

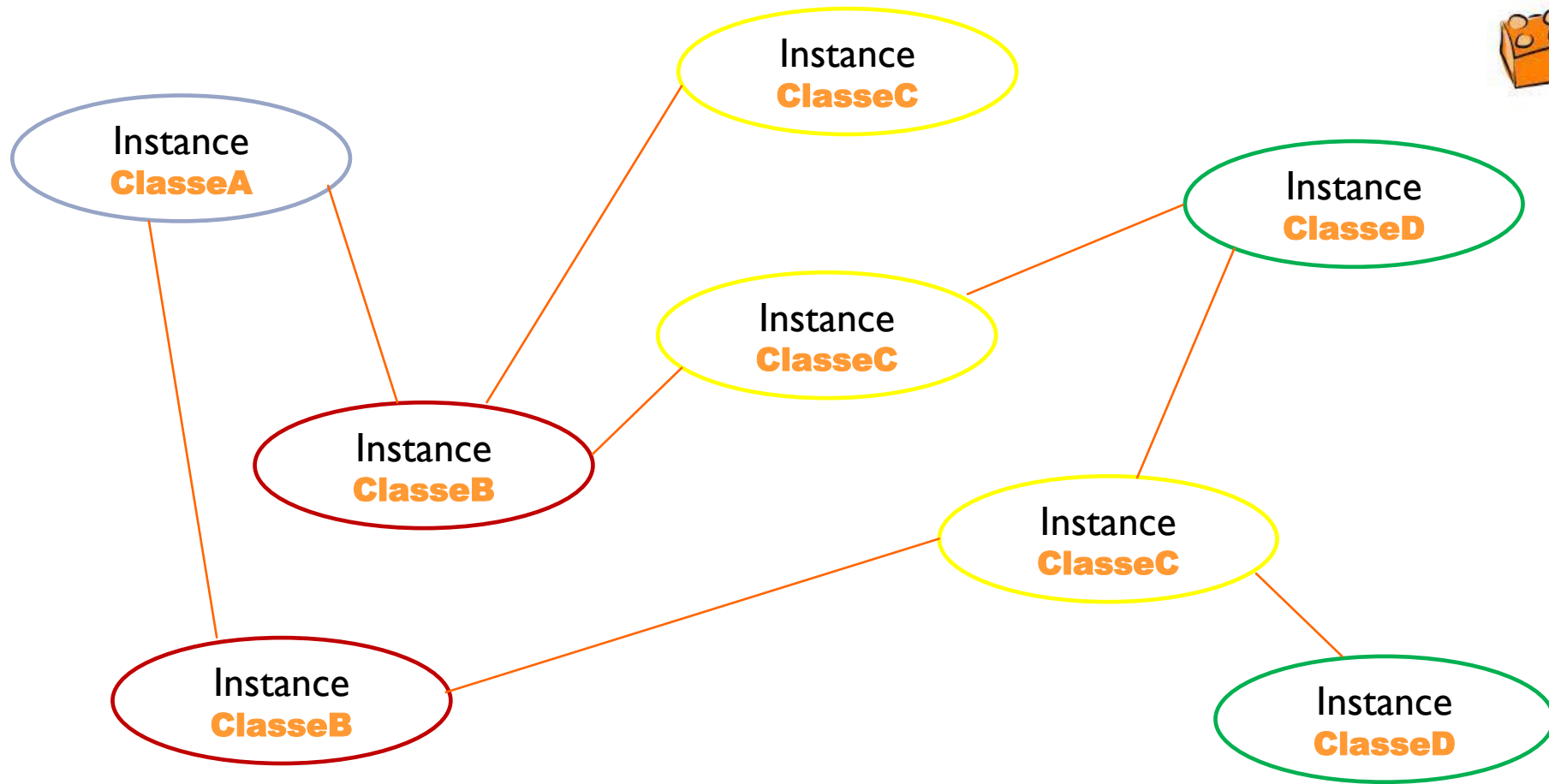
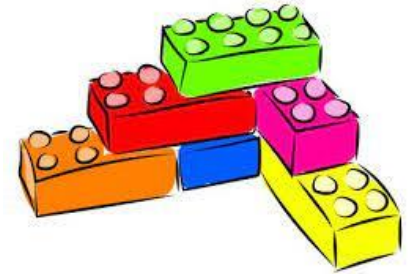


