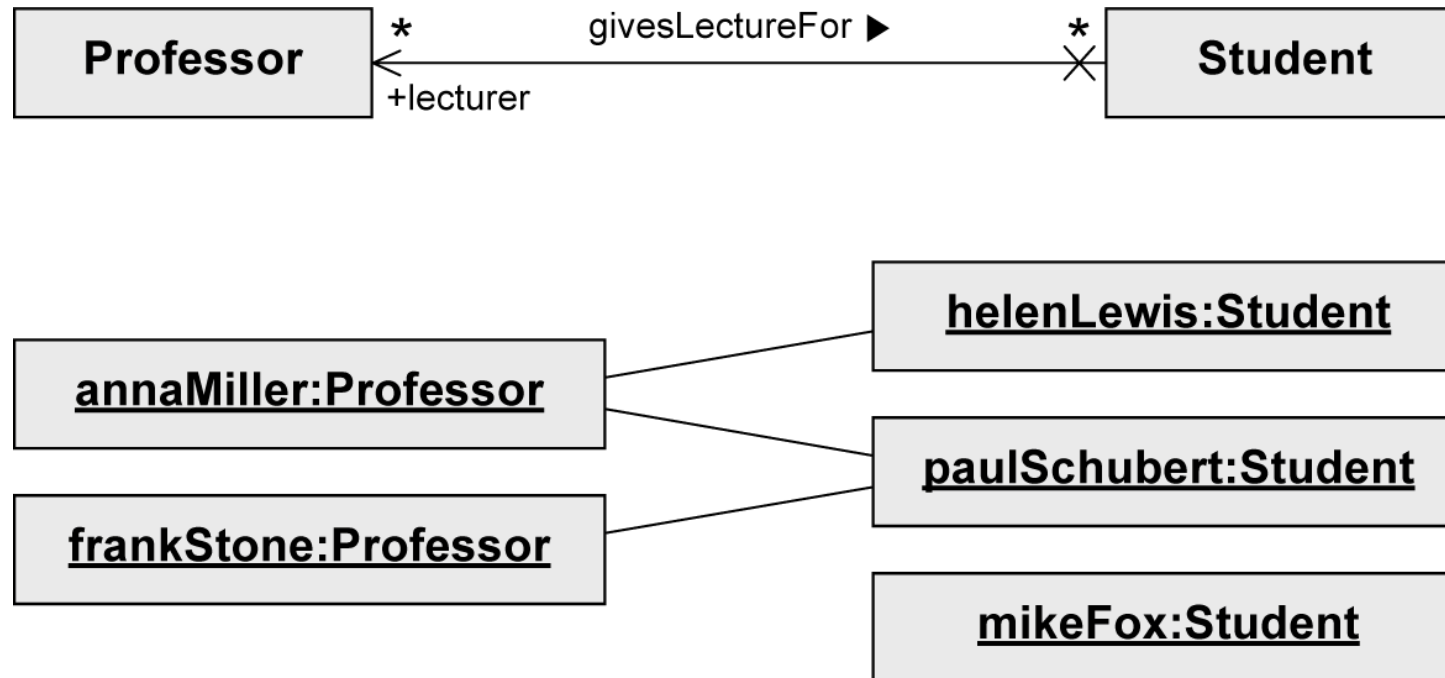
```
class Person {  
  
    public String firstName;  
    public String lastName;  
    private Date dob;  
    protected String[] address;  
    private static int pNumber;  
    public static int getPNumber()  
{...}  
    public Date getDob() {...}  
}
```

# 4. ESPECIFICACIÓN DE CLASES: DIFERENTES NIVELES DE DETALLE



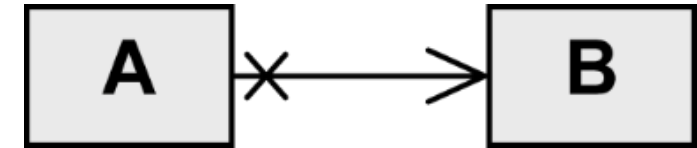
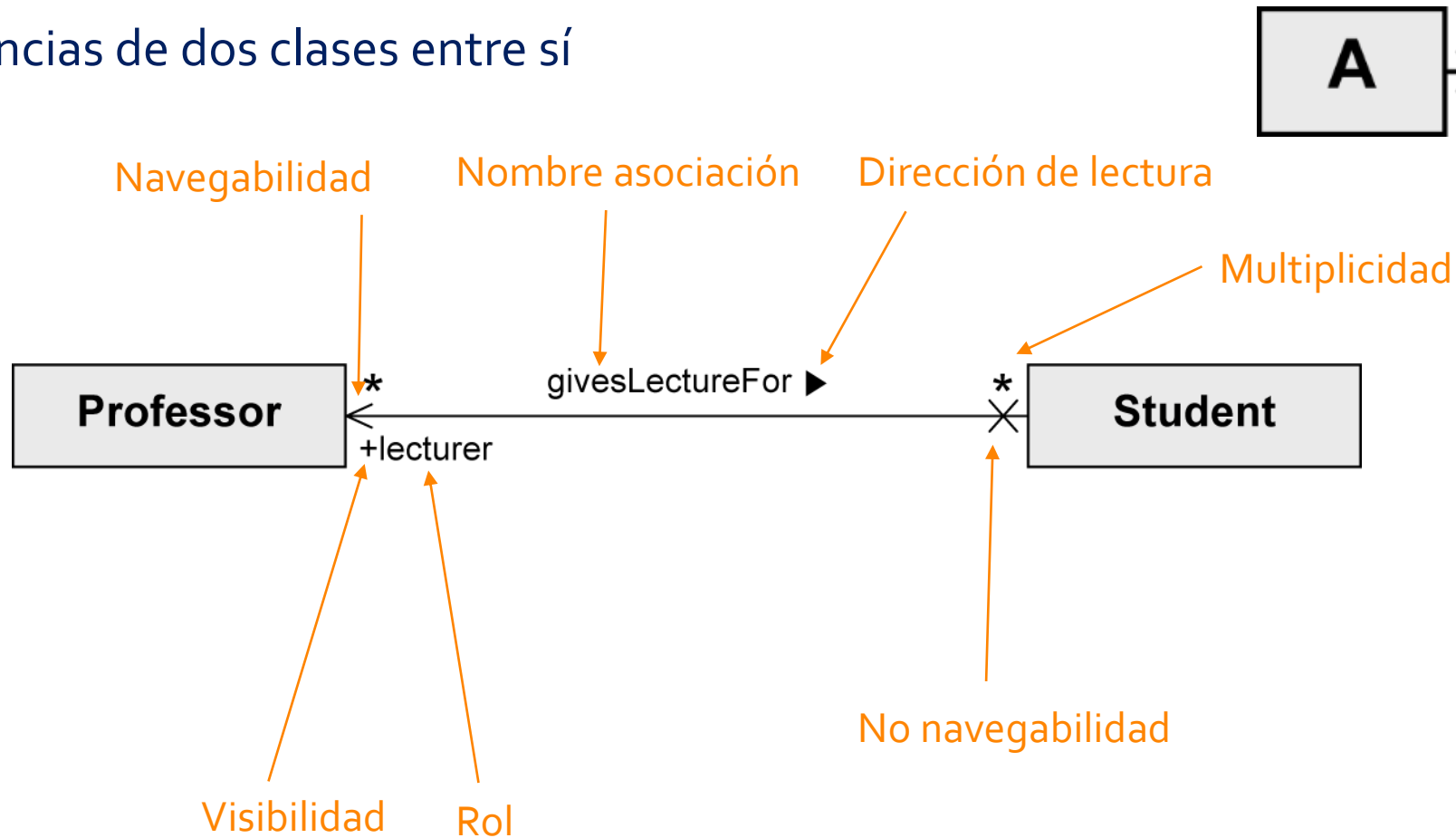
# 5. ASOCIACIÓN

Modela posibles relaciones entre instancias de clases:



# 5. ASOCIACIONES BINARIAS

Conecta instancias de dos clases entre sí



# 5. ASOCIACIONES BINARIAS. NAVEGABILIDAD

**Navegabilidad:** un objeto conoce sus objetos asociados y, por lo tanto, puede acceder a sus atributos y operaciones visibles

Indicado por punta de flecha abierta

No navegabilidad

Indicado por cruz

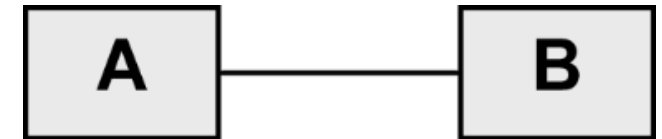
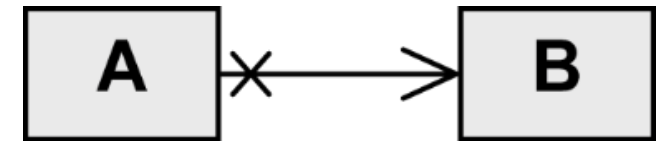
Ejemplo:

