



Chapitre 4

Relations particulières

Plan

- Association réflexive
- Classe-association
- Sous-association
- Association unidirectionnelle
- Association ordonnée
- Association disjointe
- Association qualifiée

Cas particuliers

On a vu 4 grands types de relations

- *De la plus faible à la plus forte :*
dépendance, association, agrégation, composition

Certains cas particuliers méritent d'être épinglés :

- les associations réflexives (une classe avec elle-même)
- les classes-associations (quand les associations ont des attributs)
- les sous-associations (association qui est un sous-ensemble d'une autre)
- quelques décorations supplémentaires :
 - les associations unidirectionnelles (restreindre la navigabilité)
 - les associations ordonnées (collections ordonnées)
 - les associations disjointes (un choix parmi plusieurs)
- les associations qualifiées (unicité sous certaines conditions)

Plan

- ▶ Association réflexive
- Classe-association
- Sous-association
- Association unidirectionnelle
- Association ordonnée
- Association disjointe
- Association qualifiée

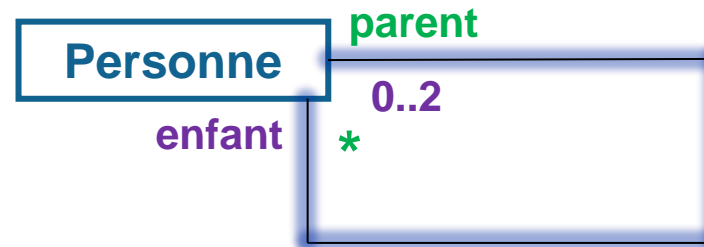
Association réflexive

= Association **entre une classe et elle-même**

Exemple : lien de parenté (biologique) entre des personnes

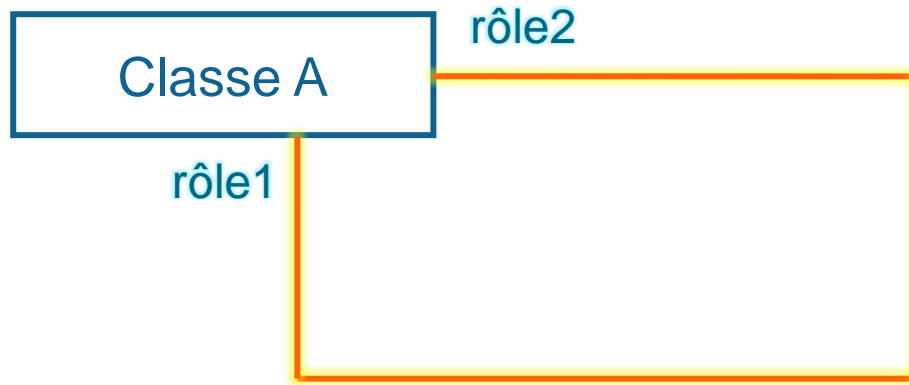


On ne présente la classe Personne qu'une seule fois.



*Note : le diagramme serait **incompréhensible sans les rôles** !*

Association réflexive



Spécification des rôles ou de la nature de l'association
obligatoire pour la compréhension

Sinon, **ambiguïté**

Association réflexive en Java

```
class Personne {  
    String nom;  
    ...  
    Personne [] enfants;  
    Personne [] parents;  
    ...  
}
```

On peut définir des classes possédant des attributs dont le type est la classe qu'on est en train de définir.

Parallèle : structure d'un chaînon dans une liste chaînée (voir cours d'OED et de C), dans laquelle on retrouve un pointeur vers un chaînon.

Association réflexive

Exercices

- Parmi les informaticiens d'une entreprise, certains sont des chefs de projet, qui dirigent jusqu'à un maximum de 6 informaticiens ; un informaticien dépend d'au plus un chef de projet.
- Dans une organisation, tout membre doit être parrainé par un autre membre (à moins qu'il ne s'agisse d'un membre fondateur) mais rien n'oblige un membre à parrainer quelqu'un : un membre peut ne pas parrainer, parrainer une personne ou en parrainer deux (un même membre ne peut pas parrainer plus de deux personnes).
- Une personne peut être mariée ou non à une autre personne.