# CLASSES - OBJETS

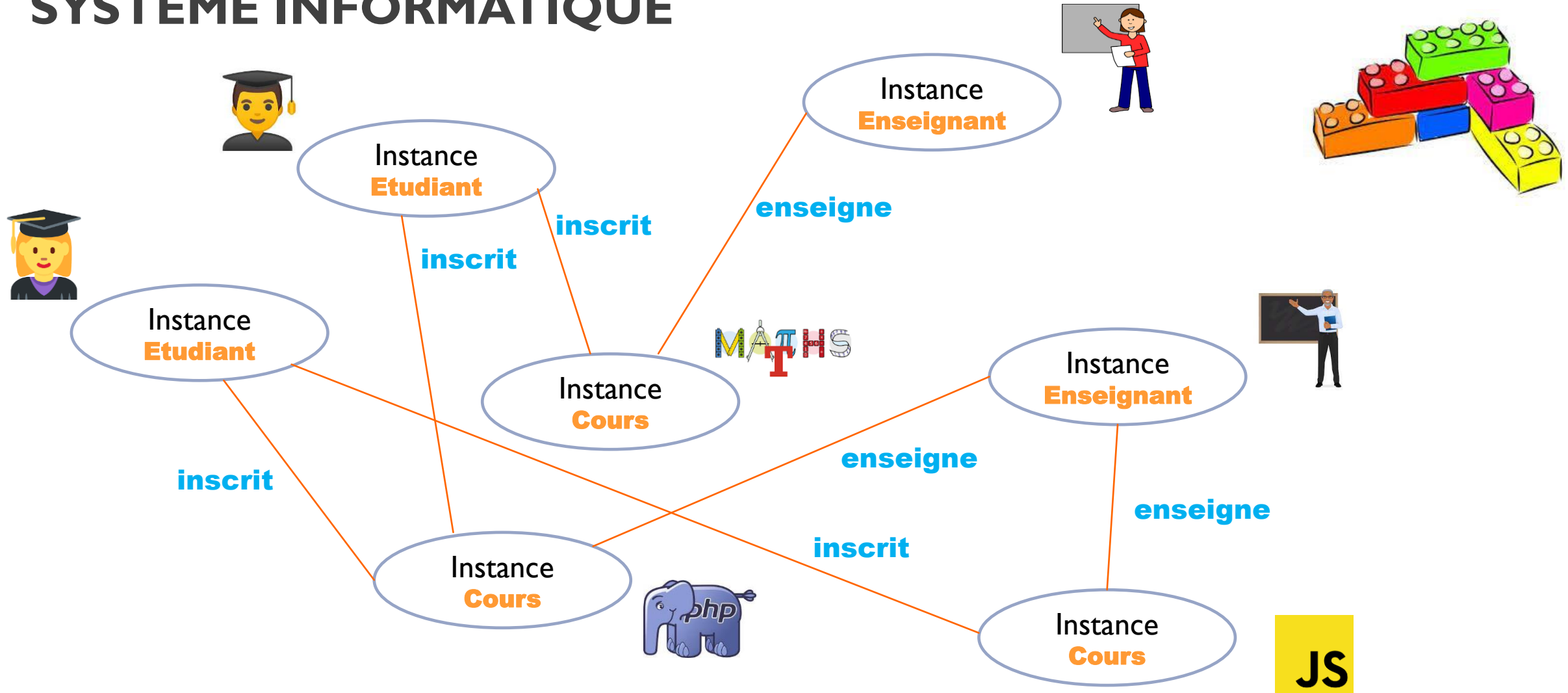
ASSOCIATIONS ENTRE CLASSES

**BTS SIO1 – ANNÉE 2022/2023 – LYCEE PERGAUD**

# SYSTÈME INFORMATIQUE

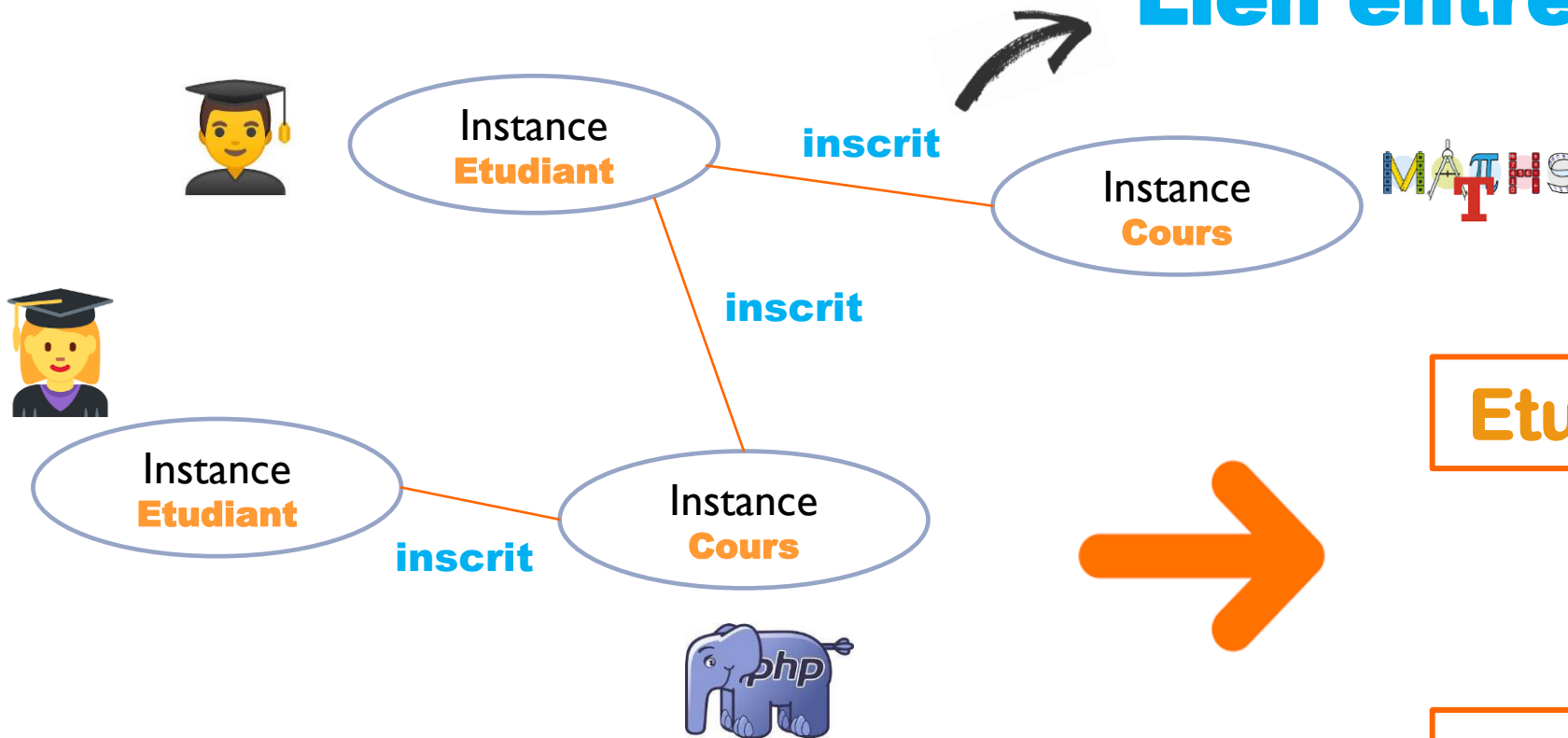


# SYSTÈME INFORMATIQUE

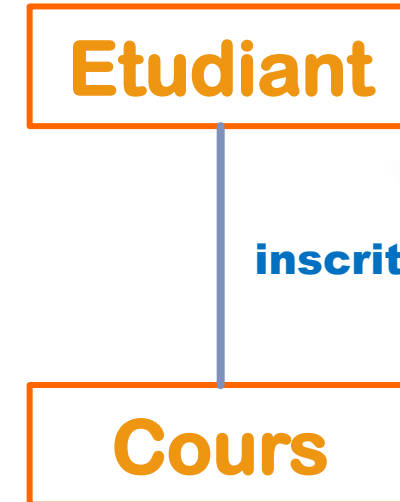


# SYSTÈME INFORMATIQUE

**Lien entre 2 instances**



**Association  
entre 2  
classes**



**Diagramme de classes**

# SYSTÈME INFORMATIQUE



**Identifier les classes et  
leurs associations**

# ETAPES

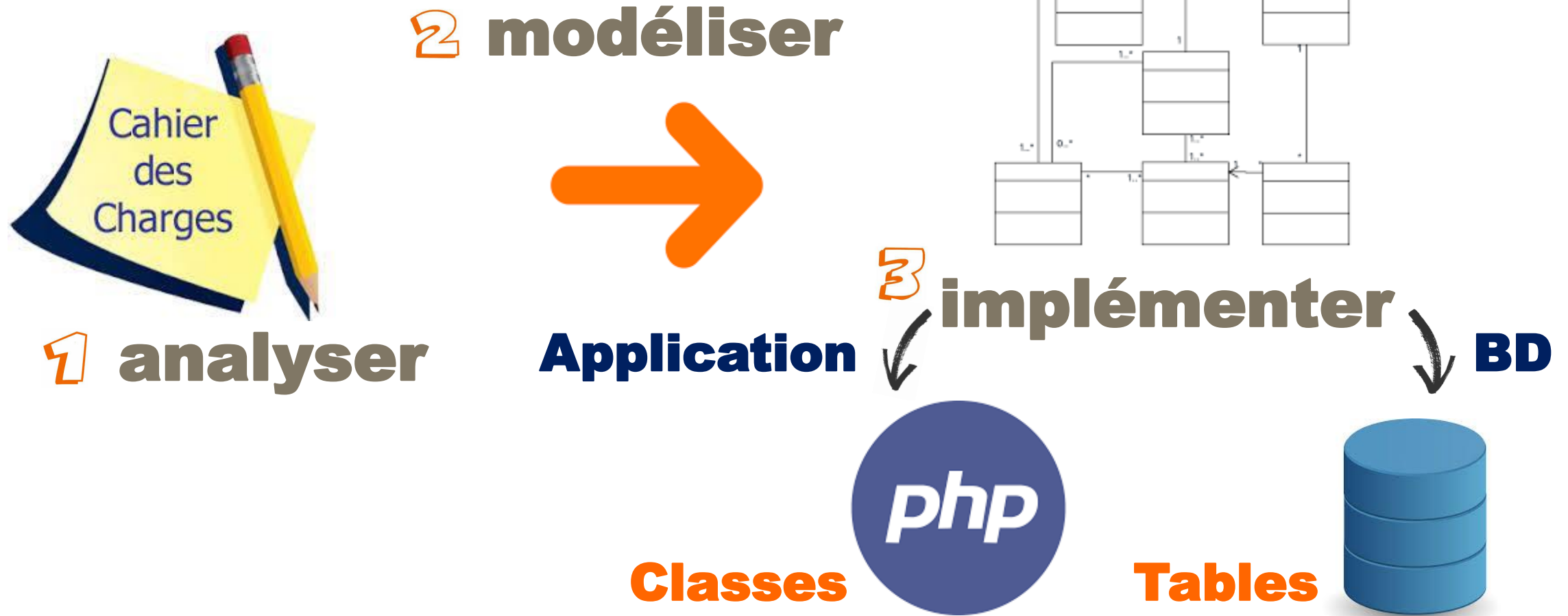


- (1) **Analyse**
- (2) **Modélisation**
- (3) **Implémentation**



# ETAPES

## Diagramme de classes



---

# MODÉLISATION



**Diagramme de classes – POO**



**Modèle Conceptuel de Données (MCD)**



# DIAGRAMME DE CLASSES

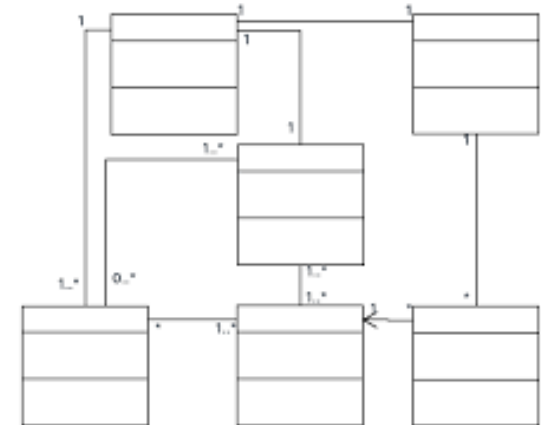


**1 analyser**

**2 modéliser**



## Diagramme de classes



# DIAGRAMME DE CLASSES



**Le diagramme de classes est un schéma utilisé pour présenter les classes ainsi que leurs relations (associations)**

# DIAGRAMME DE CLASSES

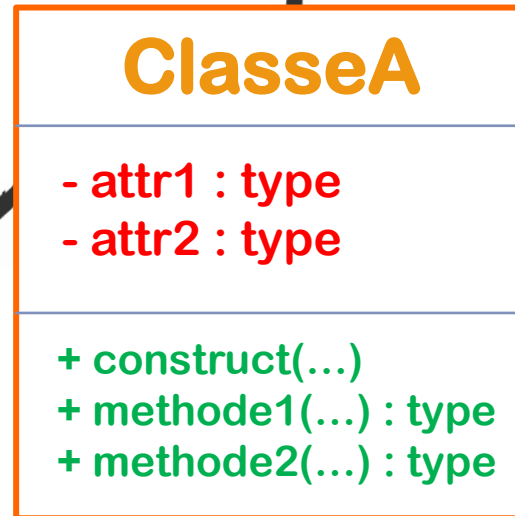


**Classe**

**nom classe**

**visibilité**

**attributs**



**privé (-)**  
**public (+)**

**méthodes**

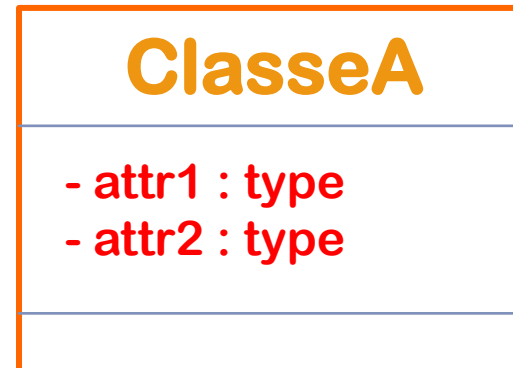
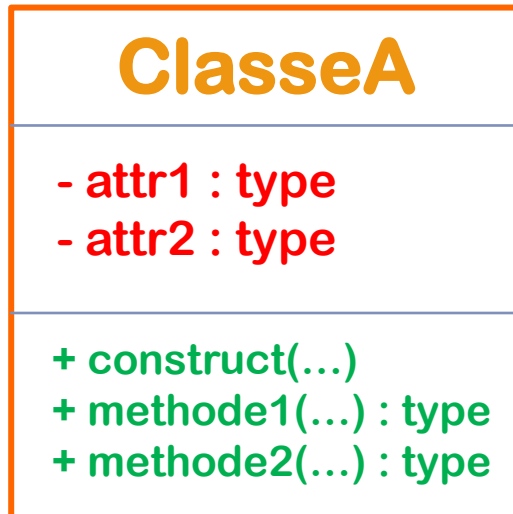
# DIAGRAMME DE CLASSES