**A** puede acceder a los atributos y operaciones visibles de **B**

**B** no puede acceder a los atributos y operaciones de **A**

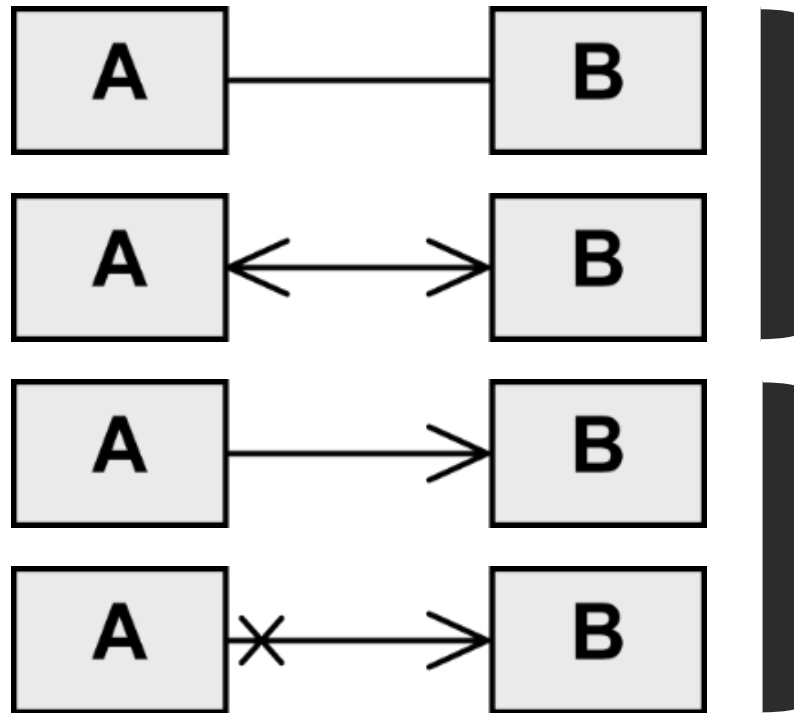
Navegabilidad indefinida

Se asume navegabilidad bidireccional

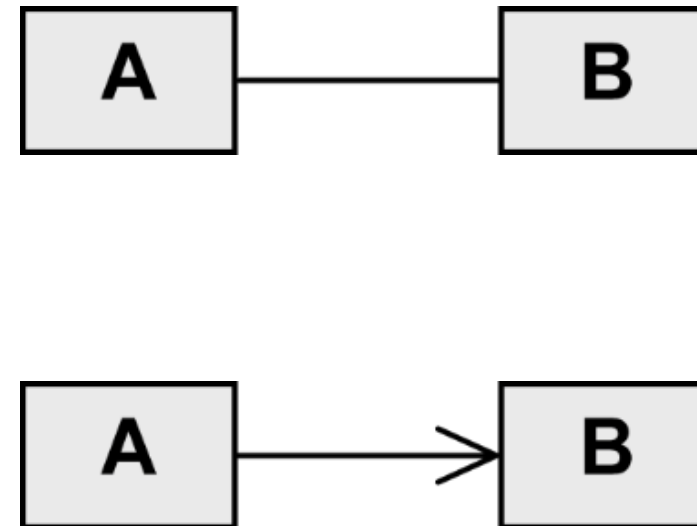


# 5. ASOCIACIONES BINARIAS. NAVEGABILIDAD

UML standard

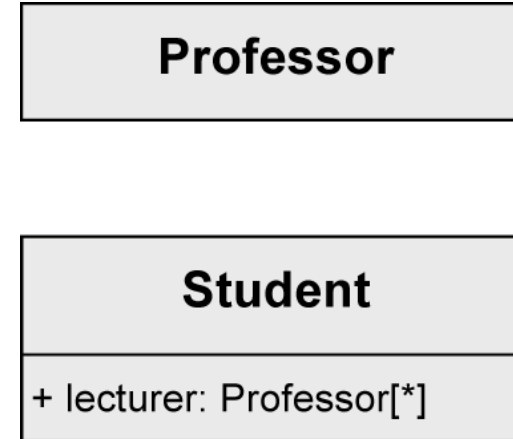
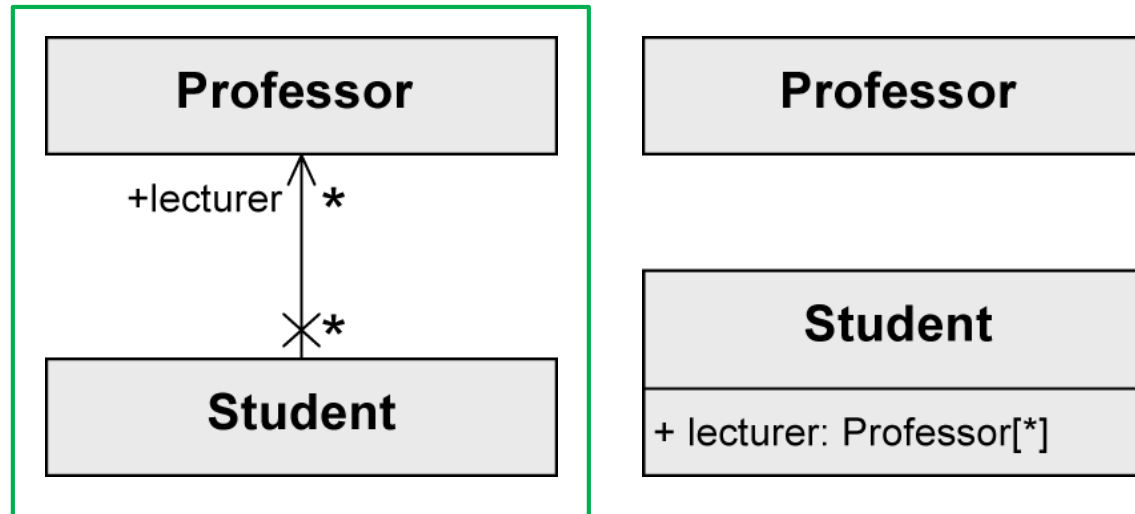


