Chapitre 3

Relations entre classes

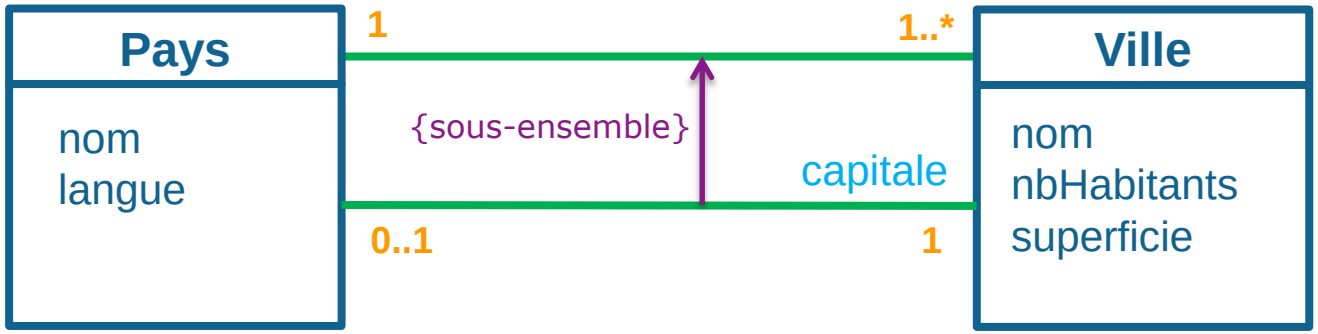
Plan

- Associations
- Décoration des associations
- Agrégations et Compositions
- Dépendances
- Modélisation UML

Types de relation

Le diagramme (de classes) UML va présenter les classes d'un projet et les relations qui les lient en se basant sur certaines conventions de représentation.

Exemple :



Types de relation

On va s'intéresser principalement à 5 types de relations :

- **Association** : un attribut est une instance d'une autre classe
  + 2 cas particuliers d'associations :
    - **Agrégation** : association où une classe est plus importante que l'autre
    - **Composition** : association de type partie/tout ou composant/composite

- **Dépendance** : une classe utilise de temps en temps une autre

- **Héritage** : une classe est un cas particulier d'une autre classe
  Voir modules suivants

Plan

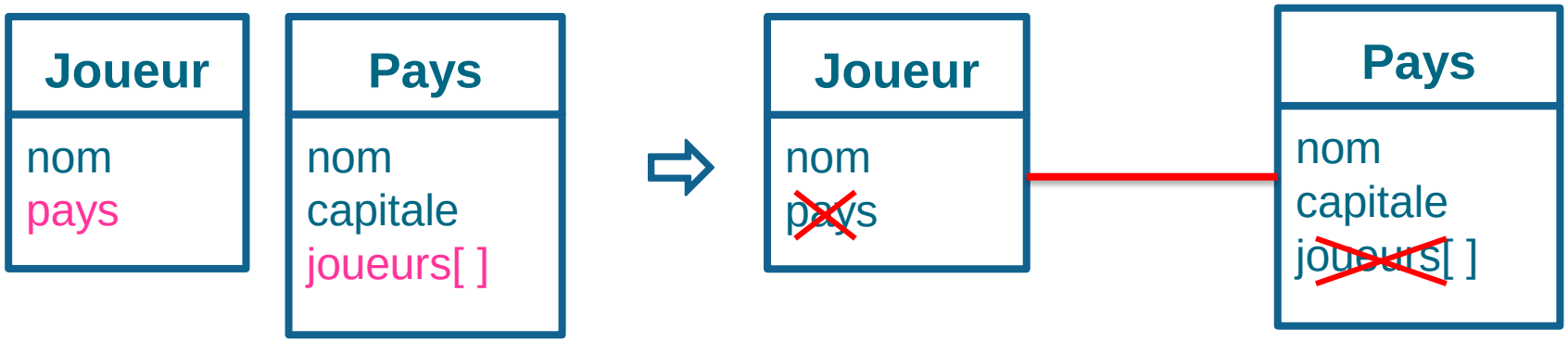
- Associations
- Décoration des associations
- Agrégations et Compositions
- Dépendances
- Modélisation UML