**Classe**



**granularité**  
**niveau de détail**



# DIAGRAMME DE CLASSES



## Association



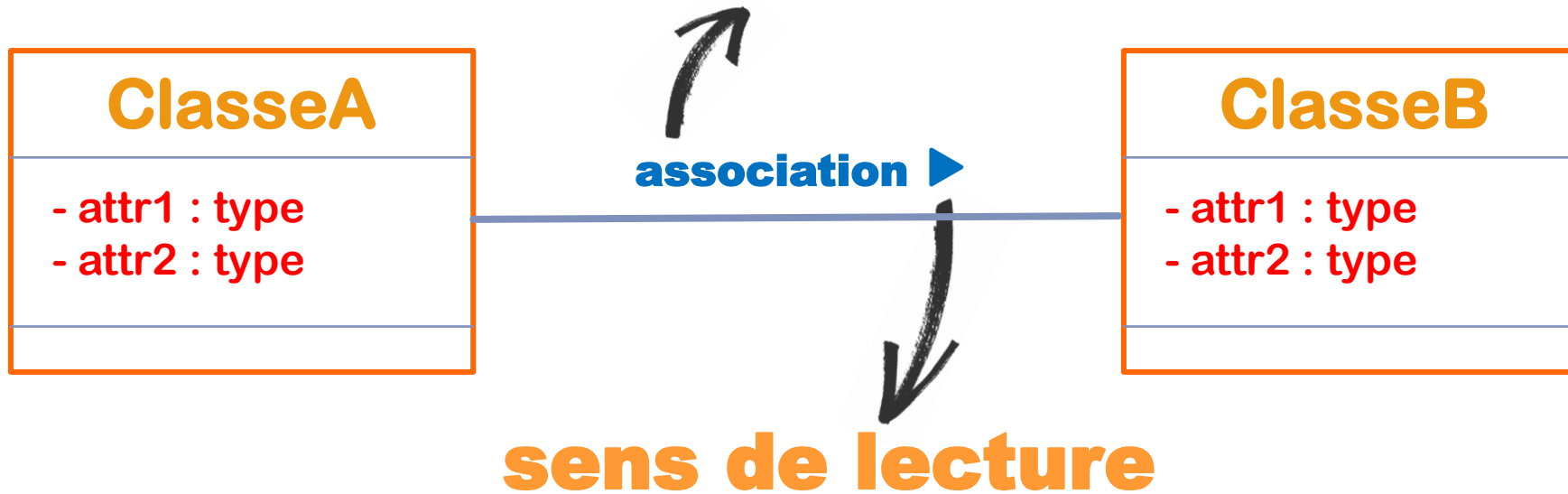
**C'est une relation entre 2 classes**  
**DONC**  
**un lien entre les instances de 2 classes**

# DIAGRAMME DE CLASSES



## Association

nom association



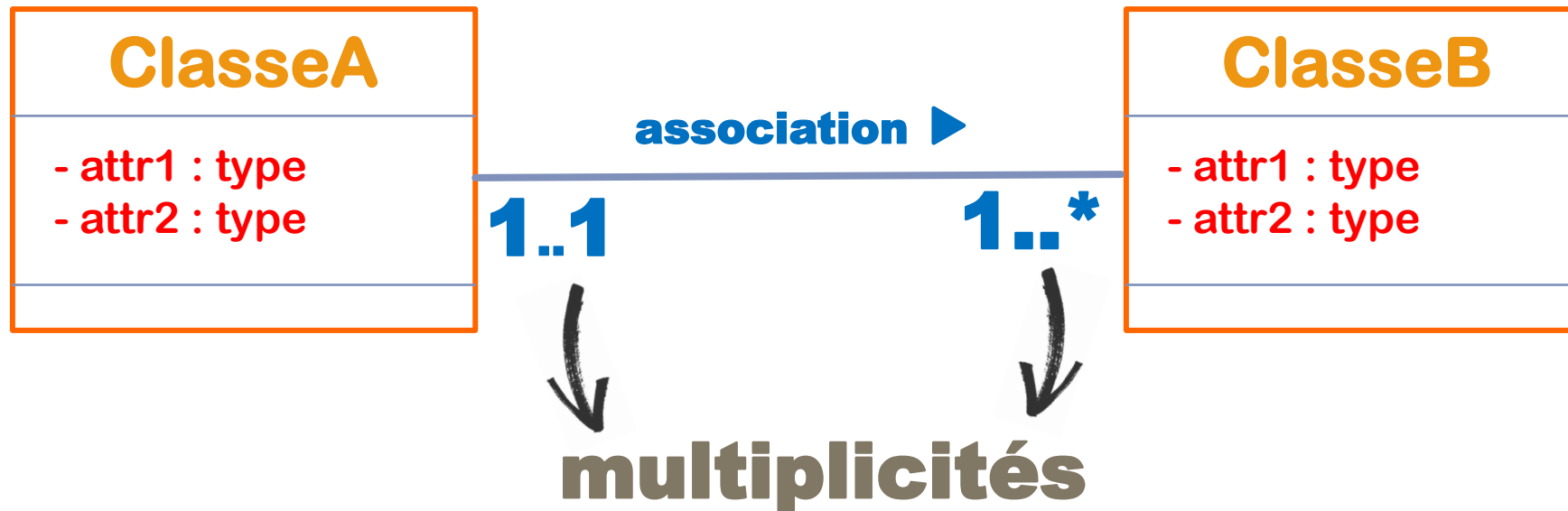
# DIAGRAMME DE CLASSES



## Multiplicités



**cardinalités**  
(MCD)



# DIAGRAMME DE CLASSES



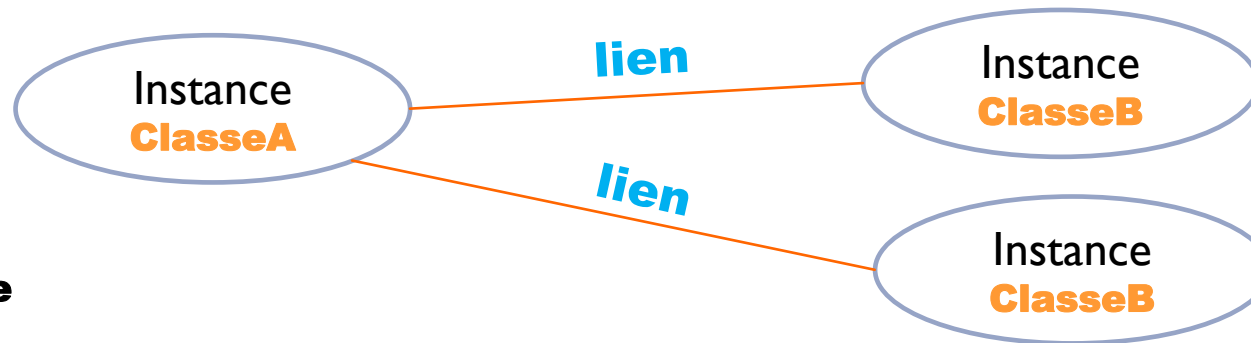
## Multiplicités



**cardinalités**  
(MCD)



Une instance de la  
classe **ClasseA** est  
**liée** à **UNE-OU-  
PLUSIEURS** instances de  
la classe **ClasseB**



Une instance de la  
classe **ClasseB** est  
**liée** à **UNE-ET-UNE-  
SEULE** instance de la  
classe **ClasseA**



# DIAGRAMME DE CLASSES



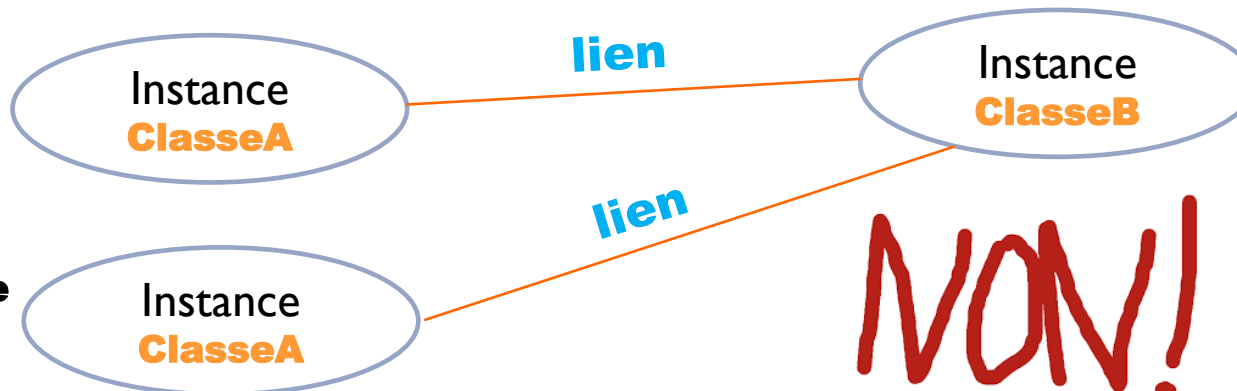
## Multiplicités



**cardinalités**  
(MCD)



Une instance de la  
classe **ClasseA** est  
**liée** à **UNE-OU-  
PLUSIEURS** instances de  
la classe **ClasseB**