Best practice



# 5. ASOCIACIONES BINARIAS COMO ATRIBUTO

Preferible



Java notación

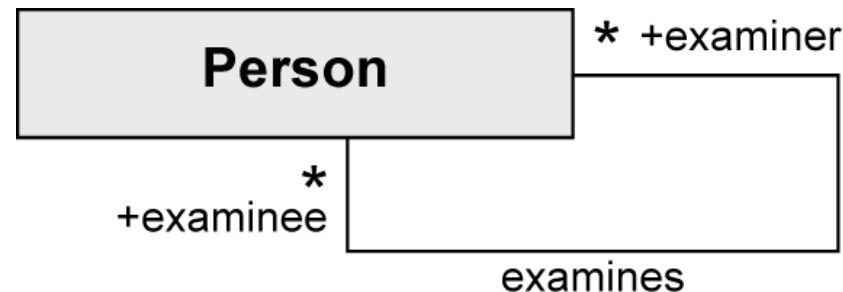
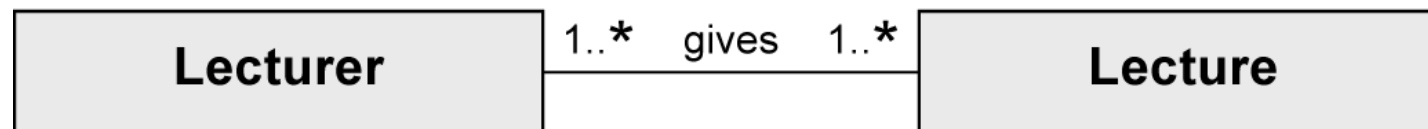
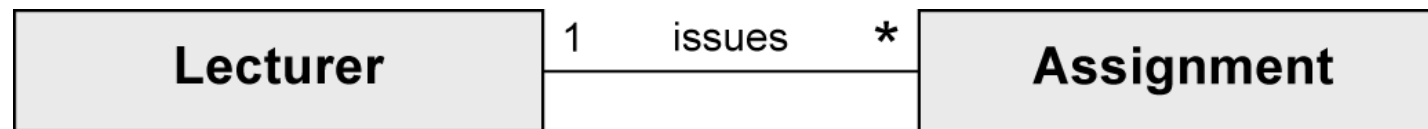
```
class Professor {...}

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



# 5. ASOCIACIONES BINARIAS COMO ATRIBUTO

**Multiplicidad:** número de objetos que pueden estar asociados con exactamente un objeto del lado opuesto.

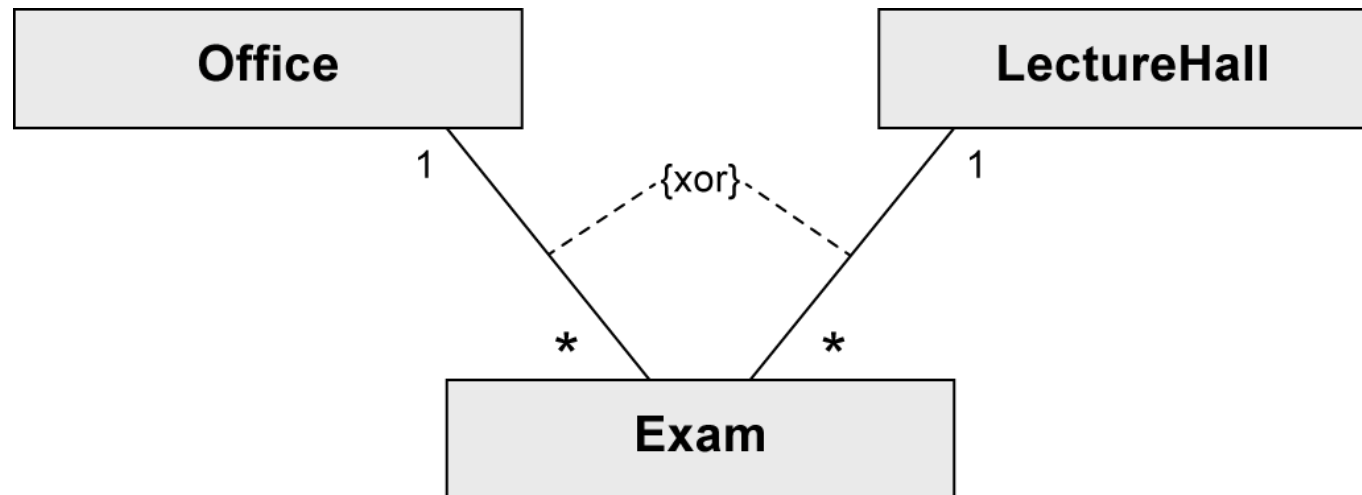
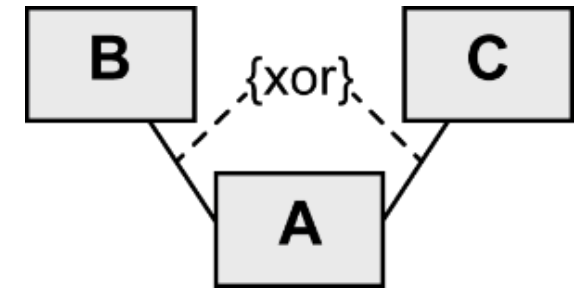


**Rol:** describe la forma en que un objeto está involucrado en una relación de asociación.

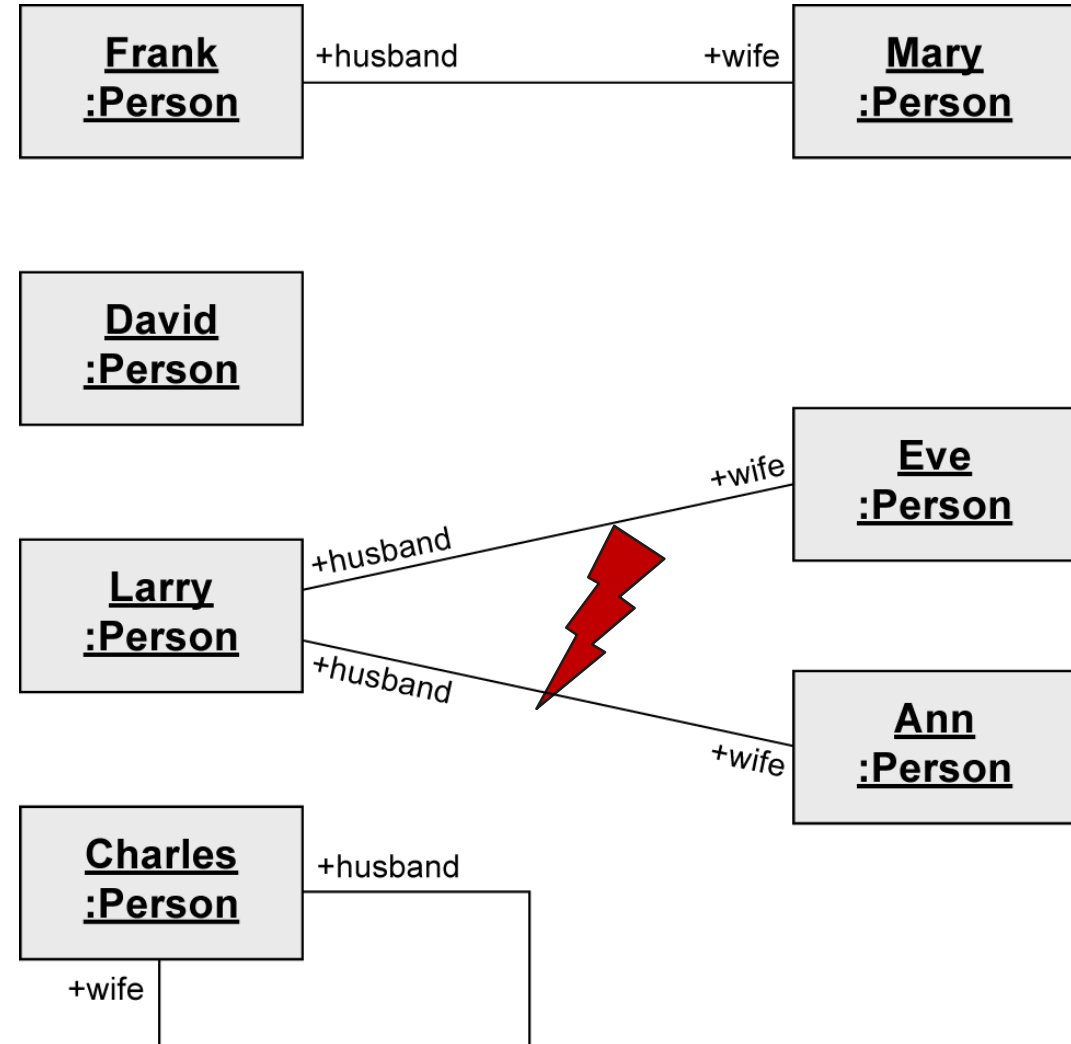
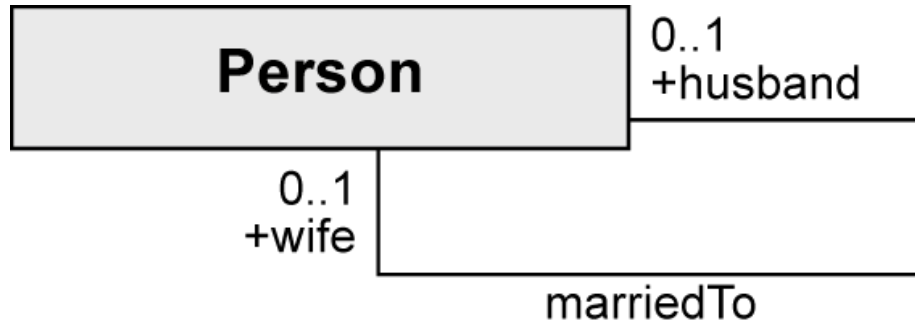
# 5. ASOCIACIONES BINARIAS COMO ATRIBUTO. XOR

## Restricción "o exclusiva"

Un objeto de clase A debe asociarse con un objeto de clase B o un objeto de clase C pero no con ambos.

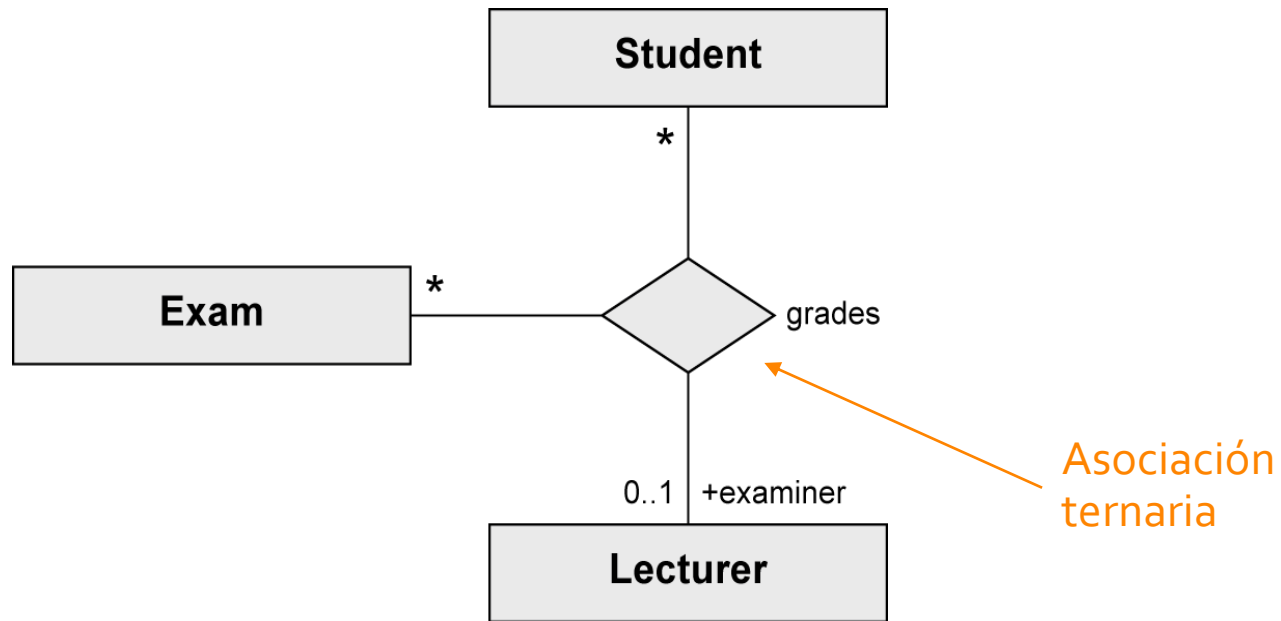
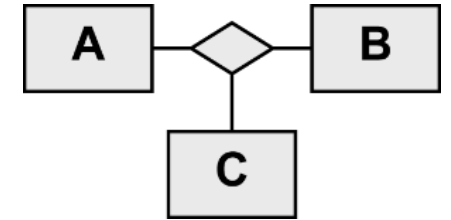


# 5. ASOCIACIÓN REFLEXIVA, UNARIA



# 5. ASOCIACIÓN N-ARIA

Más de dos objetos asociados están involucrados en la relación.  
No se definen indicaciones de navegación.



# 5. ASOCIACIÓN N-ARIA

Ejemplo:

$(\text{Student}, \text{Exam}) \rightarrow (\text{Lecturer})$

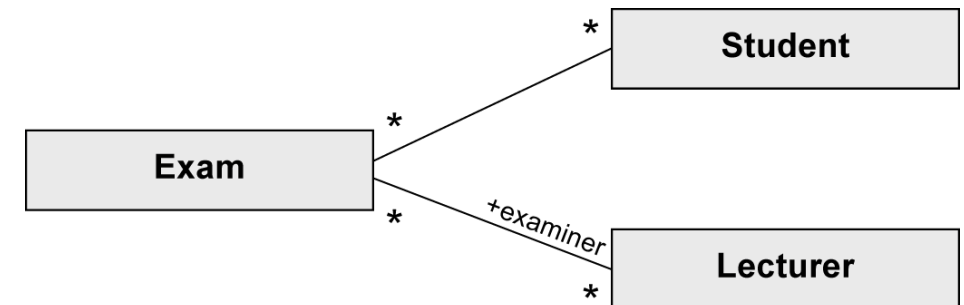
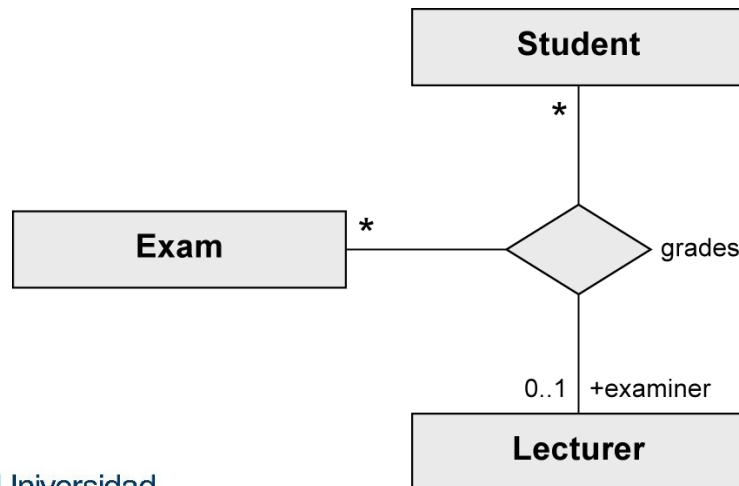
Un estudiante toma un examen con uno o ningún profesor

$(\text{Exam}, \text{Lecturer}) \rightarrow (\text{Student})$

Cualquier número de estudiantes puede realizar un examen con un profesor.

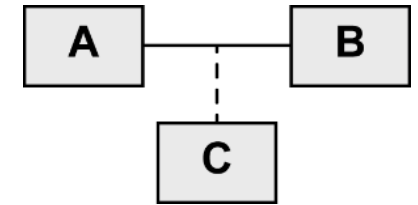
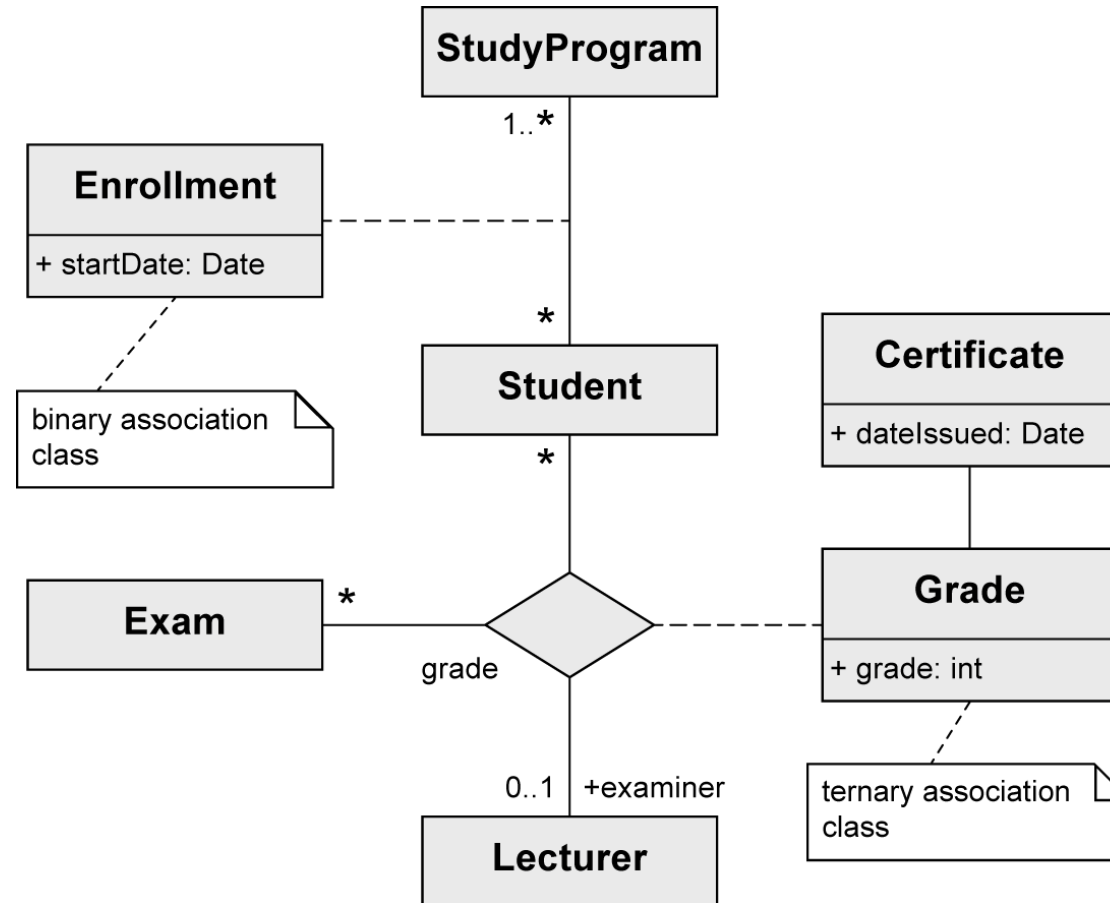
$(\text{Student}, \text{Lecturer}) \rightarrow (\text{Exam})$

Un estudiante puede ser calificado por un profesor para cualquier número de exámenes.



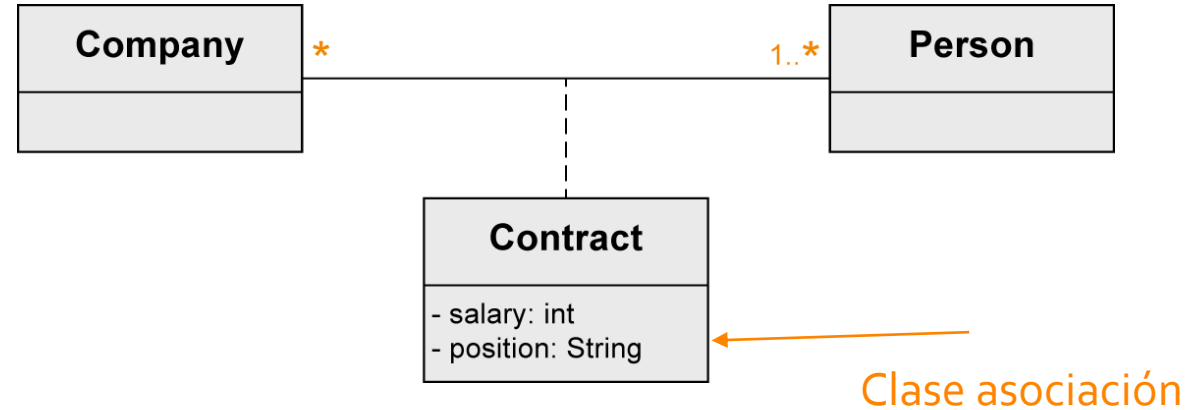
# 5. CLASE ASOCIACIÓN

Asignar atributos a la relación entre clases en lugar de a una clase en sí:

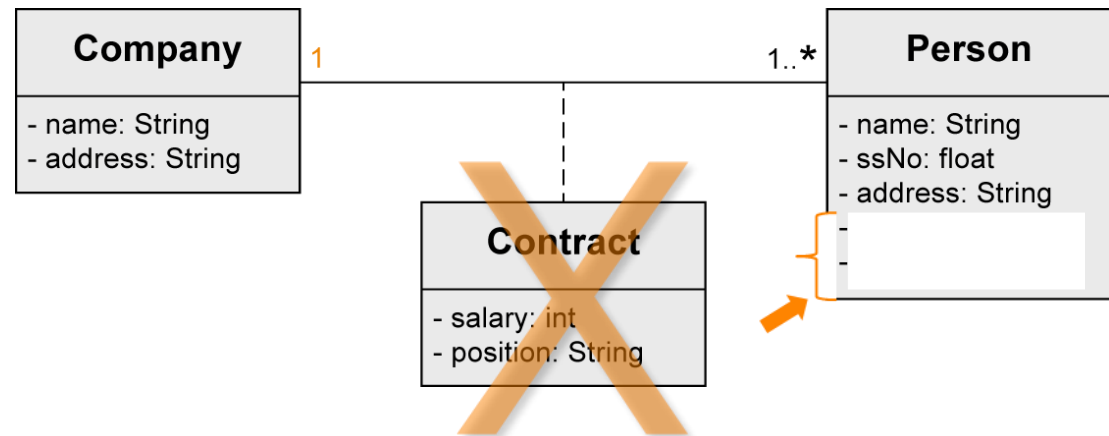


# 5. CLASE ASOCIACIÓN

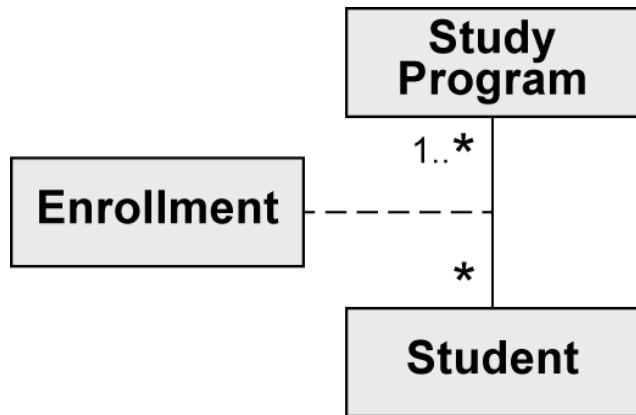
Necesario al modelar asociaciones n: m



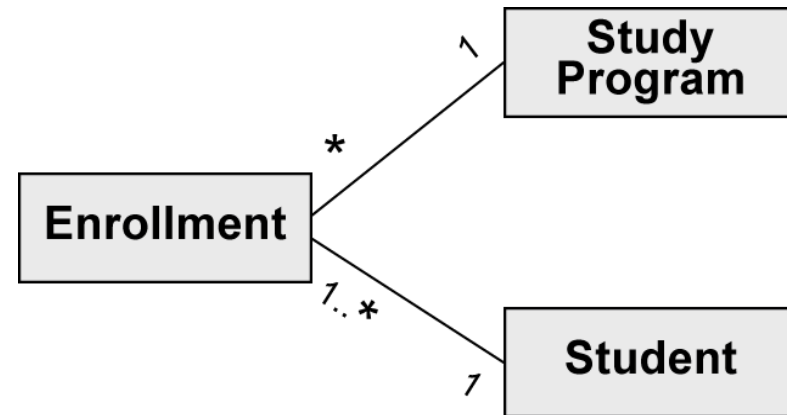
Con 1: 1 o 1: n posible pero no necesario



# 5. CLASE ASOCIACIÓN VS CLASE NORMAL



Un estudiante puede inscribirse en un "Study Program" concreto **cada vez**



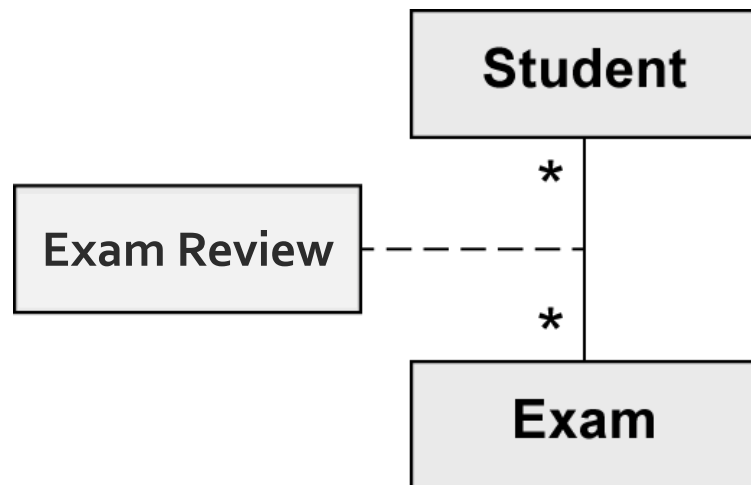
Un estudiante puede tener **múltiples** Inscripciones para un Programa de Estudios (el mismo)