Plan

- Association réflexive
- ▶ Classe-association
- Sous-association
- Association unidirectionnelle
- Association ordonnée
- Association disjointe
- Association qualifiée

Classe-association

Exemple

Au cours de l'année, les étudiants d'IG1 doivent réaliser divers travaux (par exemple : travail de DO, calculatrice matricielle de MTD...)

Chaque étudiant se voit attribuer une cote par travail.

Où indiquer cette cote ?



Une cote par étudiant \Rightarrow la cote serait la même pour tous les travaux de l'étudiant

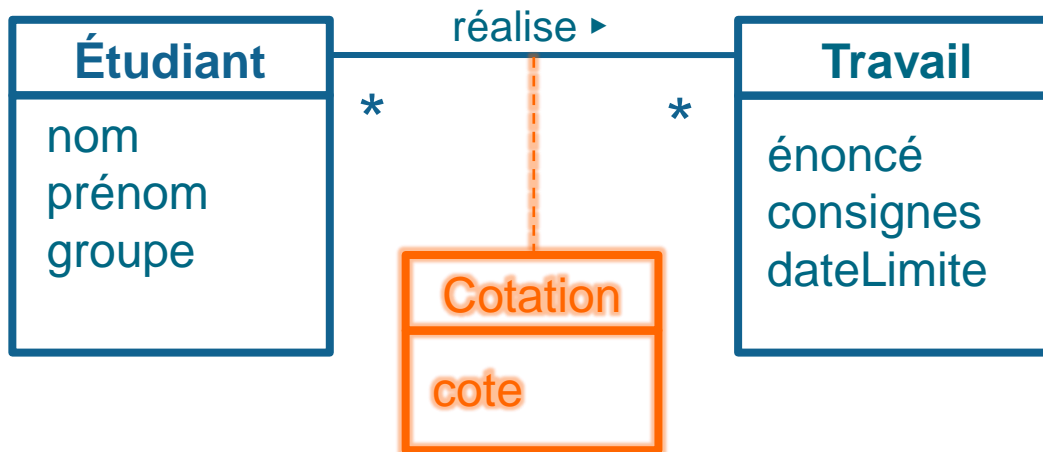
Un cote par travail \Rightarrow la même cote serait attribuée à tous les étudiants pour un même travail