Association

Association entre deux classes

= chacune des classes a un attribut instance de l'autre

= lien durable entre les objets des deux classes

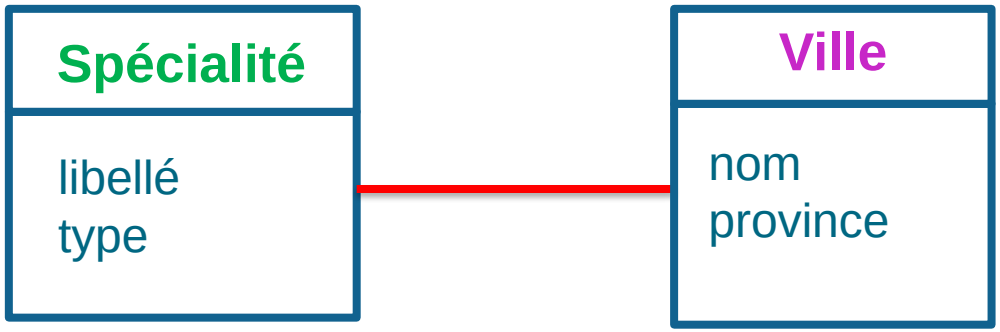


Représentation en UML : ligne pleine entre les classes

Note : on ne note pas dans les classes les attributs concernés par l'association (ils sont représentés par la ligne).

Association

Autre exemple : les spécialités belges et leur ville d'origine



La ligne représente deux attributs qu'on ne note plus dans le diagramme :

- un attribut `Ville villeOrigine` dans Spécialité
- un attribut `Spécialité [] spécialités` dans Ville

Cela correspond à une navigabilité bidirectionnelle :

- À partir d'une spécialité, on peut trouver sa ville d'origine.
- À partir d'une ville, on peut trouver ses spécialités.

Association

Notation : ligne pleine

Signification :

- (A) contient une référence à (B) sous la forme d'un attribut et vice versa

Plan

- Associations
- ▸ Décoration des associations
- Agrégations et Compositions
- Dépendances
- Modélisation UML

Décoration d'association : nature

Préciser la signification d'une relation en la décorant/documentant…

En indiquant la **nature** de la relation (verbe conjugué + flèche)

Décoration d'association : rôles

Préciser la signification d'une relation en la décorant/documentant…

En indiquant les **rôles** des classes associées (noms)

Décoration d'association : multiplicités

Préciser la signification d'une relation en la décorant/documentant…

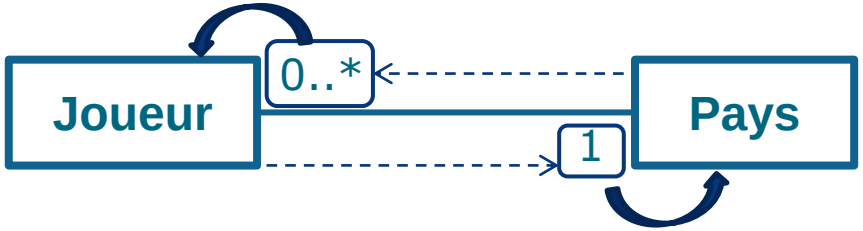
En indiquant les **multiplicités** ou **cardinalités**

Décoration d'association : multiplicités

Les cardinalités/multiplicités précisent le nombre d'objets pouvant participer à une relation.

1	un et un seul
0..1	zéro ou un
1..*	un à plusieurs
0 ..* ou *	0 à plusieurs
M..N	entre M à N (entiers naturels)

Elles se placent à côté de la classe dont elles indiquent le nombre.



indique le sens de la lecture

- *Un joueur appartient à 1 pays*
- *Un pays peut avoir n'importe quel nombre de joueurs*

Exercice 1

Répondez aux questions suivantes en interprétant ce diagramme UML

- Quels sont les attributs (variables d'instance en java) qui traduisent ces deux relations ?
- Interprétez la décoration "0..1" en la faisant apparaître dans une phrase.
- Si un propriétaire ne possède qu'un seul bien immobilier et le vend, peut-on garder l'objet Personne correspondant ?
- Plusieurs personnes peuvent-elles posséder un bien en copropriété ?
- Un bien immobilier peut-il être loué par un groupe de locataires (collocation) ?

Exercice 1

Répondez aux questions suivantes en interprétant ce diagramme UML

- Quels sont les attributs (variables d'instance en java) qui traduisent ces deux relations ?