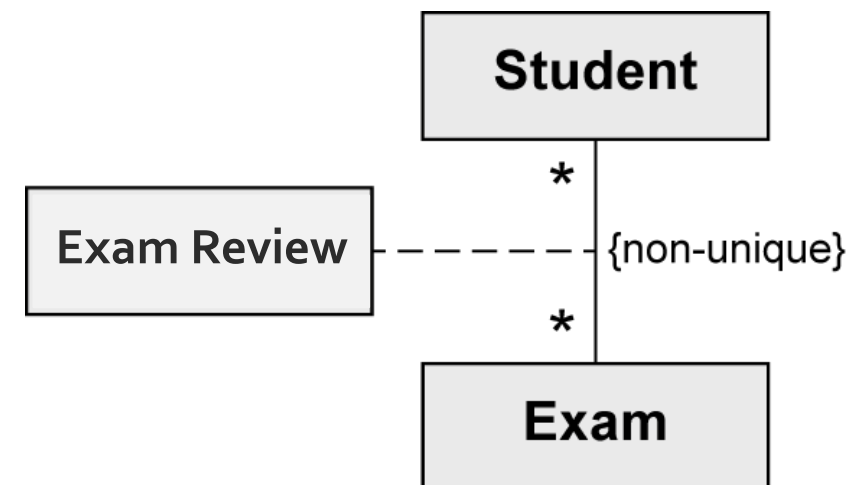
# 5. CLASE ASOCIACIÓN. UNIQUE. NON UNIQUE

**Default:** sin duplicados

**Non unique:** duplicados permitidos

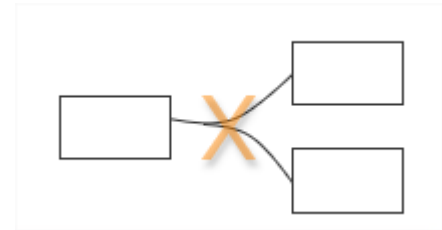
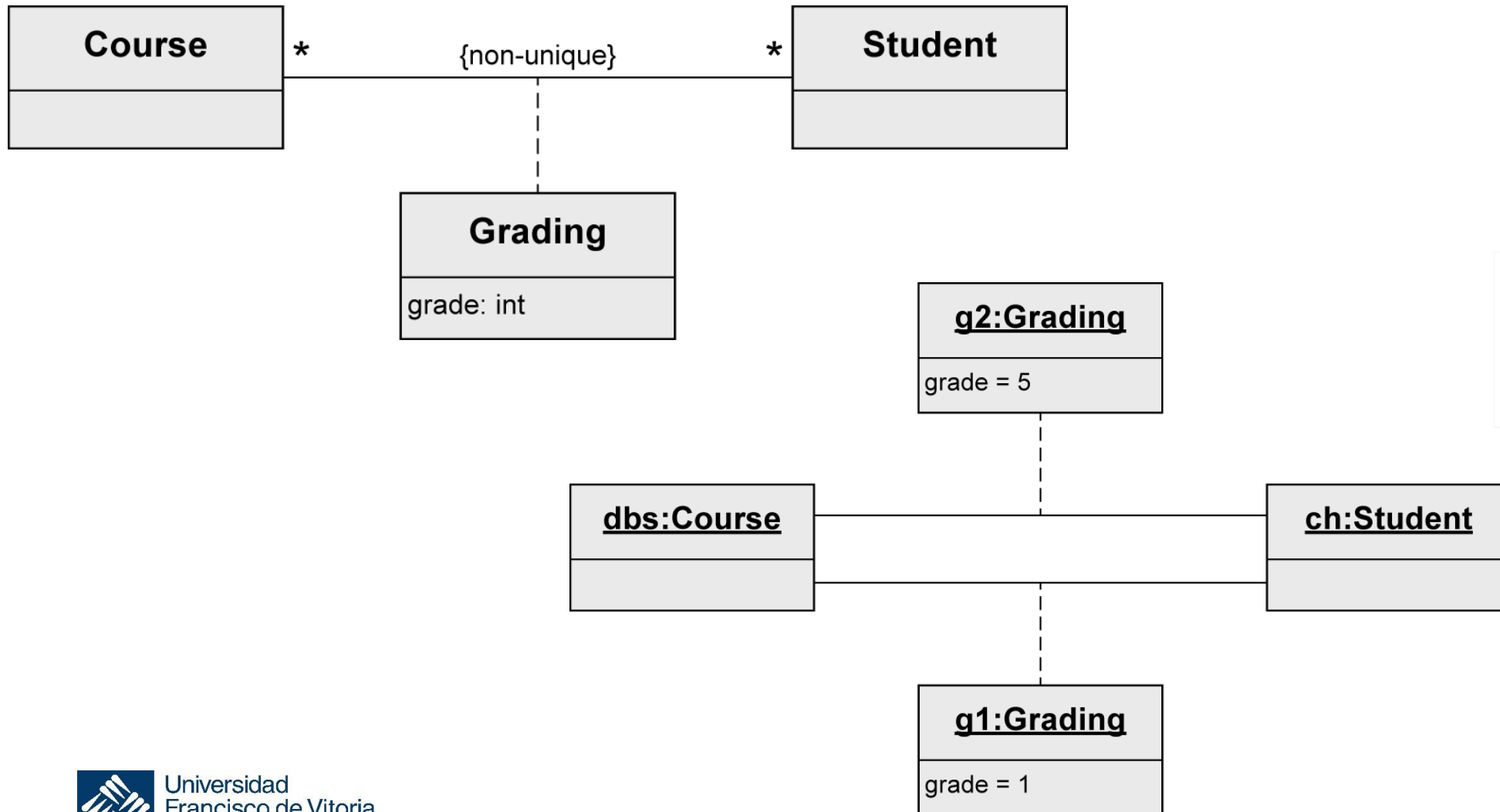


A un estudiante solo se le puede otorgar una revisión de examen para un examen específico **cada vez**



Un estudiante puede tener **más de una revisión de examen** para un examen específico.

# 5. CLASE ASOCIACIÓN. UNIQUE. NON UNIQUE



# 5. AGREGACIÓN

Forma especial de asociación.

Se usa para expresar que una clase es parte de otra clase.

Propiedades de la asociación de agregación:

Transitivo: si B es parte de A y C es parte de B, C también es parte de A.

Asimétrico: no es posible que A sea parte de B y B sea parte de A a la vez.

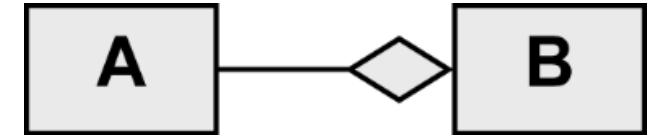
Dos tipos:

Agregación compartida.

Composición.

# 5. AGREGACIÓN COMPARTIDA

Expresa una pertenencia débil de las partes a un todo  
= Las partes también existen independientemente del todo



La multiplicidad en el extremo de la agregación puede ser  $> 1$   
= Un elemento puede formar parte de varios otros elementos simultáneamente

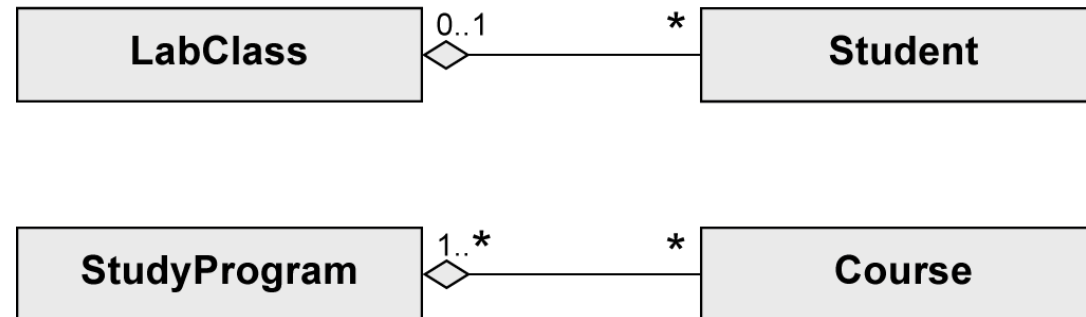
Abarca un gráfico acíclico dirigido

Sintaxis: diamante hueco en el extremo de agregación

Ejemplo:

**Student** es parte de **LabClass**

**Course** es parte de **StudyProgram**



# 5. COMPOSICIÓN

Dependencia de existencia entre el objeto compuesto y sus partes

Una parte solo puede estar contenida como máximo en un objeto compuesto en un momento específico en el tiempo

Multiplicidad en el extremo de agregación máx. 1

-> Los objetos compuestos forman un árbol



Si se elimina el objeto compuesto, también se eliminan sus partes.

Sintaxis: diamante macizo en el extremo de agregación

Example: **Beamer** es parte de **LectureHall** es parte de **Building**

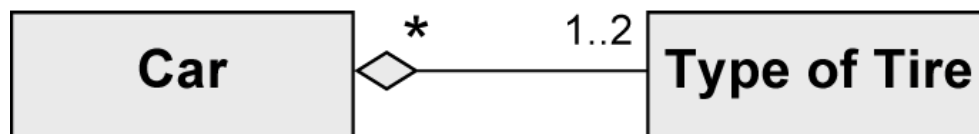
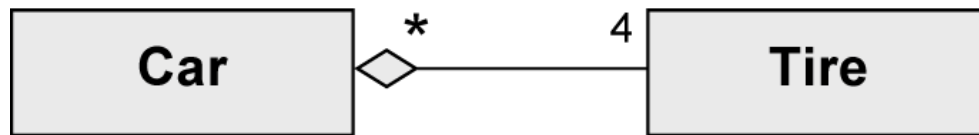
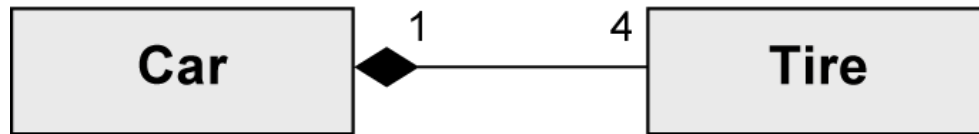
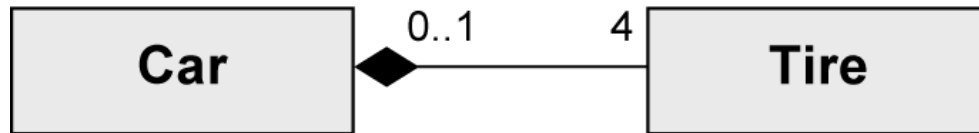
Si se elimina el Edificio,  
también se elimina el LectureHall

El Proyector puede existir sin Sala,  
pero si está contenido en Sala mientras se elimina,  
el Proyector también se elimina



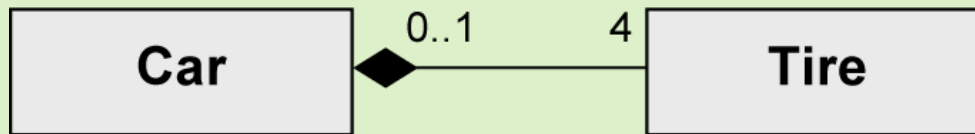
# 5. AGREGACIÓN Y COMPOSICIÓN COMPARTIDAS

Qué modelo aplica