Classe-association

La valeur de la cote dépend à la fois de l'étudiant et du travail

⇒ C'est une caractéristique de l'association

⇒ Classe-association avec pour attribut la cote

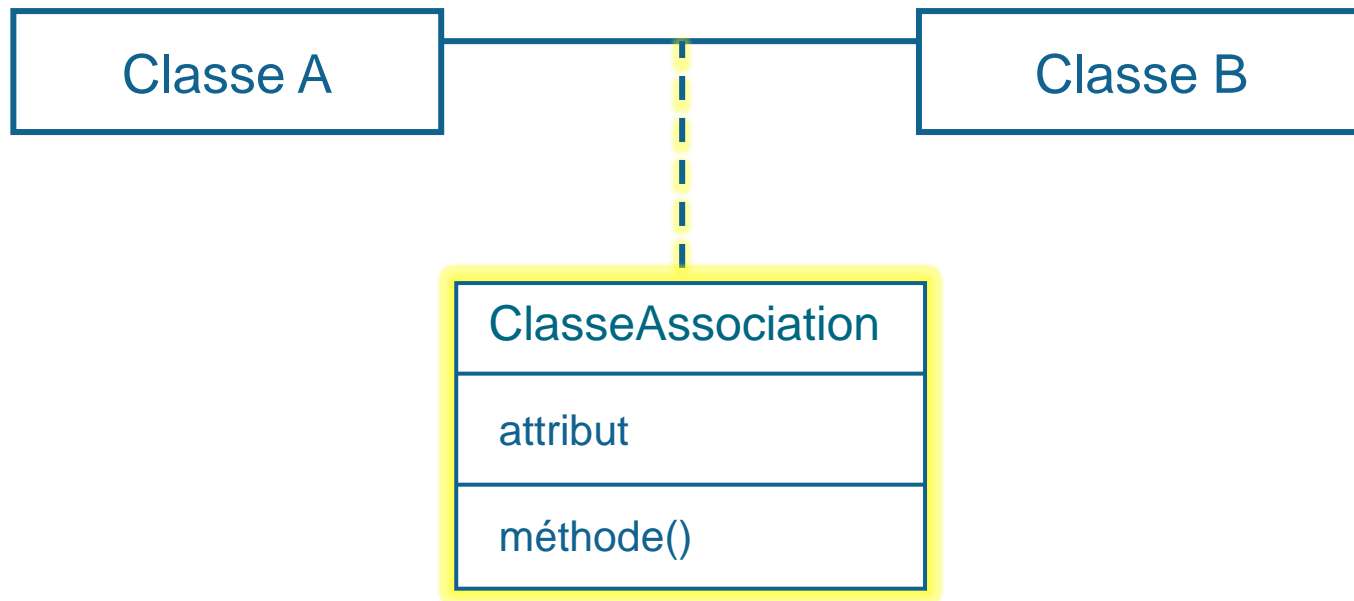


L'association devient
une classe (appelée
classe-association)

Classe-association

Quand un attribut ou une méthode ne peut être attaché(e) à aucune des deux classes d'une association, c'est qu'il s'agit d'une caractéristique de l'association.

⇒ on l'attache à l'association elle-même via une classe-association !



Classe-association en Java

Option 1 : tableaux parallèles dans Étudiant

```
class Étudiant {  
    ...  
    Travail [] travauxRéalisés;  
    double [] cotes;  
    ...  
}
```

Option 2 : classe-association utilisée dans Étudiant et/ou Travail

```
class Cotation {  
    Travail travail;  
    Étudiant étudiant;  
    double cote;  
}
```

```
class Étudiant {  
    ...  
    Cotation [] résultats;  
    ...  
}
```

Plus d'autres options encore...

Classe-association

Exercices

- Une personne travaille pour plusieurs sociétés. Selon la société dans laquelle elle est employée, son salaire de base et son titre sont différents.
- Les utilisateurs ont accès à plusieurs ordinateurs. En fonction de l'ordinateur qu'ils utilisent, ils ont des droits d'accès différents.

Classe-association

Exercices

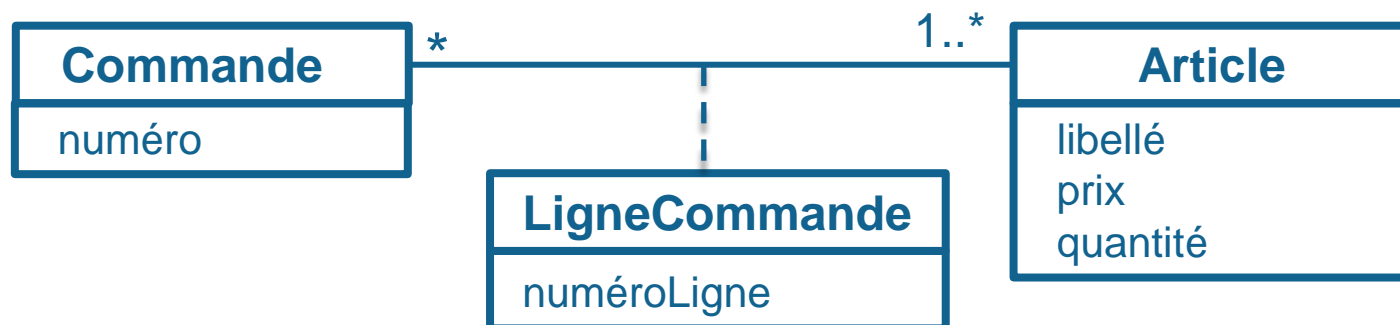
- Pour chaque participation d'un coureur à une course, on doit retrouver le coureur qui a participé, la course à laquelle il a participé, la date de son inscription et le numéro de dossard qu'il a reçu pour la course. Un coureur peut bien sûr s'inscrire à plusieurs courses. Pour chaque coureur ayant participé à une course, on doit pouvoir retrouver sa position à l'arrivée.