Dans la classe Personne :

propriétés [ ] : BienImmobilier

locations [ ] : BienImmobilier

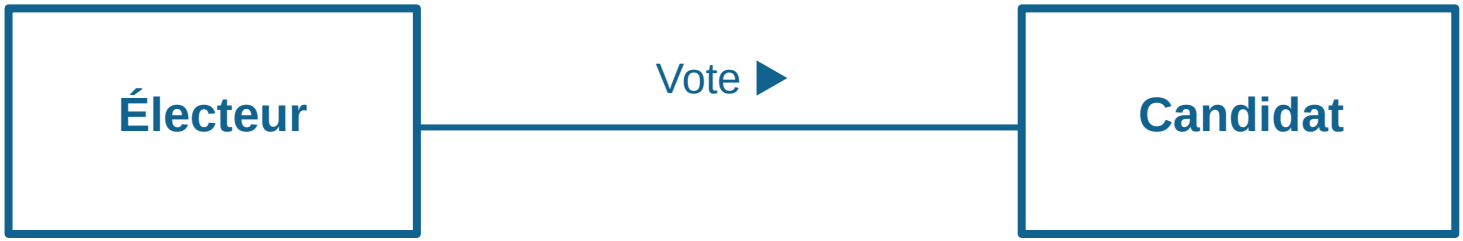
Dans la classe BienImmobilier :

propriétaire : Personne

locataire [0..1] : Personne

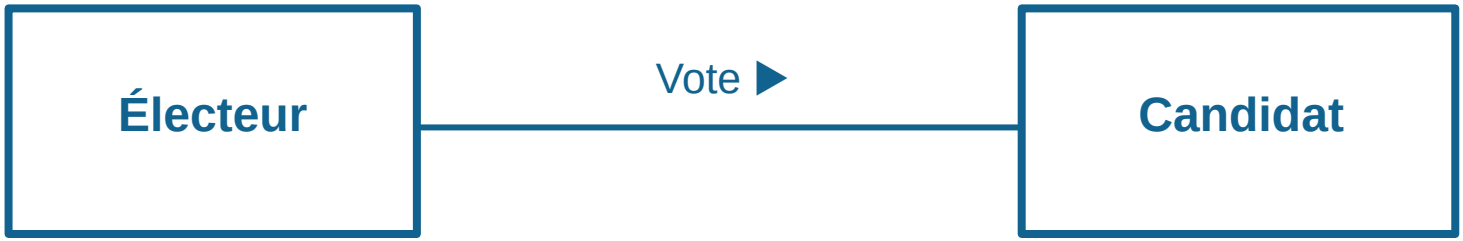
Exercice 2

Complétez les multiplicités du diagrammes UML ci-dessous, sachant qu'un électeur vote pour ***au moins un*** candidat



Exercice 2

Complétez les multiplicités du diagrammes UML ci-dessous, sachant qu'un électeur vote pour ***au plus un*** candidat



Exercice 3

Modélisez sous la forme de diagrammes UML

- Un continent a une superficie et un nom. Un pays a un nom et est situé sur un seul continent. On connait le nombre approximatif d'habitants de chaque pays.

Exercice 4

Modélisez sous la forme de diagrammes UML

- Chaque point est déterminé par ses coordonnées en X et en Y. Toute droite porte un nom et est déterminée par deux points.

Exercice 5

Modélisez sous la forme de diagrammes UML

- Un appartement a une adresse, un loyer éventuel (s'il est à louer) et au moins 2 pièces. Une pièce a une superficie et peut avoir un balcon

- Qu'ajouter au schéma si on souhaite préciser aussi la superficie totale de chaque appartement ?

Exercice 6

Modélisez sous la forme de diagrammes UML

- Des courses à pied sont organisées à des dates précises. On connaît leur distance en Kms. Pour certaines courses, il y a un temps limite pour les parcourir.

- Par coureur, on répertorie son nom, son adresse et son numéro de Gsm, si du moins il accepte de le donner. Certains coureurs sont des athlètes de handisport.

- Un coureur peut participer à plusieurs courses.