# 5. AGREGACIÓN Y COMPOSICIÓN COMPARTIDAS

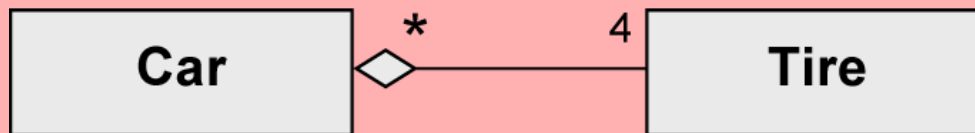
Qué modelo aplica



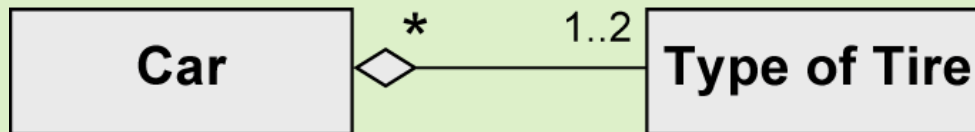
Un neumático puede existir sin un coche. Un neumático pertenece a un coche como máximo.



Un neumático **no puede existir** sin un coche.



Un neumático puede pertenecer a varios coches



Un automóvil tiene uno o dos tipos de neumáticos. Varios coches pueden tener el mismo tipo de neumáticos.

-----Yes-----  
-----No-----  
-----Yes-----

# 5. GENERALIZACIÓN

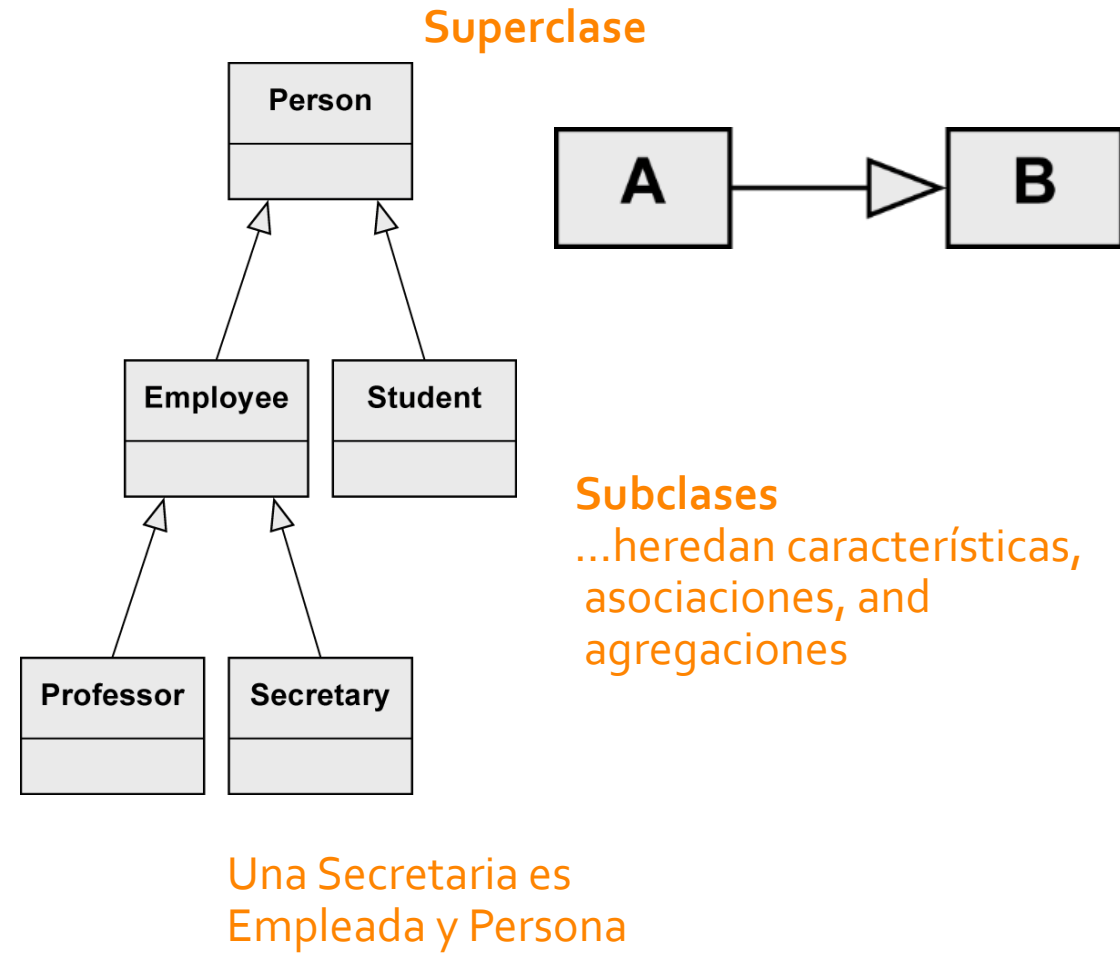
Las características (atributos y operaciones), asociaciones y agregaciones que se especifican para una clase general (superclase) se heredan en sus subclases.

Cada instancia de una subclase es simultáneamente una instancia indirecta de la superclase.

La subclase hereda todas las características, asociaciones y agregaciones de la superclase, excepto las privadas.

La subclase puede tener más características, asociaciones y agregaciones.

Las generalizaciones son transitivas.





# 5. GENERALIZACIÓN. CLASE ABSTRACTA

Se usa para **resaltar características comunes de sus subclases**.

Se utiliza para garantizar que no haya instancias directas de la superclase.

Solo se pueden instanciar sus subclases no abstractas.

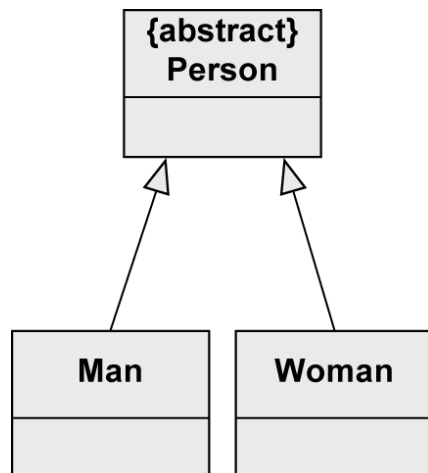
Útil en el contexto de relaciones de generalización.

Notación: palabra clave {abstract} o nombre de la clase en cursiva.

{abstract}  
A

{abstract}  
Person

*Person*



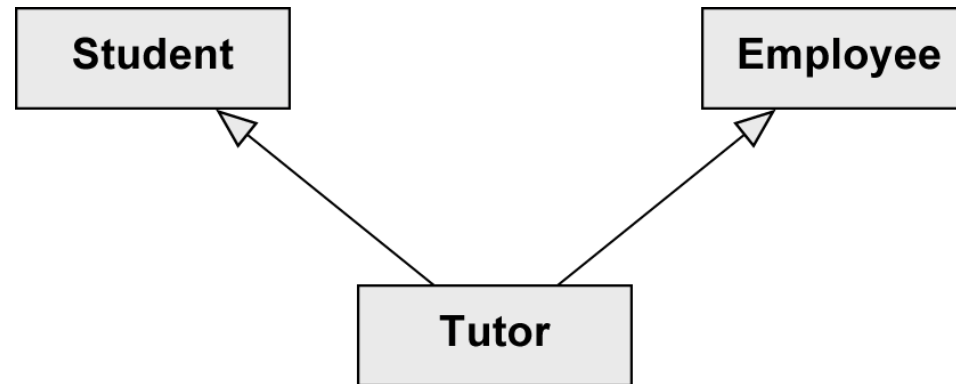
No existe el objeto Persona como tal

Dos tipos de persona: Hombre y Mujer

# 5. GENERALIZACIÓN. HERENCIA MÚLTIPLE

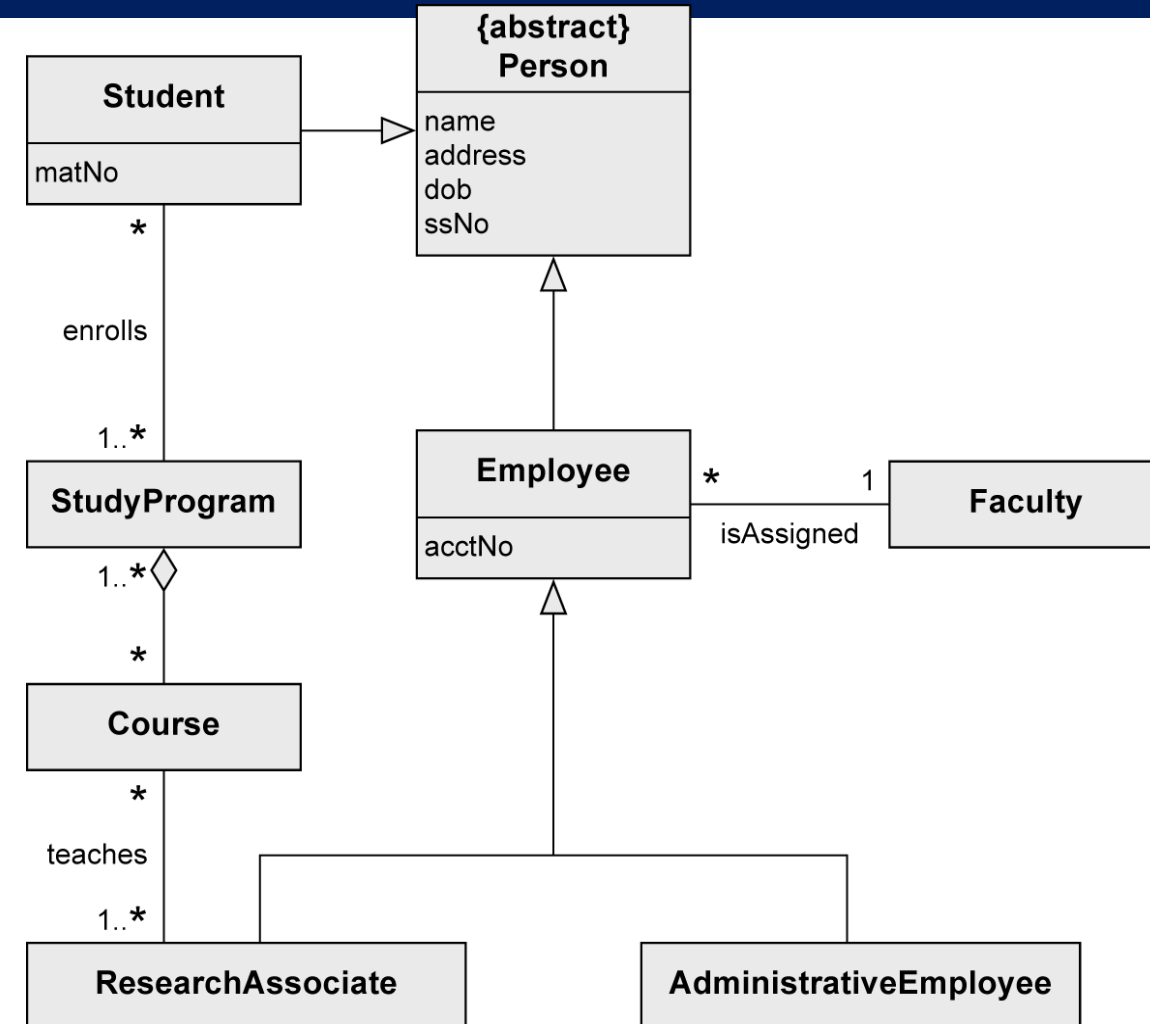
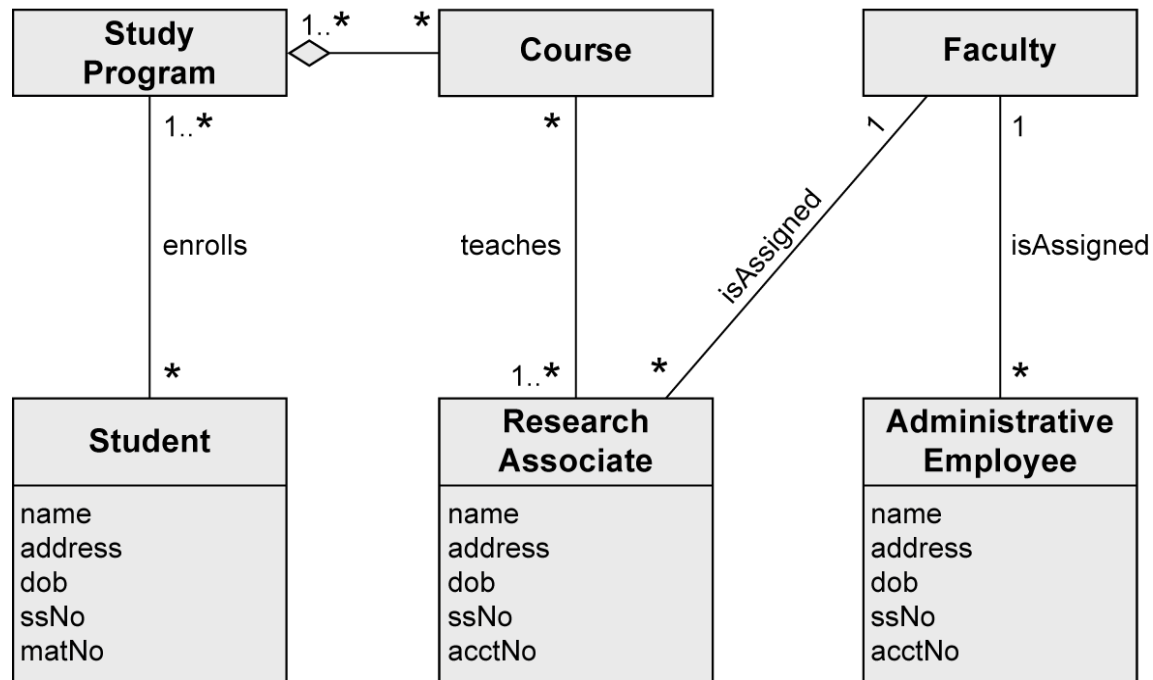
UML permite la herencia múltiple.  
Una clase puede tener varias superclases.

Ejemplo:



Un Tutor es tanto Empleado como Estudiante

# 5. GENERALIZACIÓN. CON Y SIN GENERALIZACIÓN



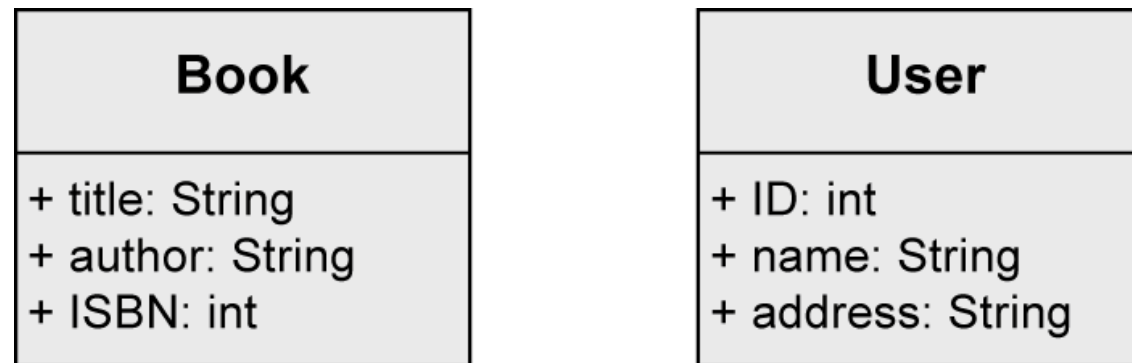
# 5. CREAR UN DIAGRAMA DE CLASE

No es posible extraer por completo clases, atributos y asociaciones de un texto en lenguaje natural automáticamente.

Directrices:

- Los sustantivos a menudo indican clases
- Los adjetivos indican valores de atributos
- Los verbos indican operaciones