Classe-association

Exercices

- Où est l'erreur dans le diagramme UML suivant ?

Une commande porte un numéro et concerne potentiellement plusieurs articles. Pour chaque article commandé, on écrit une ligne (numérotée) dans la commande ; on indique le libellé de l'article, son prix et la quantité commandée.



Plan

- Association réflexive
- Classe-association
- ▶ Sous-association
- Association unidirectionnelle
- Association ordonnée
- Association disjointe
- Association qualifiée

Sous-association

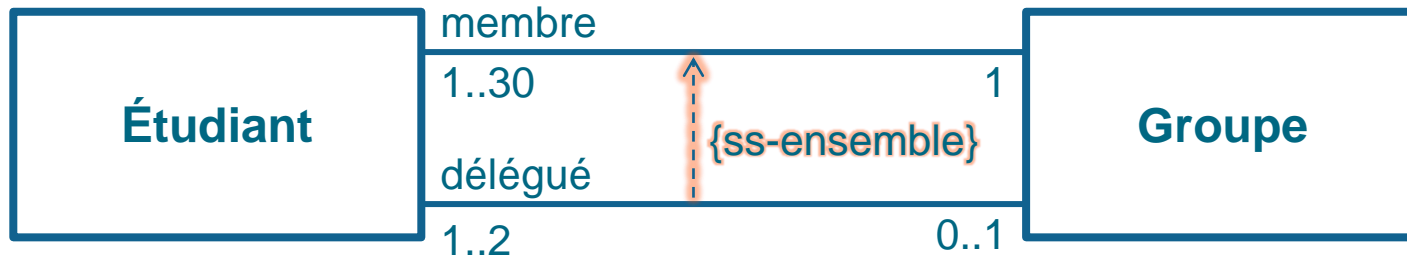
Exemple. Les étudiants de IG1 sont divisés en groupes. Dans chaque groupe, il y a un ou deux délégué(s).



D'après ce diagramme, un étudiant pourrait être membre du groupe B tout en étant délégué pour le groupe C !

Sous-association

On peut préciser que l'association "délégué" doit être un sous-ensemble de l'association "membre".