Une instance de la  
classe **ClasseB** est  
**liée** à **UNE-ET-UNE-  
SEULE** instance de la  
classe **ClasseA**

# DIAGRAMME DE CLASSES



## Multiplicités



## cardinalités (MCD)

**a..b**



minimum



maximum

**0..1** (0,1)

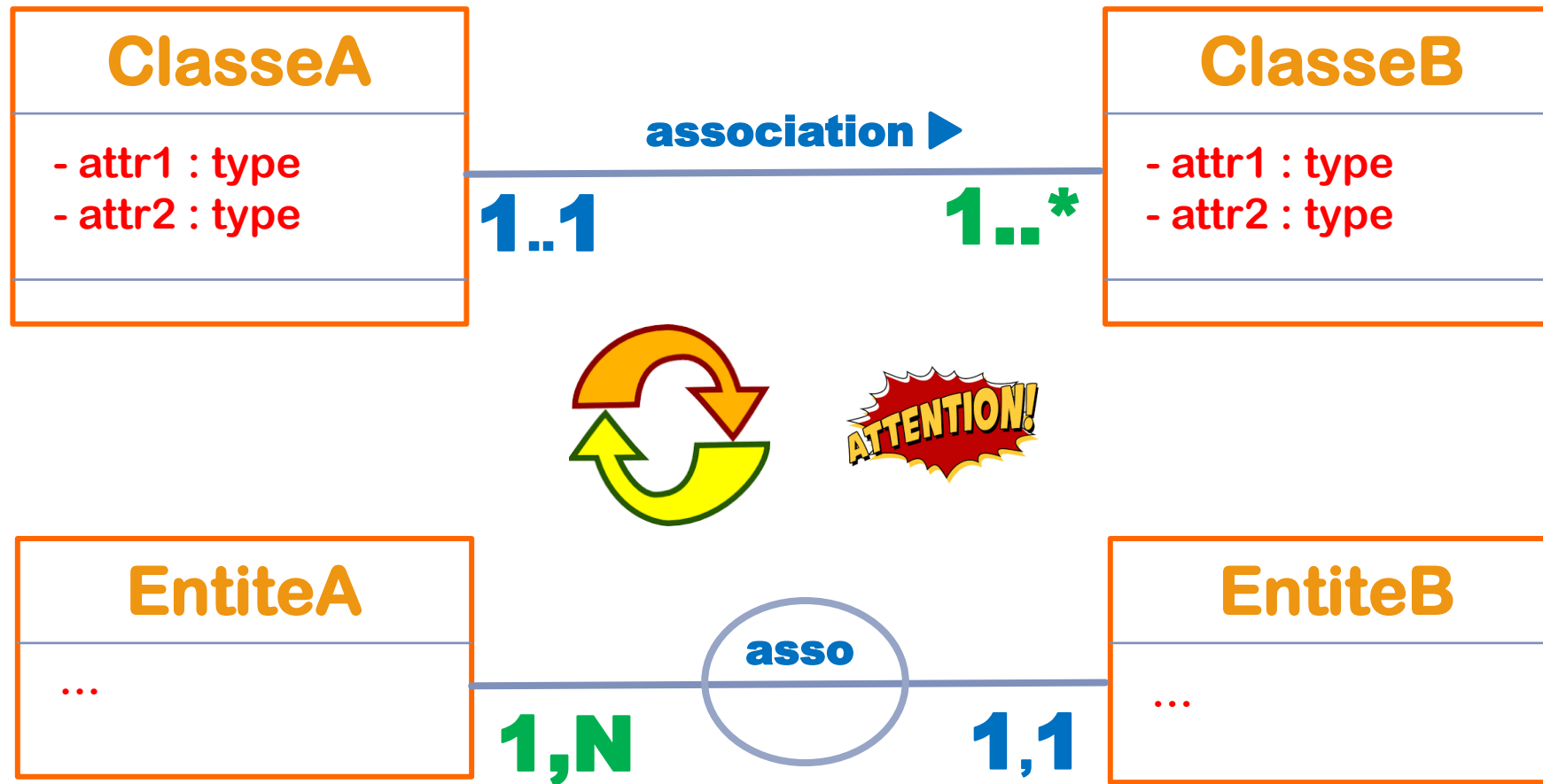
**1..1 ou 1** (1,1)

**0..\* ou \*** (0,N)

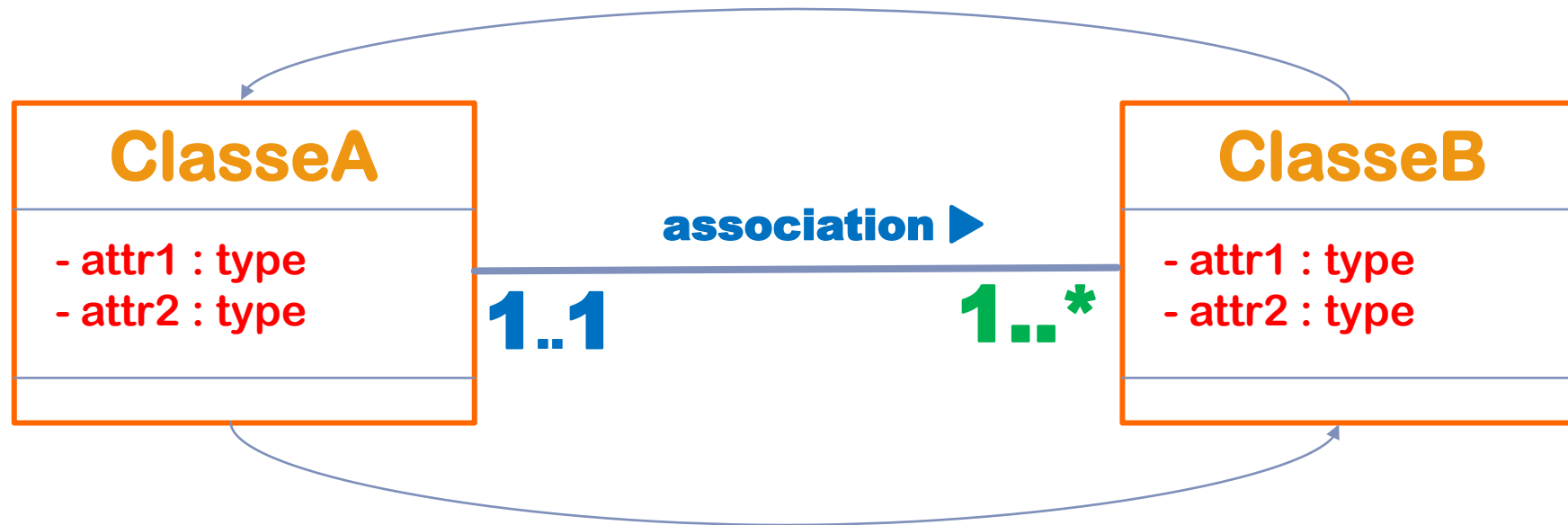
**1..\*** (1,N)

**N..M** (N,M)

# DIAGRAMME DE CLASSES



# DIAGRAMME DE CLASSES



➔ Association **BIDIRECTIONNELLE**

# DIAGRAMME DE CLASSES

EXAMPLE



## **Le contexte**

On souhaite gérer les livres proposés par la bibliothèque Albert Camus de Besançon. Un livre est caractérisé par un ISBN, un titre, une date de parution et un nombre de pages. Un livre est écrit par un auteur. Un auteur est caractérisé par un prénom et un nom.

## **Travail demandé**

Proposer le diagramme de classes permettant de modéliser la situation contextuelle présentée ci-dessus.

# DIAGRAMME DE CLASSES

# Analyse



# DIAGRAMME DE CLASSES



1

**les classes**

**les attributs**



## **Le contexte**

On souhaite gérer les **livres** proposés par la bibliothèque Albert Camus de Besançon. Un **livre** est caractérisé par un **ISBN**, un **titre**, une **date de parution** et un **nombre de pages**. Un **livre** est écrit par un **auteur**. Un **auteur** est caractérisé par un **prénom** et un **nom**.

## **Travail demandé**

Proposer le diagramme de classes permettant de modéliser la situation contextuelle présentée ci-dessus.

# DIAGRAMME DE CLASSES



## 2

## les associations



### Le contexte

On souhaite gérer les **livres** proposés par la bibliothèque Albert Camus de Besançon. Un **livre** est caractérisé par un **ISBN**, un **titre**, une **date de parution** et un **nombre de pages**. Un **livre** est écrit par un **auteur**. Un **auteur** est caractérisé par un **prénom** et un **nom**.

### Travail demandé

Proposer le diagramme de classes permettant de modéliser la situation contextuelle présentée ci-dessus.