Ejemplo: *El sistema de gestión de la biblioteca almacena a los usuarios con su ID, nombre y dirección únicos, así como libros con su título, autor y número de ISBN. Ann Foster quiere usar la biblioteca.*



Pregunta: ¿Qué pasa con Ann Foster?

# 6.EJEMPLO

- Una universidad consta de múltiples facultades que se componen de varios institutos. Cada facultad y cada instituto tiene un nombre. Se conoce una dirección para cada instituto.
- Cada facultad está dirigida por un decano, que es un empleado de la universidad.
- Se conoce el número total de empleados. Los empleados tienen un número de seguro social, un nombre y una dirección de correo electrónico. Existe una distinción entre personal de investigación y administrativo.
- Los investigadores asociados están asignados a al menos un instituto. Se conoce el campo de estudio de cada investigador asociado. Además, los investigadores asociados pueden participar en proyectos durante un cierto número de horas, y se conocen el nombre, la fecha de inicio y la fecha de finalización de los proyectos. Algunos investigadores asociados llevan a cabo cursos. Entonces se les llama conferenciantes.
- Los cursos tienen un número (ID) único, un nombre y una duración semanal en horas.

# 6.EJEMPLO. PASO 1. DEFINICIÓN DE CLASES

Una universidad consta de múltiples facultades que se componen de varios institutos. Cada facultad y cada instituto tiene un nombre. Se conoce una dirección para cada instituto.

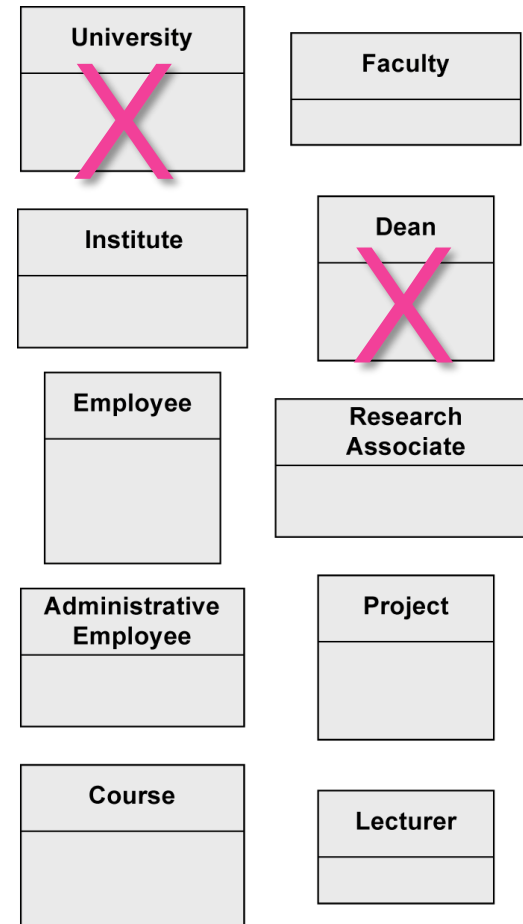
Cada facultad está dirigida por un decano, que es un empleado de la universidad.

Se conoce el número total de empleados. Los empleados tienen un número de seguro social, un nombre y una dirección de correo electrónico. Existe una distinción entre personal de investigación y administrativo.

Los investigadores asociados están asignados a al menos un instituto. Se conoce el campo de estudio de cada investigador asociado. Además, los investigadores asociados pueden participar en proyectos durante un cierto número de horas, y se conocen el nombre, la fecha de inicio y la fecha de finalización de los proyectos. Algunos investigadores asociados imparten cursos. Entonces se les llama profesores.

Los cursos tienen un número (ID) único, un nombre y una duración semanal en horas.

## Modelamos el sistema "Universidad"



Decano no tiene más atributos que otro empleado

# 6.EJEMPLO. PASO 2. DEFINICIÓN DE ATRIBUTOS

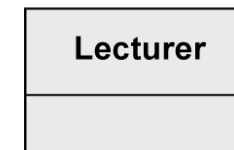