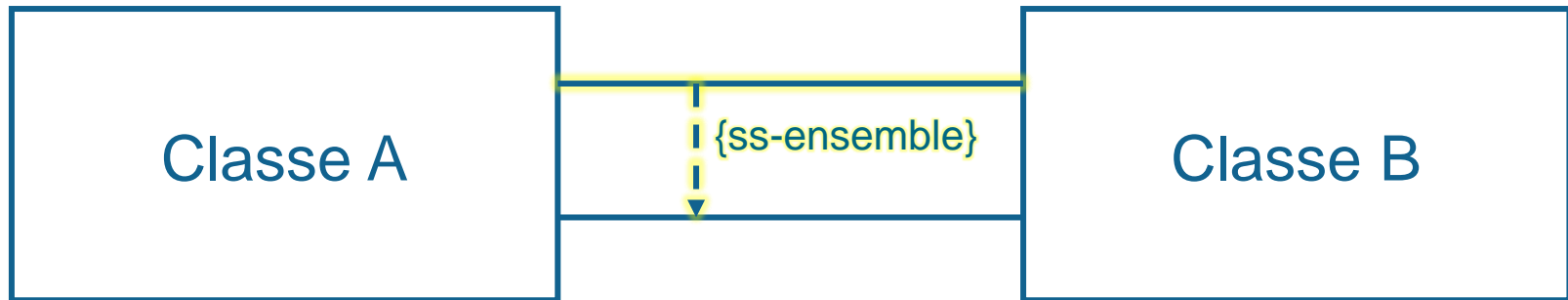


Cela signifie que, s'il y a un lien de type "délégué" entre un étudiant **e** et un groupe **g**, alors il doit y avoir un lien de type "membre" entre cet étudiant **e** et ce groupe **g**.

(L'ensemble des liens de type "délégué" est un sous-ensemble de l'ensemble des liens de type "membre")

Association avec sous-ensemble

La décoration {sous-ensemble} permet d'indiquer qu'une association doit être un sous-ensemble de l'autre association.



Pour qu'un objet de A soit lié à un objet B selon l'association du dessus, il faut que ces objets soient liés selon l'association du dessous.

Sous-association en Java

Option 1 : tableaux distincts + vérification lors de l'ajout d'un délégué (l'étudiant à ajouter doit être membre).

```
class Groupe {  
    Étudiant [] membres;  
    Étudiant [] délégués;  
    ...  
}
```

Option 2 : tableau de booléens pour indiquer l'appartenance à la sous-association

```
class Groupe {  
    Étudiant [] membres;  
    boolean [] estDélégué;  
    ...  
}
```

Plus d'autres options encore...

Plan

- Association réflexive
- Classe-association
- Sous-association
- ▶ Association unidirectionnelle
- Association ordonnée
- Association disjointe
- Association qualifiée

Association unidirectionnelle

Rappel sur l'association



Navigabilité dans les deux directions :

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

On peut rendre la navigabilité unidirectionnelle en ajoutant une flèche.

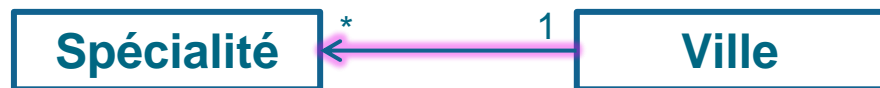
○ **"Une spécialité voit sa ville" :**

- Attribut Ville villeOrigine dans Spécialité.
- Mais **pas l'inverse** !



○ **"Une ville voit ses spécialités" :**

- Attribut Spécialité [] spécialités dans Ville
- Mais **pas l'inverse** !



Plan

- Association réflexive
- Classe-association
- Sous-association
- Association unidirectionnelle
- ▶ Association ordonnée
- Association disjointe
- Association qualifiée

Association ordonnée

Dans une entreprise, chaque jour, les employés reçoivent une liste de tâches à accomplir.



Dans une entreprise, chaque jour, les employés reçoivent une liste de tâches à accomplir *dans un ordre précis*.



- La contrainte est traduite par l'annotation {ordonné}.

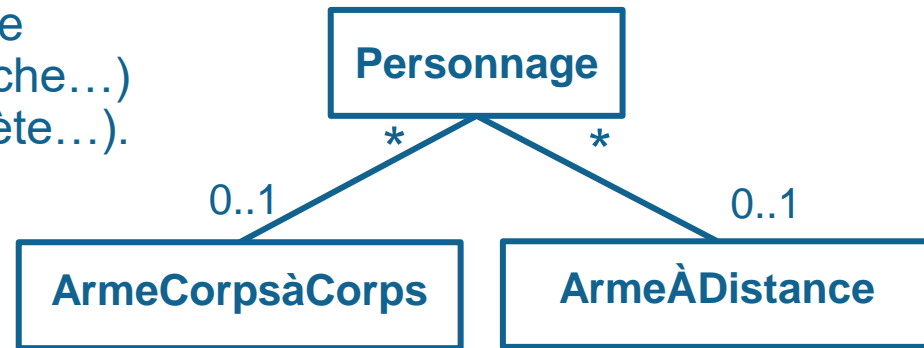
Note. Les tableaux sont des structures ordonnées mais Java possède d'autres structures, parfois mieux adaptées, qui ne le sont pas.