# DIAGRAMME DE CLASSES

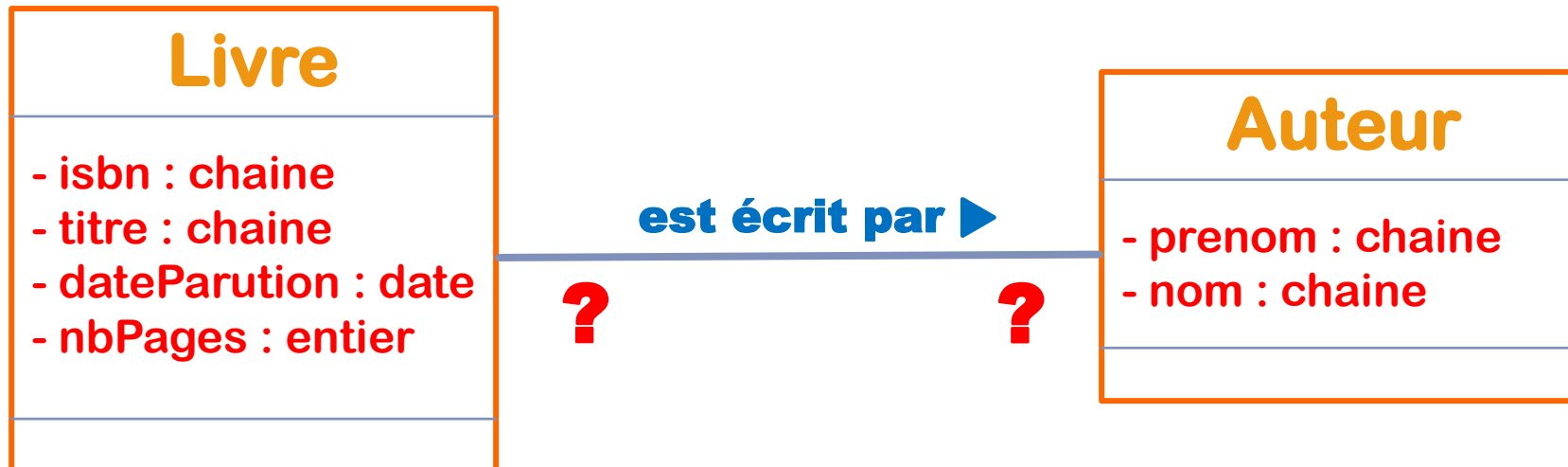


# Modélisation

# DIAGRAMME DE CLASSES



## Diagramme de classes



# DIAGRAMME DE CLASSES



## Multiplicités



Un auteur écrit ? livre(s)

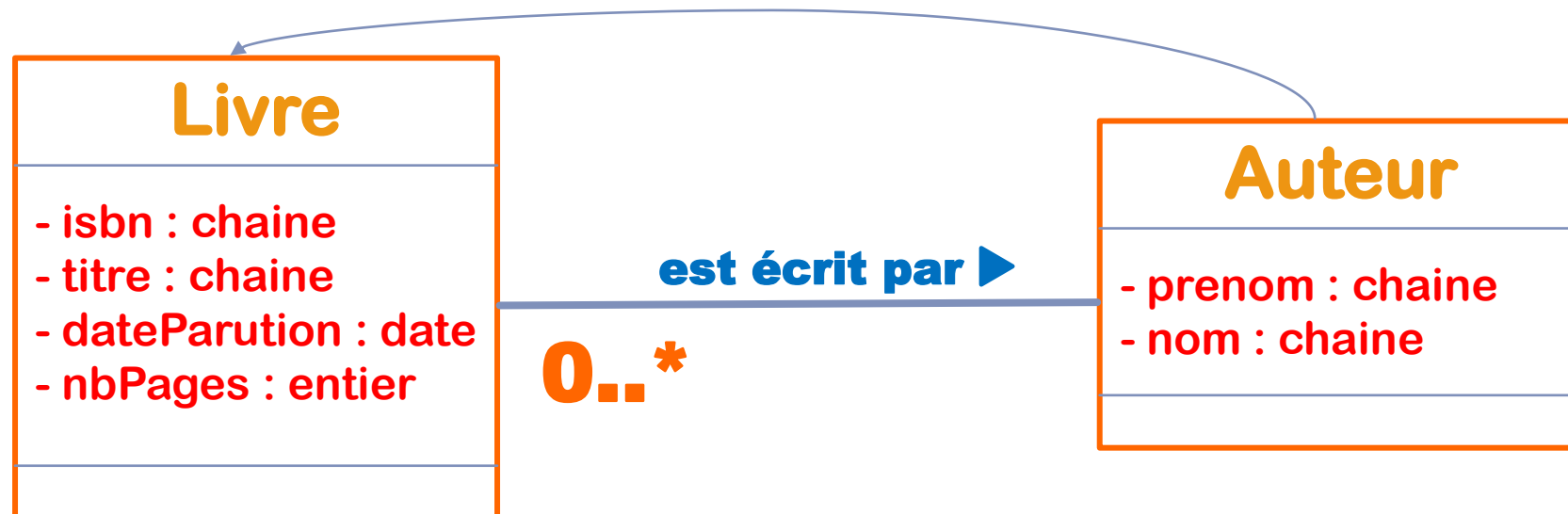


# DIAGRAMME DE CLASSES



## Multiplicités

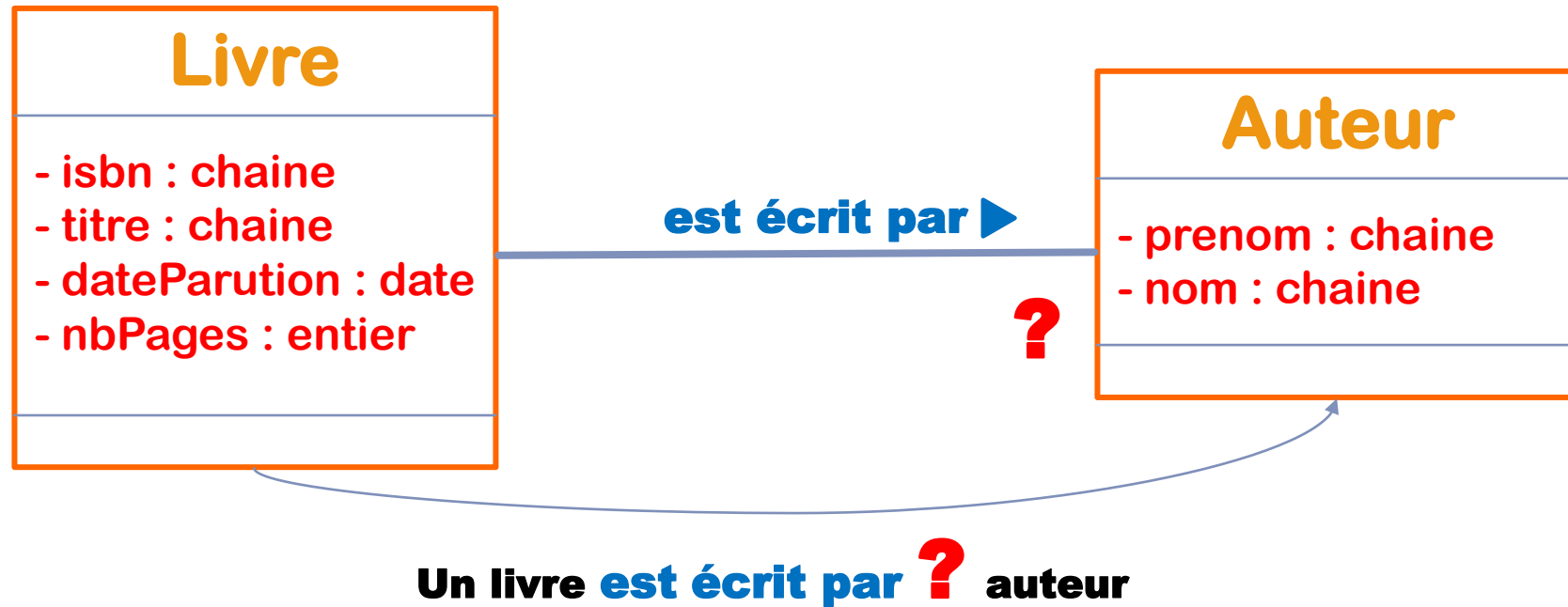
Un auteur écrit 0 ou PLUSIEURS livre(s)