Exercice 7

Modélisez sous la forme d'un diagramme UML :

- une classe **AnneeEtude**, chaque année d'étude (à l'Henallux) étant décrite par une section et un numéro de bloc (1, 2 ou 3) ;
- une classe **Étudiant**, chaque étudiant étant décrit par son nom, son prénom, sa date de naissance et son année d'étude ;
- une classe **Professeur**, chaque professeur étant décrit par son nom, son matricule, sa spécialité principale et un numéro de gsm s'il accepte de le communiquer ;
- une classe **ActivitéApprentissage**, chaque activité d'apprentissage étant décrit par un nom, un nombre d'heures, l'année d'étude où elle est organisée et le(s) professeur(s) qui s'en charge(nt) ;
- une classe **Voiture**, chaque voiture (d'un professeur) étant décrite par sa plaque, sa marque (pour le contrôle du parking) et le professeur auquel elle appartient.

Exercice 8

Modélisez sous la forme d'un diagramme UML :

- Dans un centre de formation, un cours, caractérisé par un titre et un nombre d'heures, aborde un ou plusieurs sujets.
- Chaque sujet est décrit par un nom et un niveau d'importance (entre 1 et 5) et n'est abordé que dans un seul cours.
- À chaque sujet sont associés des questions (d'examen) décrites par un énoncé et un niveau de difficulté. Une question peut couvrir entre 1 et 3 sujets.
- Un examen, caractérisé par une date, comporte entre 5 et 10 questions principales auxquelles viennent s'ajouter au maximum 3 questions facultatives. Une même question peut être utilisée dans plusieurs examens.

Plan

- Associations
- Décoration des associations
- ▸ Agrégations et Compositions
- Dépendances
- Modélisation UML

Agrégation

Agrégation

= forme particulière d'association asymétrique,
où une classe est plus importante que l'autre

L'importance relative des classes dépend du contexte !

Exemple

Agrégation

Notation : ligne pleine avec losange du côté de l'agrégat

Signification :

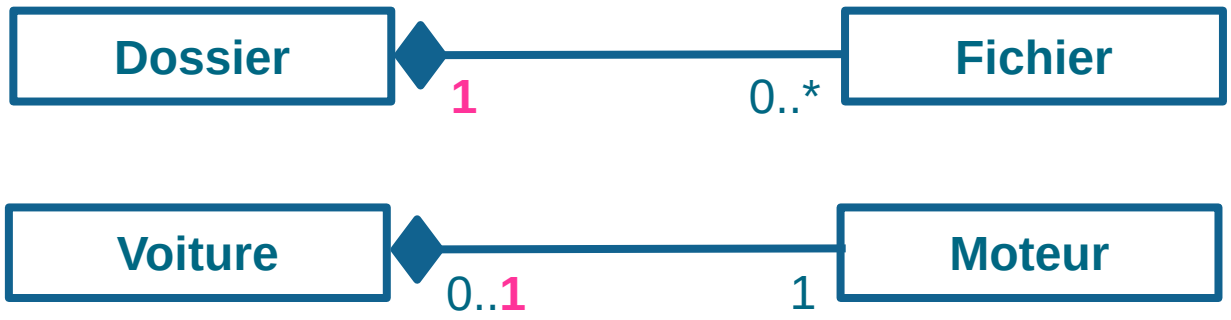
- (A) et (B) sont en association, mais (A) joue un rôle plus important.
- (B) pourrait être une partie / un esclave de (A).

Composition

Composition = forme plus forte d'agrégation où

- la multiplicité au niveau du composite est <= 1
  c'est-à-dire une composante ne peut pas faire partie de plusieurs composites en même temps !
- le cycle de vie des parties dépend de celui du composite
  c'est-à-dire quand le composite cesse d'exister en mémoire, les composantes sont elles aussi détruites (leur existence n'a aucun sens si le composite n'existe plus).