Plan

- Association réflexive
- Classe-association
- Sous-association
- Association unidirectionnelle
- Association ordonnée
- ▶ Association disjointe
- Association qualifiée

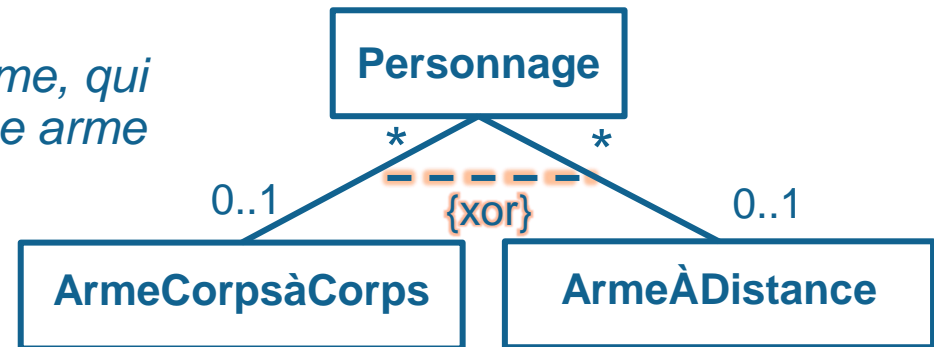
Association disjointe

Dans un jeu, un personnage peut combattre avec une arme de corps à corps (épée, hache...) et/ou avec une arme à distance (arc, arbalète...).



*Contrainte : "Le personnage utilise une arme, qui est **soit** une arme de corps à corps **soit** une arme à distance."*

Disjonction exclusive
= **1 et 1 seule option**



En Java : s'assurer qu'un seul des attributs de **Personnage** (entre **armeCorpsàCorps** et **armeÀDistance**) possède une valeur.

Plan

- Association réflexive
- Classe-association
- Sous-association
- Association unidirectionnelle
- Association ordonnée
- Association disjointe
- ▶ Association qualifiée

Association qualifiée

Sur un échiquier, il y a 64 cases (A1, ..., A8, B1, ..., H8) qui peuvent être occupées ou libres.



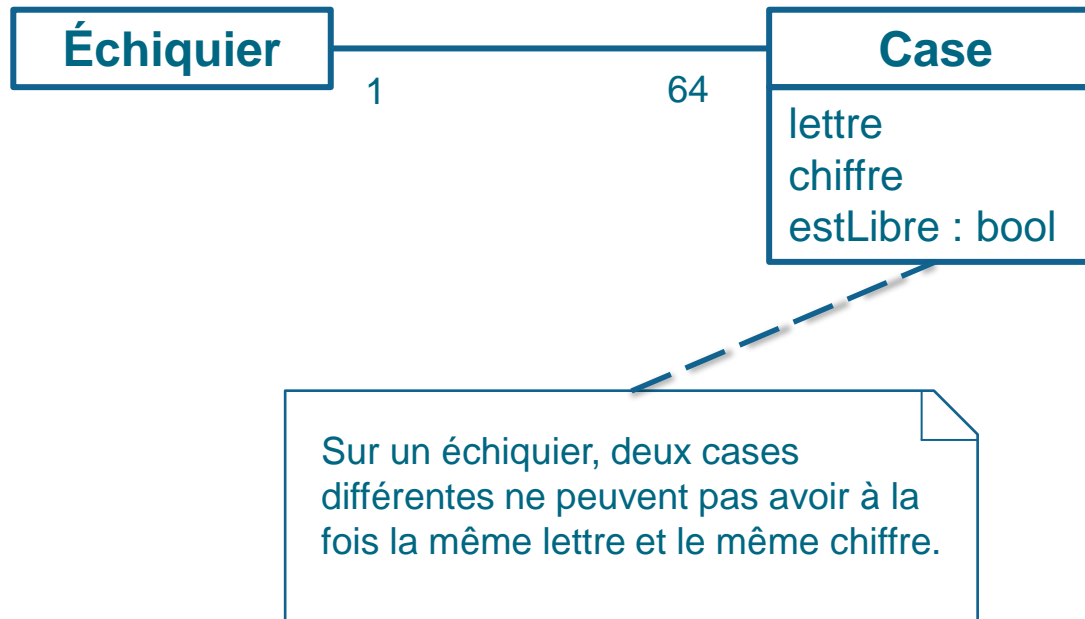
Avec ce schéma,
plusieurs cases peuvent avoir la même combinaison lettre-chiffre sur un même échiquier !