# DIAGRAMME DE CLASSES



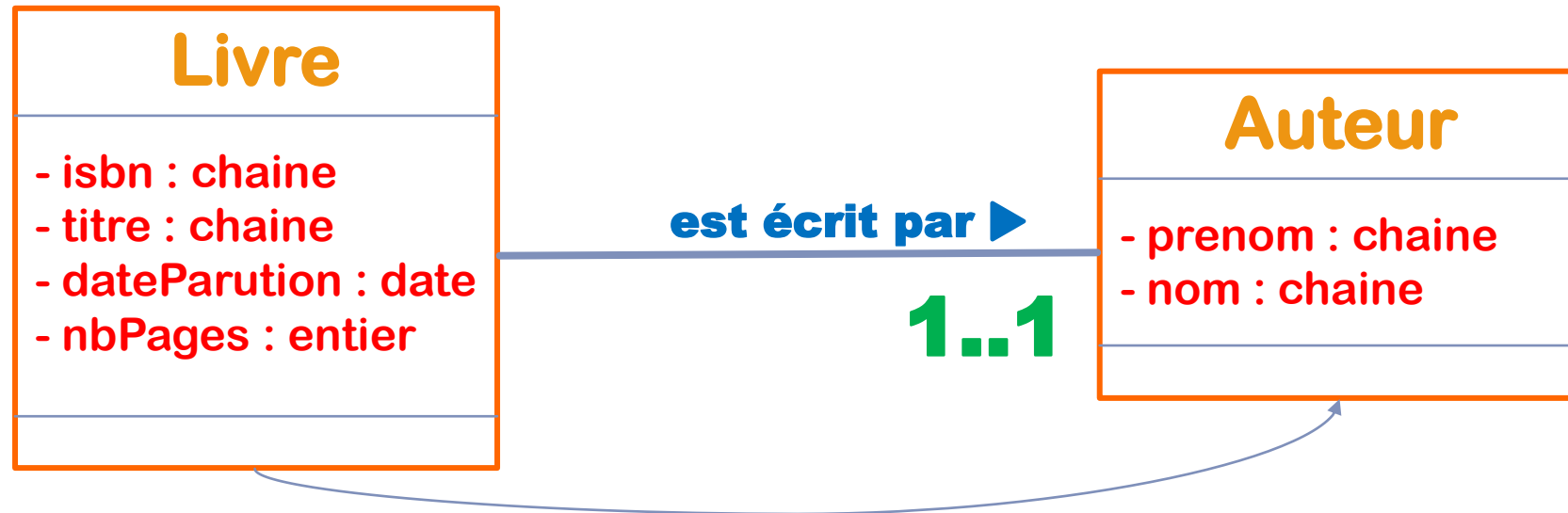
## Multiplicités



# DIAGRAMME DE CLASSES



## Multiplicités



Un livre **est écrit par** **UN ET UN SEUL** auteur

# DIAGRAMME DE CLASSES



## Association



**1..1**

Un livre **est écrit par UN ET UN SEUL** auteur

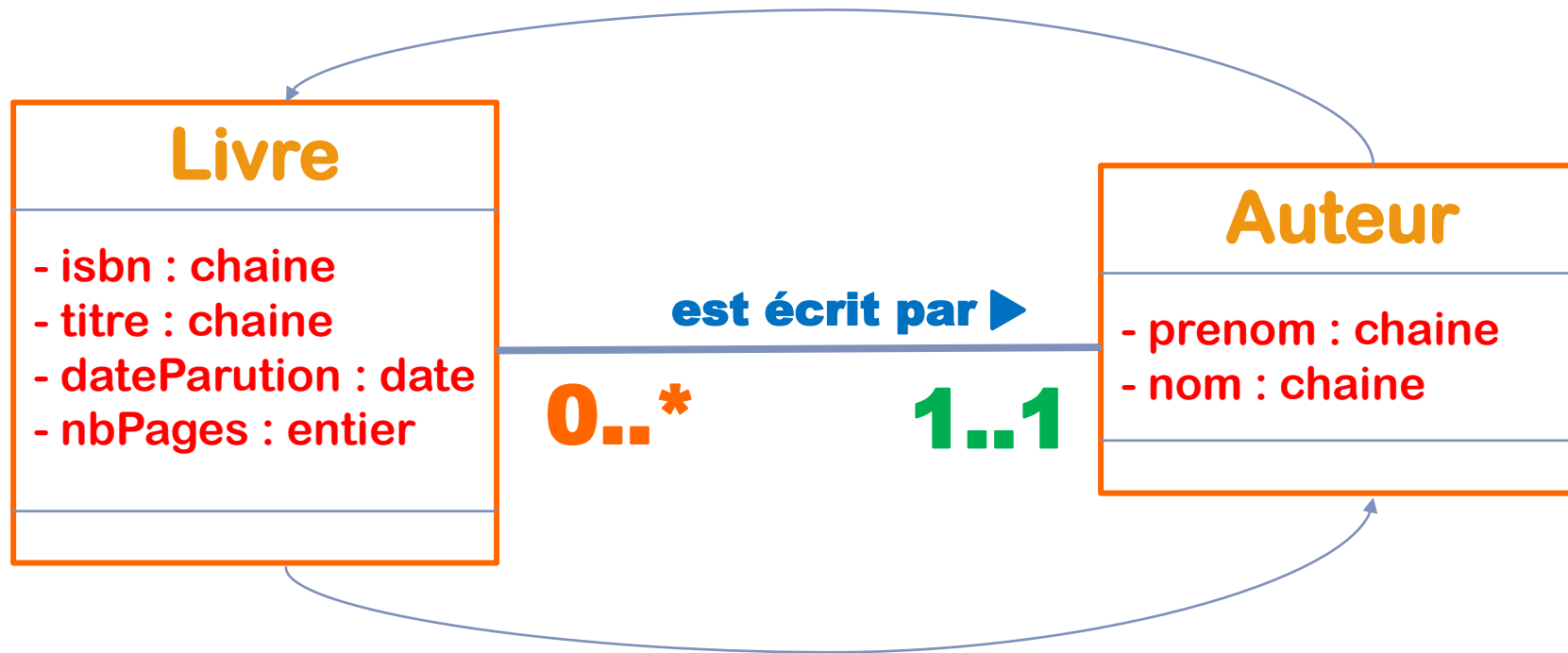
Une instance de la classe **Livre** (un livre) est liée à **UNE ET UNE SEULE** instance de la classe **Auteur**

**0..\***

Un auteur **écrit 0 OU PLUSIEURS** livre(s)

Une instance de la classe **Auteur** (un auteur) est liée à **0 OU PLUSIEURS** instance(s) de la classe **Livre**

# DIAGRAMME DE CLASSES



➔ Association **BIDIRECTIONNELLE**

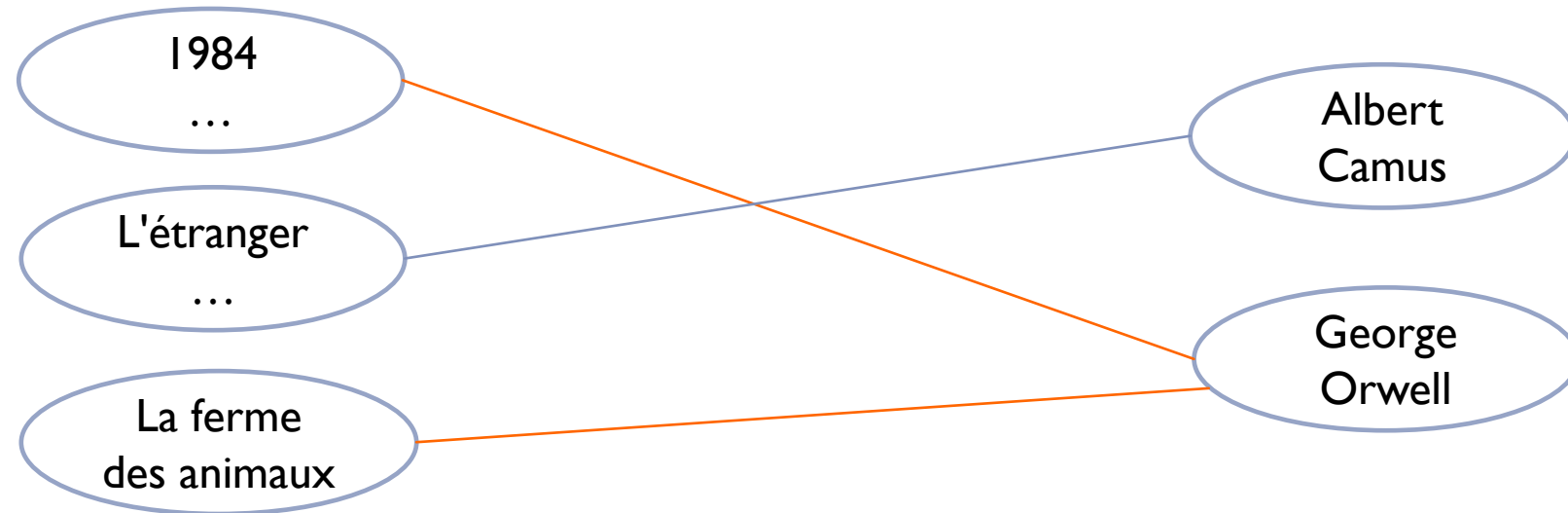


# DIAGRAMME DE CLASSES

**classes**  
(association)



**instances**  
(lien)



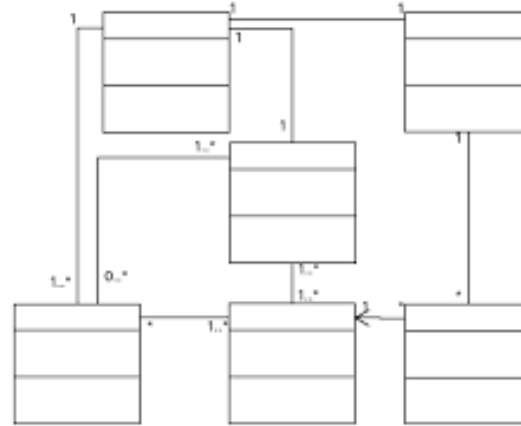
# DIAGRAMME DE CLASSES



# Implémentation

# DIAGRAMME DE CLASSES

## Diagramme de classes



KNOW THE RULES!