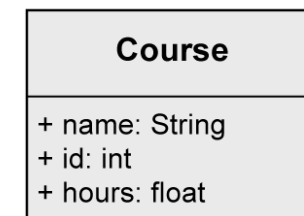
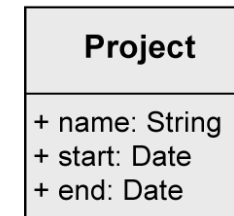
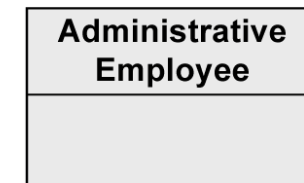
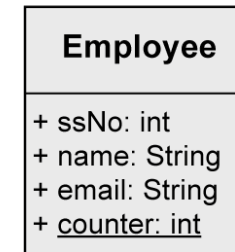
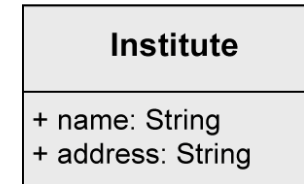
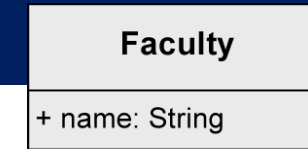
Una universidad consta de múltiples facultades que se componen de varios institutos. Cada facultad y cada instituto tiene un nombre. Se conoce una dirección para cada instituto.

Cada facultad está dirigida por un decano, que es un empleado de la universidad.

Se conoce el número total de empleados. Los empleados tienen un número de seguridad social, un nombre y una dirección de correo electrónico. Existe una distinción entre personal de investigación y administrativo.

Los investigadores asociados están asignados a al menos un instituto. Se conoce el campo de estudio de cada investigador asociado. Además, los investigadores asociados pueden participar en proyectos durante un cierto número de horas, y se conocen el nombre, la fecha de inicio y la fecha de finalización de los proyectos. Algunos investigadores asociados imparten cursos. Entonces se les llama profesores.

Los cursos tienen un número único (ID), un nombre y una duración semanal en horas.



# 6.EJEMPLO. PASO 2. DEFINICIÓN DE RELACIONES

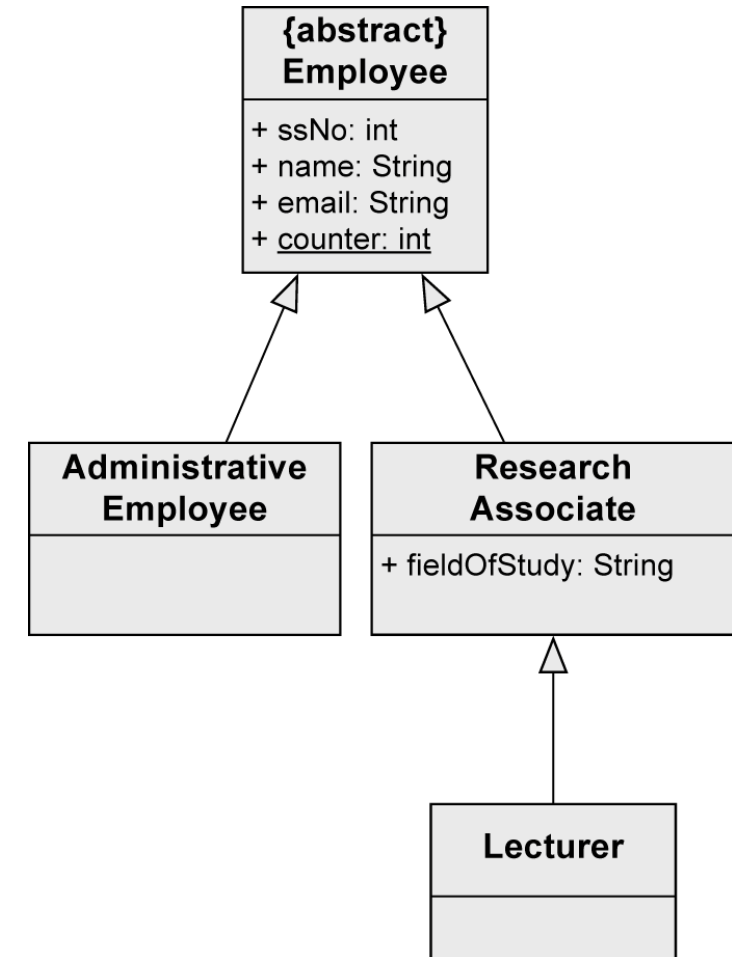
Tres tipos de relaciones:

- Asociación
- Generalización
- Agregación

Indicación de una generalización

"Hay una distinción entre personal de investigación y administrativo".

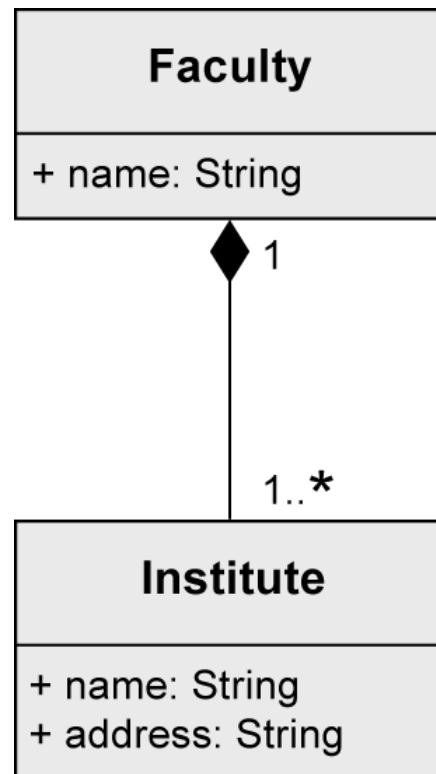
"Algunos investigadores asociados imparten cursos. Entonces se les llama profesores".





# 6.EJEMPLO. PASO 2. DEFINICIÓN DE RELACIONES

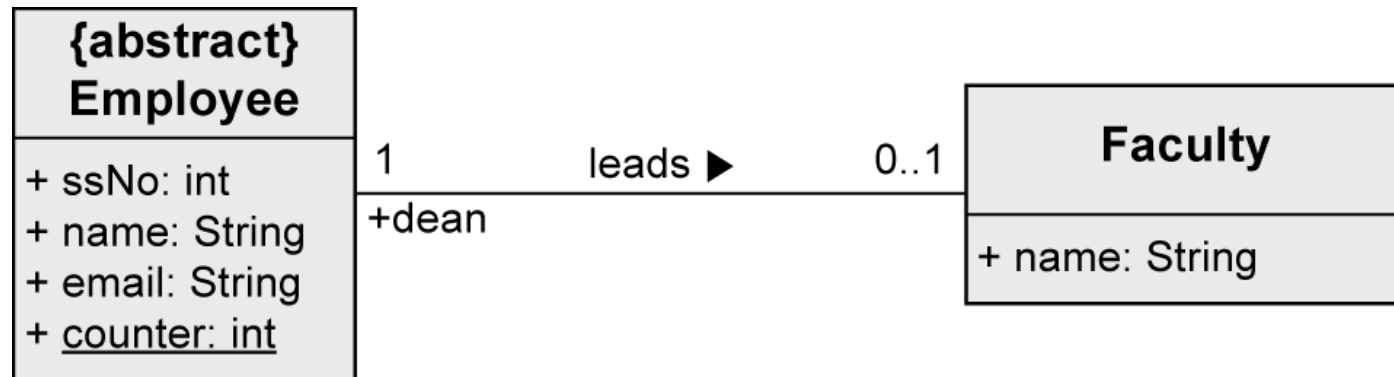
"Una universidad consta de múltiples facultades que se componen de varios institutos".



Composición para mostrar la dependencia de la existencia.

# 6.EJEMPLO. PASO 2. DEFINICIÓN DE RELACIONES

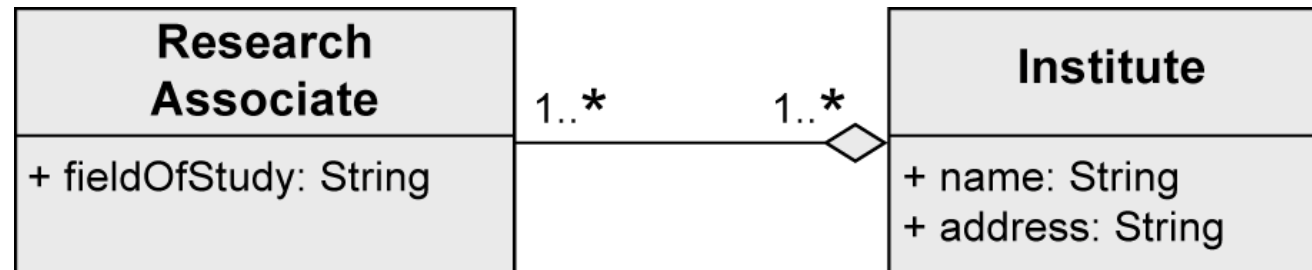
“Cada facultad está dirigida por un decano, que es empleado de la universidad”



En la relación de “dirección”, el empleado asume el rol de decano.

# 6.EJEMPLO. PASO 2. DEFINICIÓN DE RELACIONES

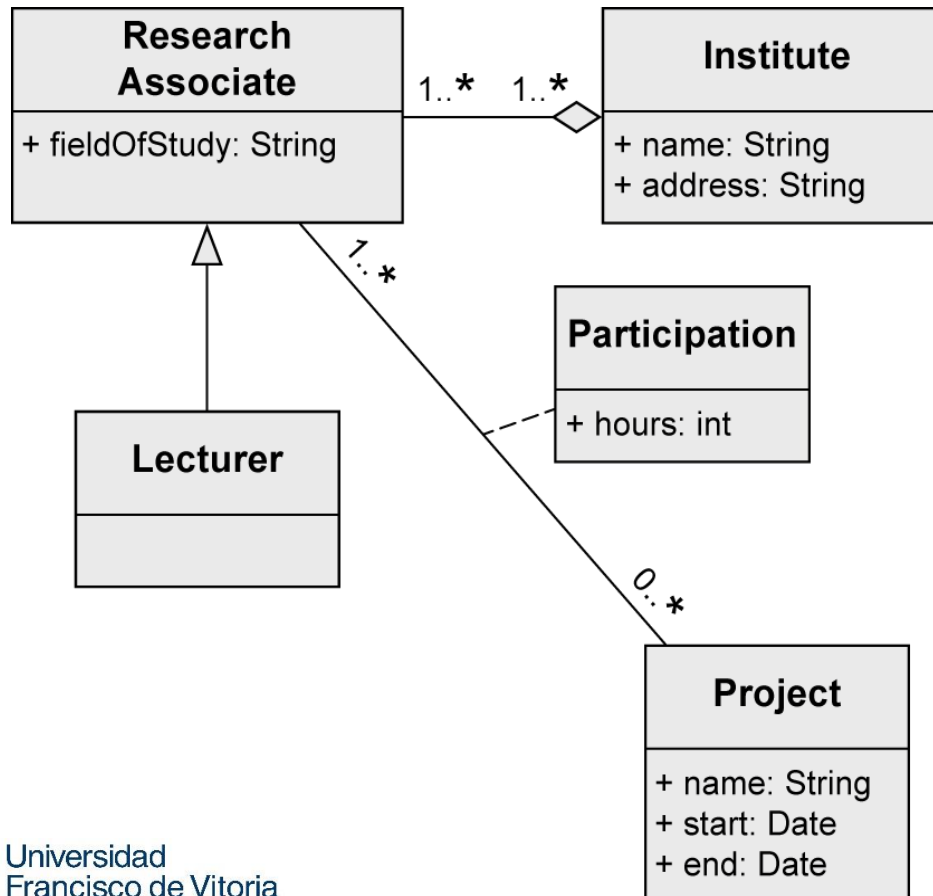
"Los investigadores asociados están asignados al menos a un instituto".