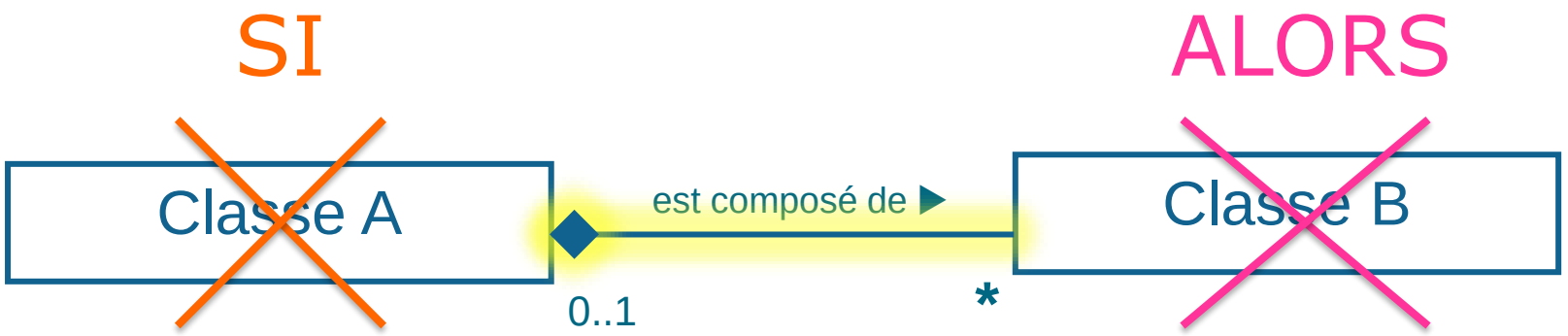
Exemples



Composition

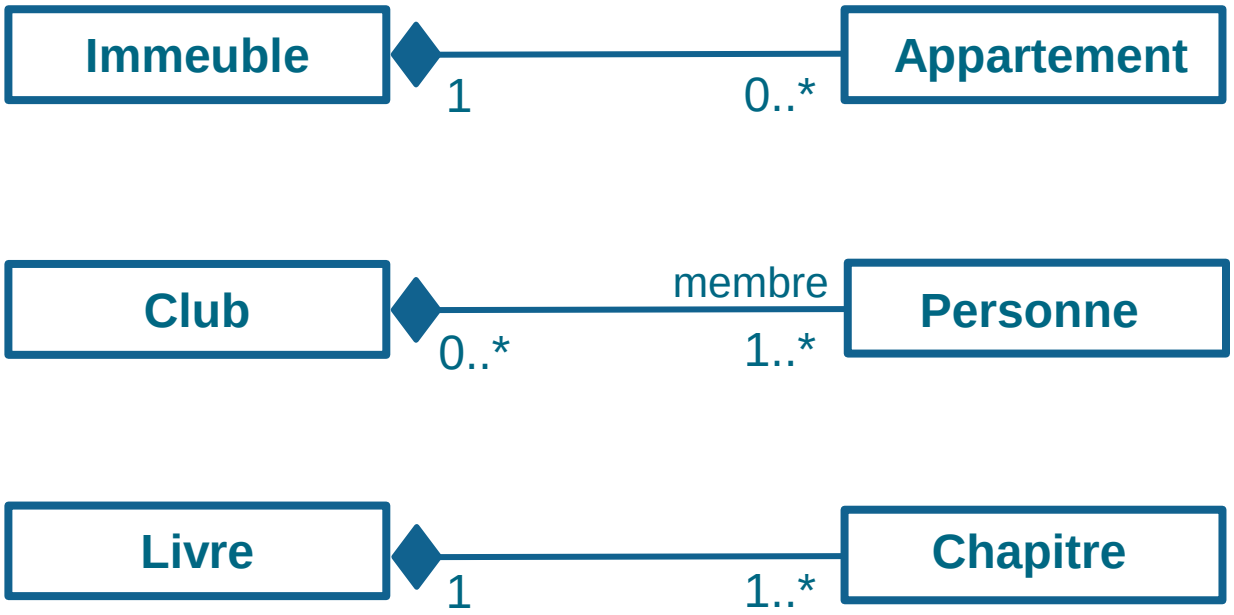
Pour être une composition, une agrégation doit vérifier les deux conditions suivantes :

- Cardinalité du côté du composite = 1 ou 0..1
- Si le composite meurt, alors, les composantes meurent



Exercices

Ces relations sont-elles bien des compositions ?



Plan

- Associations
- Décoration des associations
- Agrégations et Compositions
- Dépendances
- Modélisation UML

Dépendance

Dépendance entre deux classes A et B

- La classe A dépend de/utilise la classe B si, dans son code, elle fait appel aux méthodes de la classe B.
- Un élément A (le "**client**") dépend d'un élément B (le "**fournisseur**").
- A utilise les services de B.

Contrairement à l'association, la dépendance est directionnelle (flèche).

Contrairement à l'association, la dépendance décrit un lien temporaire.