**implémenter**

**Règles**



**Règles**

**Classes**



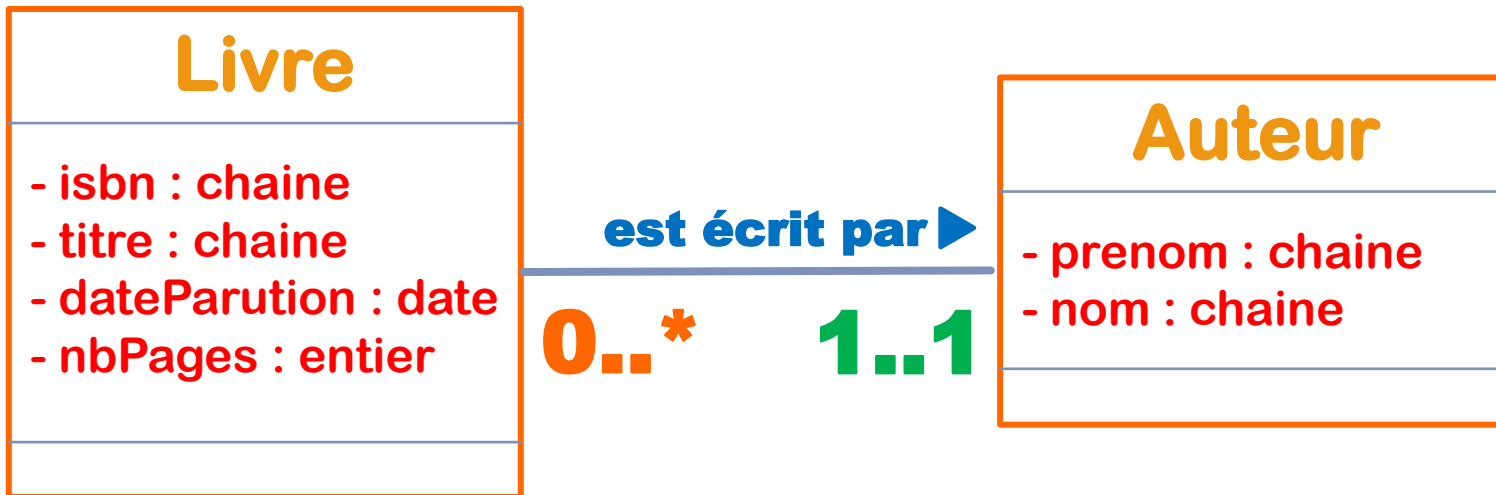
**Tables**

# DIAGRAMME DE CLASSES



**implémenter**

**Règles ?**



# DIAGRAMME DE CLASSES

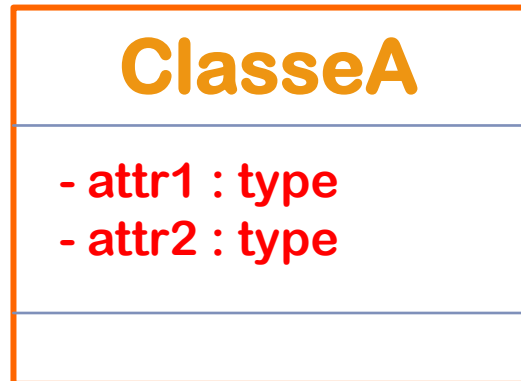


**Classe**



**implémenter**

## Règles



```
class ClasseA {  
    private type $attr1;  
    private type $attr2;  
    // ...  
}
```

# DIAGRAMME DE CLASSES



## Classe



# implémenter

### Livre

- isbn : chaine
- titre : chaine
- dateParution : date
- nbPages : entier



```
class Livre
```

```
{  
    private string $isbn;  
    private string $titre;  
    private int $nbPages;  
    private DateTime $dateParution;  
}
```

### Auteur

- prenom : chaine
- nom : chaine



```
class Auteur
```

```
{  
    private string $nom;  
    private string $prenom;  
}
```

# DIAGRAMME DE CLASSES



## Association



# implémenter



```
class Livre
{
    private string $isbn;
    private string $titre;
    private int $nbPages;
    private DateTime $dateParution;
    // Association avec la classe Auteur
    private Auteur $auteur;
}
```

Un livre **est écrit par UN ET UN SEUL** auteur

Une instance de la classe **Livre** (un livre) est liée à **UNE ET UNE SEULE** instance de la classe **Auteur**



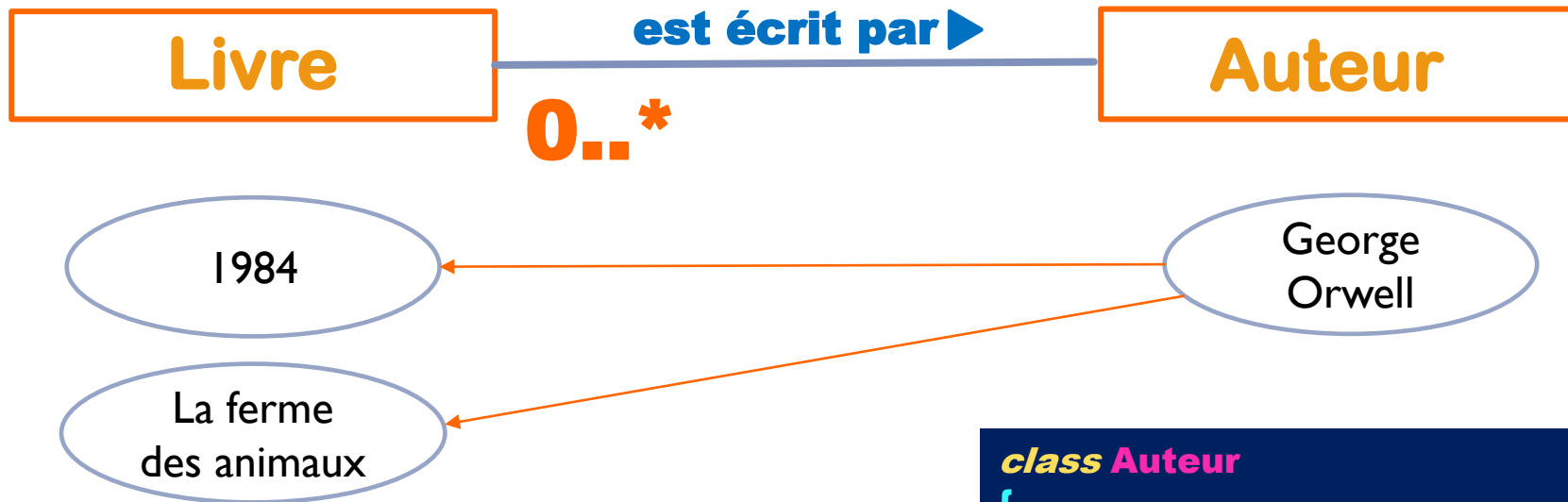
# DIAGRAMME DE CLASSES



## Association



# implémenter



Un auteur **écrit 0 OU PLUSIEURS** livre(s)

Une instance de la classe **Auteur** (un auteur) est liée à **0 OU PLUSIEURS** instance(s) de la classe **Livre**



```
class Auteur
{
    private string $nom;
    private string $prenom;
    // Association avec la classe Livre
    private array $livres = [];
}
```



# DIAGRAMME DE CLASSES

Let's go!