**Agregación compartida** para mostrar que "Investigadores" forman parte de un Instituto, pero no existe dependencia

# 6.EJEMPLO. PASO 2. DEFINICIÓN DE RELACIONES

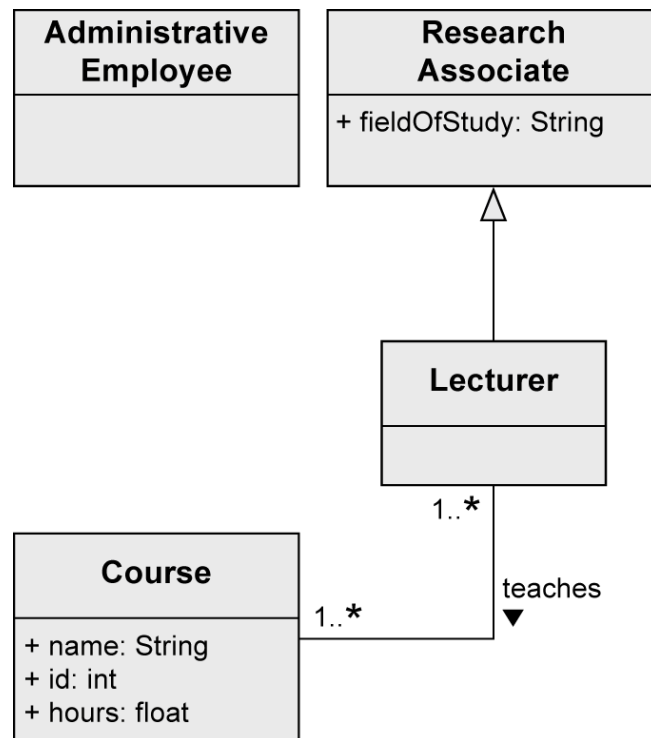
“Además, los investigadores asociados pueden participar en proyectos durante un cierto número de horas”.



La clase asociación permite almacenar el número de horas por cada Proyecto de cada Investigador asociado

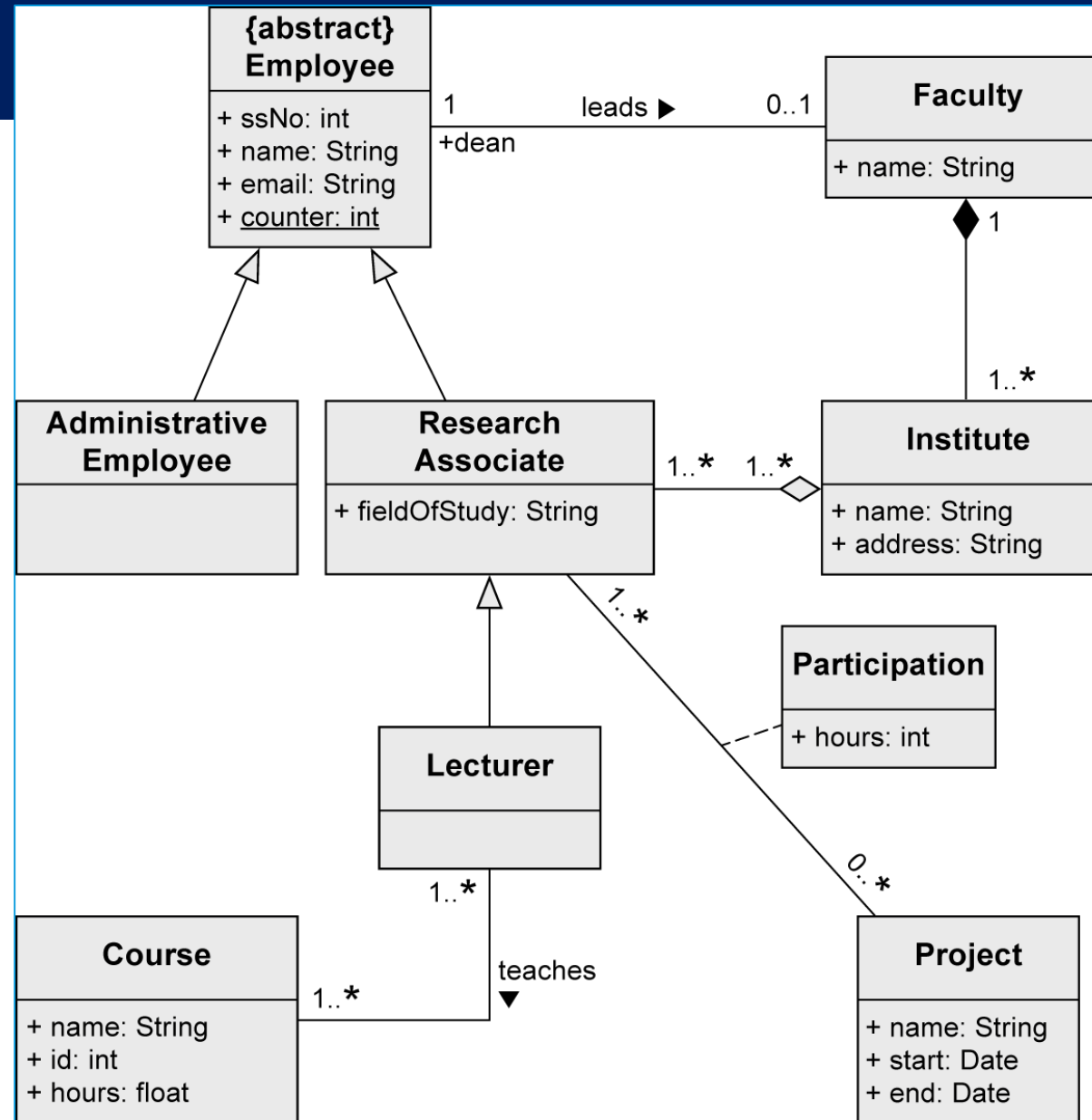
# 6.EJEMPLO. PASO 2. DEFINICIÓN DE RELACIONES

"Algunos investigadores asociados imparten cursos. Entonces se les llama profesores".

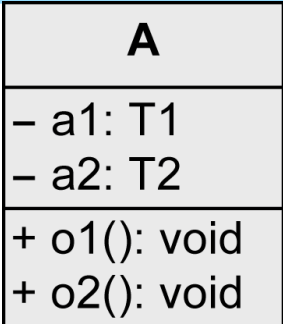
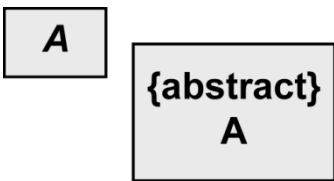
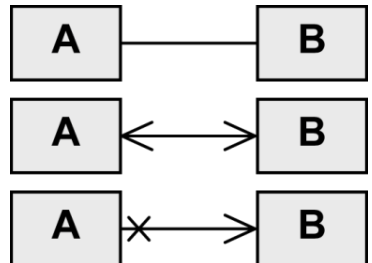


Profesor hereda todas las características, de Investigador asociado. Además, un profesor tiene una asociación: imparte el curso.

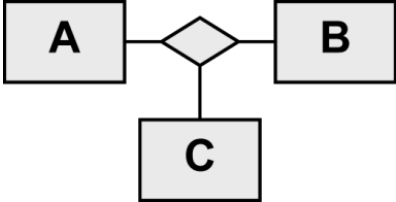
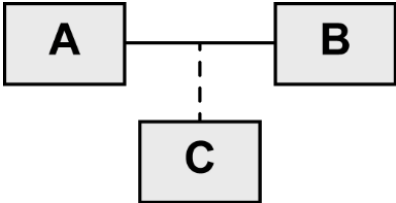
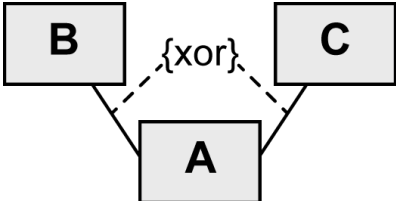
# 6.EJEMPLO. DIAGRAMA DE CLASE COMPLETO



# 7. NOTACIÓN. RESUMEN

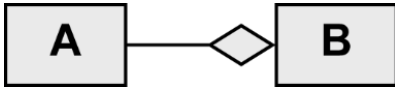

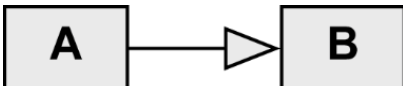

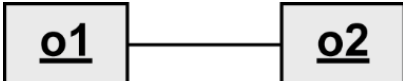
Nombre	Notación	Descripción
<b>Clase</b>	 <pre>classDiagram     class A {         -a1: T1         -a2: T2         +o1(): void         +o2(): void     }     </pre>	Descripción de la estructura y comportamiento de un conjunto de objetos.
<b>Clase abstracta</b>	 <pre>classDiagram     class A {         &lt;&lt;abstract&gt;&gt;     }     </pre>	Clase que no se puede instanciar
<b>Asociación</b>	 <pre>classDiagram     class A     class B     A -- B     A &lt;--&gt; B     A --&gt; "*" B     </pre>	Relación entre clases: navegabilidad no especificada, navegable en ambas direcciones, no navegable en una dirección

# 7. NOTACIÓN. RESUMEN

Nombre	Notación	Descripción
<b>Asociación N-aria</b>		Relación entre n clases. En el ejemplo 3 clases.
<b>Clase asociación</b>		Descripción más detallada de una asociación
<b>Relación XOR</b>		Un objeto de C está en relación con un objeto de A o con un objeto de B pero no con ambos



# 7. NOTACIÓN. RESUMEN

Nombre	Notación	Descripción
<b>Agregación compartida</b>		Relación parte-todo (A es parte de B)
<b>Agregación fuerte= Composición</b>		Relación parte-todo dependientes de la existencia (A es parte de B)
<b>Generalización</b>		Relación de herencia (A hereda de B)
<b>Objeto</b>		Instancia de una clase
<b>Relación</b>		Relación entre objetos