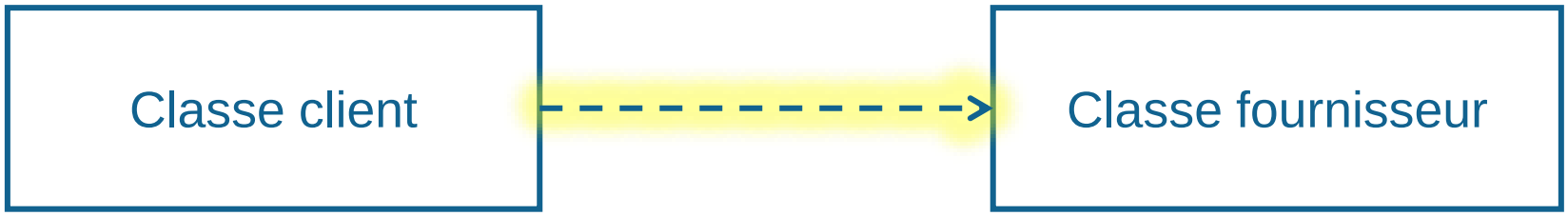
Exemple

- On doit pouvoir imprimer une facture. Dans la classe Facture, on trouve une méthode `imprime(Imprimante imp)` qui utilise les services de la classe Imprimante.

- **Facture = client** ; **Imprimante = fournisseur**

Dépendance

Notation : flèche en pointillés



Signification :

- (Client) dépend / a besoin de / utilise (Fournisseur)
- Il s'agit d'une relation temporaire, brève, pas permanente.
- (Client) n'a pas d'attribut de type (Fournisseur).
- (Fournisseur) est utilisé **comme argument/type de retour d'une méthode** de (Client).
- Si (Fournisseur) est modifiée, (Client) devra peut-être changer.

Plan

- Associations
- Décoration des associations
- Agrégations et Compositions
- Dépendances
- ▸ Modélisation UML

Modélisation = art (subjectif), pas science !

Le contexte est très important.

Étape 1 : Exemple de départ

Un film peut être inspiré d'un livre.

Modélisation = art (subjectif), pas science !

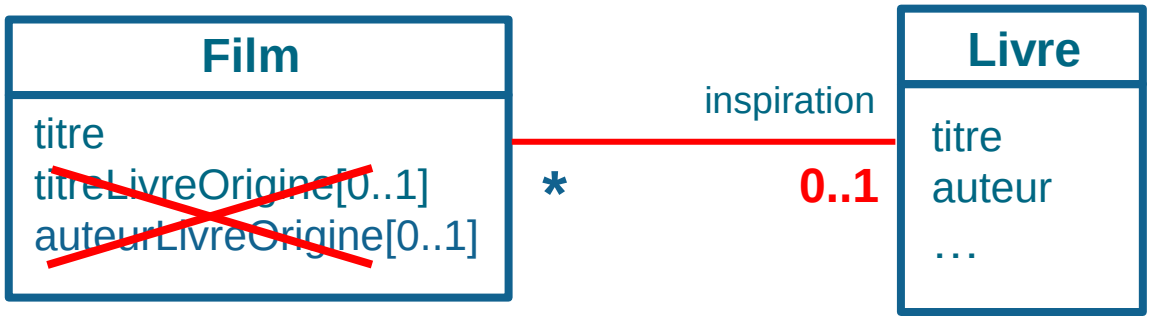
Étape 2 : *On veut retenir l'auteur du livre en question.*

Que pensez-vous de la solution suivante ?

Modélisation = art (subjectif), pas science !

Étape 2 : *On veut retenir l'auteur du livre en question.*

En quoi cette solution-ci est-elle meilleure ?



Étape 3 : *Et si certains films sont inspirés de plusieurs livres ?*

Modélisation = art (subjectif), pas science !

"Un bon modèle n'est pas un modèle où on ne peut plus rien ajouter mais un modèle où on ne peut plus rien enlever" (A. Saint-Exupéry)

Importance du choix :

- Qu'est-ce qui doit être précisé dans le modèle ?
- Que laisse-t-on à l'imagination / la décision du programmeur ?
- Quel niveau de détail (granularité) adopter dans le diagramme ?

- **D'où importance de bien lire l'énoncé !!!**
- **Danger : suivre sa propre pensée plutôt que l'énoncé !!!**
  - *Dans un cas réel, vous auriez l'occasion de discuter avec le client pour clarifier les choses ou lui proposer des alternatives que vous pensez préférables…*
  - *Dans le cas des exercices du cours, ce n'est pas possible.*
  - *Il faut donc suivre les idées de l'énoncé et pas les vôtres !*