# DIAGRAMME DE CLASSES



**Association bidirectionnelle**



**Complicqué à  
gérer**



**Rendre l'association**  
**UNIDIRECTIONNELLE**

# DIAGRAMME DE CLASSES

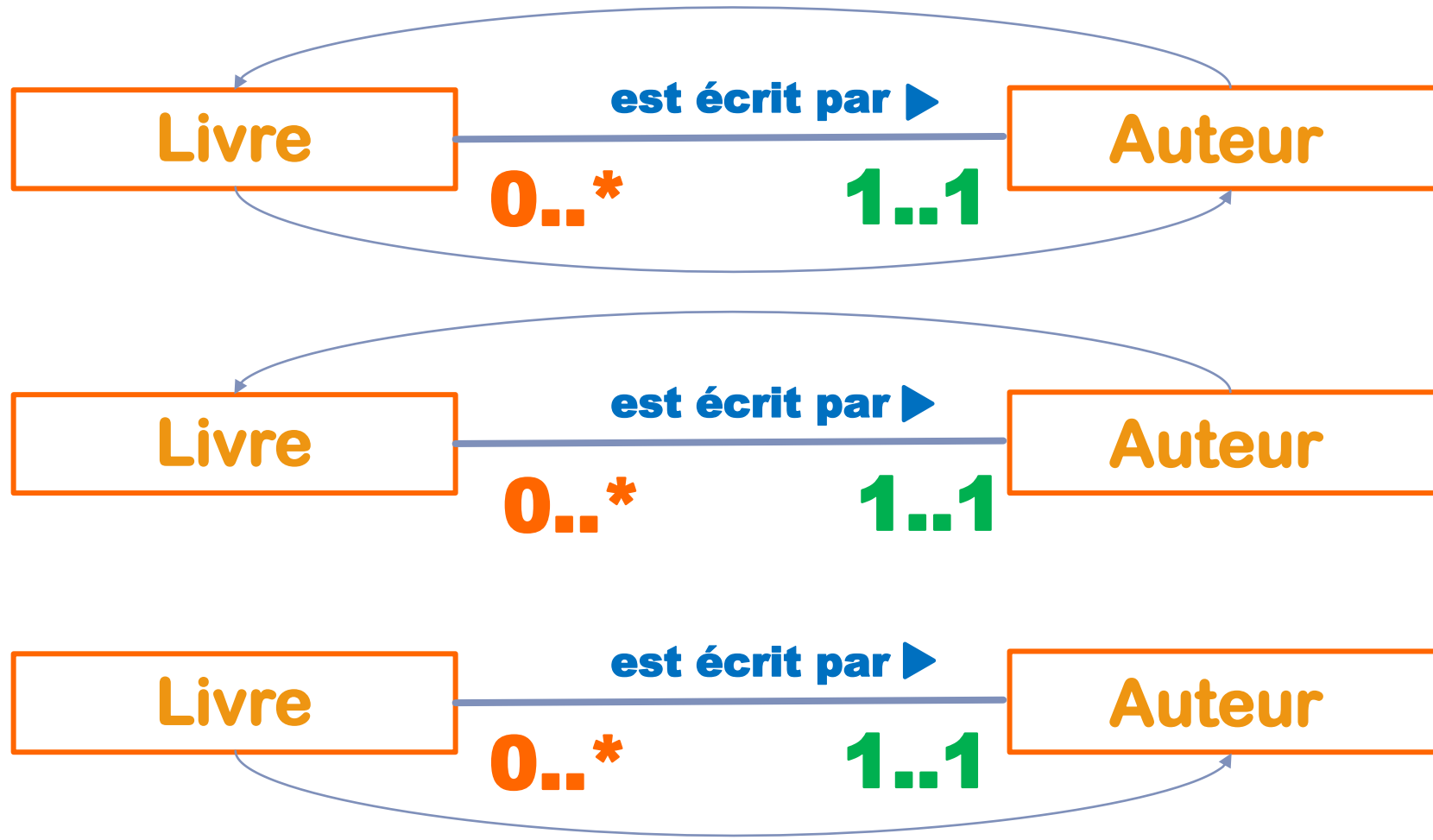


## Association unidirectionnelle



**C'est une association entre 2 classes  
qui s'interprète uniquement dans un  
SEUL SENS**

# DIAGRAMME DE CLASSES



Quel est  
l'auteur du  
livre

?

Quel sont  
les livres de  
l'auteur

?

# DIAGRAMME DE CLASSES

Quel est  
l'auteur d'un  
livre

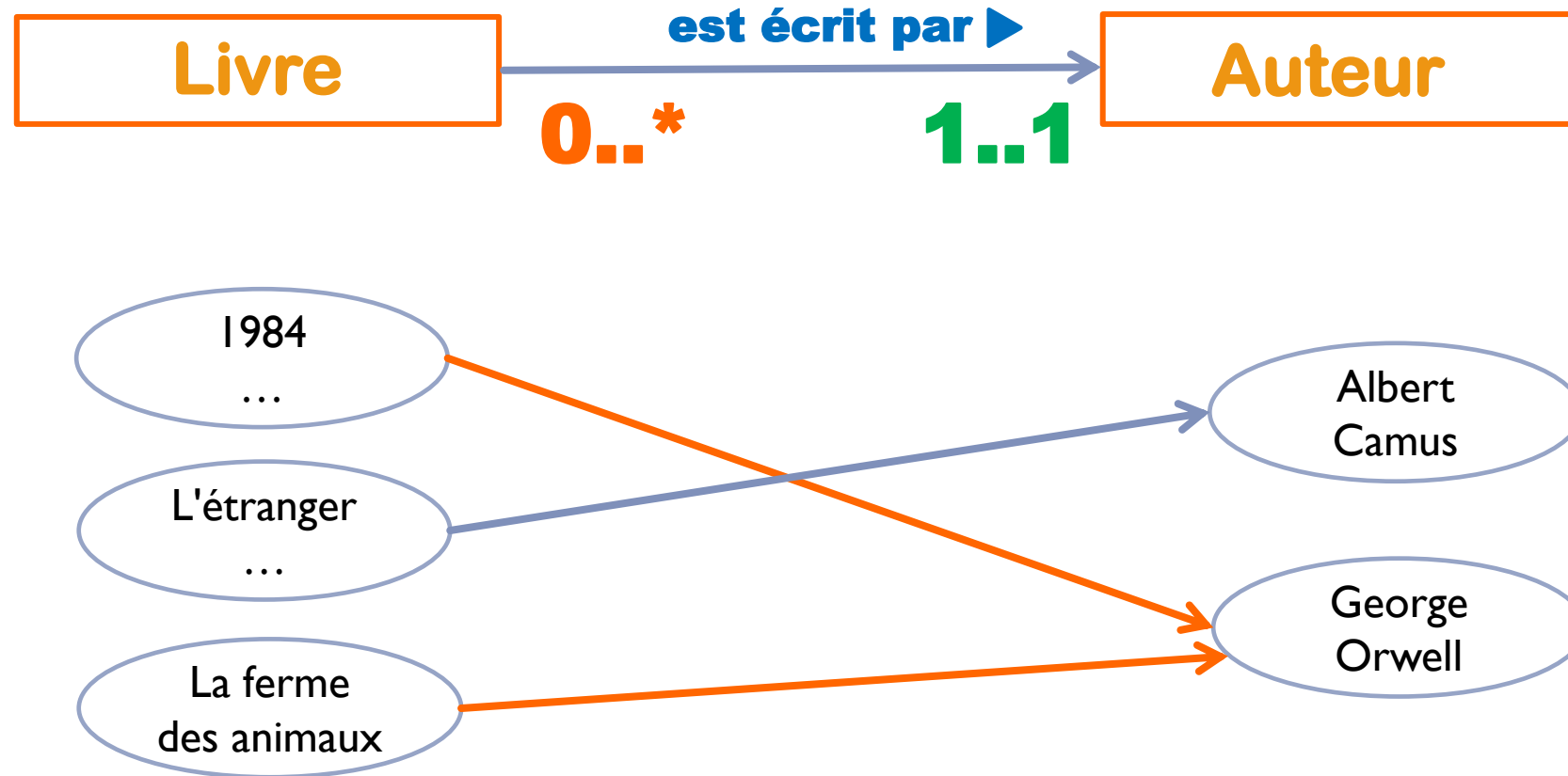


## DIAGRAMME DE CLASSES



**A partir d'un objet **Livre**, on peut naviguer vers (accéder à) l'objet **Auteur** associé**  
**MAIS**  
**Pas l'inverse**

# DIAGRAMME DE CLASSES



# DIAGRAMME DE CLASSES



## Navigabilité



implémenter



```
class Livre
{
    private string $isbn;
    private string $titre;
    private int $nbPages;
    private DateTime $dateParution;
    // Association avec la classe Auteur
    private Auteur $auteur;
}
```

Un livre est lié  
**OBLIGATOIREMENT**  
à un auteur

```
class Auteur
{
    private string $nom;
    private string $prenom;
}
```



# DIAGRAMME DE CLASSES



## Navigabilité



implémenter



```
class Livre
{
    private string $isbn;
    private string $titre;
    private int $nbPages;
    private DateTime $dateParution;
    // Association avec la classe Auteur
    private Auteur $auteur;
}
```

```
public function __construct(
    string $isbn,
    string $titre,
    DateTime $dateParution,
    int $nbPages,
    Auteur $auteur
){
    $this->isbn = $isbn;
    $this->titre = $titre;
    $this->dateParution = $dateParution;
    $this->nbPages = $nbPages;
    $this->auteur = $auteur;
}
```

constructeur

# DIAGRAMME DE CLASSES



## Navigabilité



implémenter



```
class Livre
{
    private string $isbn;
    private string $titre;
    private int $nbPages;
    private DateTime $dateParution;
    // Association avec la classe Auteur
    private ?Auteur $auteur;
}
```

Un livre n'est pas  
**OBLIGATOIREMENT**  
lié à un auteur

```
class Auteur
{
    private string $nom;
    private string $prenom;
}
```

# DIAGRAMME DE CLASSES



## Navigabilité



# implémenter



```
class Livre
{
    private string $isbn;
    private string $titre;
    private int $nbPages;
    private DateTime $dateParution;
    // Association avec la classe Auteur
    private ?Auteur $auteur;
}
```

```
public function __construct(
    string $isbn,
    string $titre,
    DateTime $dateParution,
    int $nbPages,
    ?Auteur $auteur = null
){
    $this->isbn = $isbn;
    $this->titre = $titre;
    $this->dateParution = $dateParution;
    $this->nbPages = $nbPages;
    $this->auteur = $auteur;
}
```

constructeur

```
public function setAuteur(?Auteur $auteur): void
{
    $this->auteur = $auteur;
}
```

setter

# DIAGRAMME DE CLASSES



**Nous souhaitons réaliser une application permettant de gérer des auteurs de livres**



**Quel est  
l'auteur du livre**



**Quel sont les  
livres de l'auteur**



# DIAGRAMME DE CLASSES



**Nous souhaitons réaliser une application permettant de gérer des auteurs de livres**



**navigabilité**



# DIAGRAMME DE CLASSES



## Navigabilité



**implémenter**



```
class Livre
{
    private string $isbn;
    private string $titre;
    private int $nbPages;
    private DateTime $dateParution;
}
```



```
class Auteur
{
    private string $nom;
    private string $prenom;
    // Association avec la classe Livre
    private array $livres = [];
}
```

```
public function ajouterLivre(Livre $livre) {
    // ...
}
```

# DIAGRAMME DE CLASSES

EXAMPLE



## Le contexte

On souhaite gérer les livres proposés par la bibliothèque Albert Camus de Besançon. Un livre est caractérisé par un ISBN, un titre, une date de parution et un nombre de pages. Un livre est écrit par un auteur. Un auteur est caractérisé par un prénom et un nom.

**On souhaite pouvoir associer à un livre une catégorie (roman, SF, aventure, ...).**

**De plus, chaque livre est édité par un éditeur. Un éditeur se caractérise par son nom et sa ville.**

## Travail demandé

**Modifier le diagramme de classes** permettant de modéliser la situation contextuelle présentée ci-dessus.

# DIAGRAMME DE CLASSES

Let's go!





# DIAGRAMME DE CLASSES