(exemple : plusieurs cases B2 sur le même échiquier)

Comment ajouter une contrainte pour éviter ce problème ?

Association qualifiée

Première option (générique) : ajouter un commentaire UML.



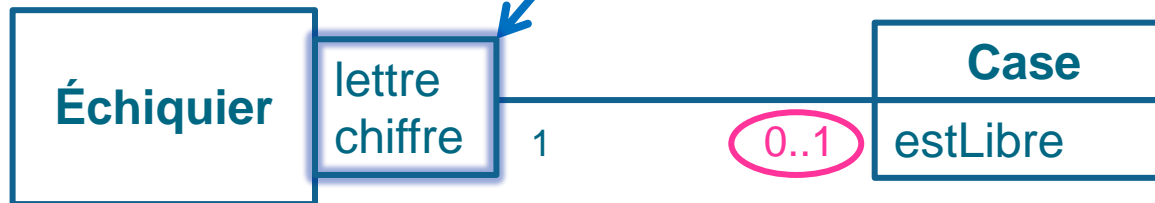
Association qualifiée

Seconde option :



La clef qui identifie chacune des cases **sur un même échiquier** est extraite de la classe et placée près de l'autre classe.

Traduction par une association qualifiée :



La multiplicité est changée !

Interprétation :

À partir d'un échiquier et d'une combinaison d'une lettre + un chiffre

⇒ On arrive à maximum une seule case

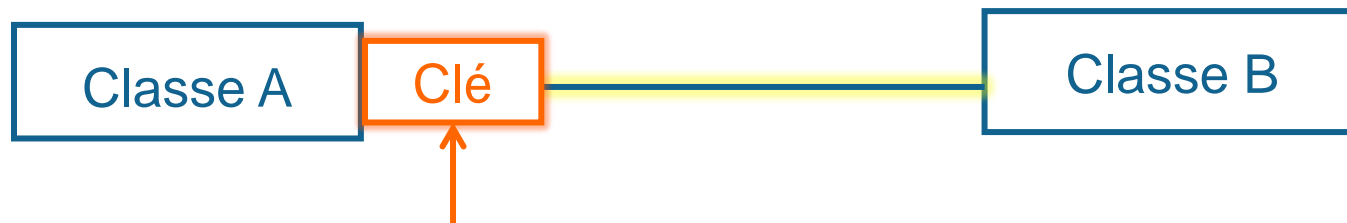
⇒ On perd l'information "64 cases par échiquier"

Association qualifiée

Une association qualifiée permet de mettre en évidence ce qui caractérise **un objet parmi sa famille**.

- C'est-à-dire ce qui identifie un objet de classe B *parmi l'ensemble des objets de classe B associés à un même objet de Classe A*.

Le(s) attribut(s) qui identifie(nt) l'objet de manière unique au sein des objets de sa famille forme(nt) la **clé** ; celle-ci est placée près de l'autre classe.



Identifie un objet de type B de manière unique
parmi tous les objets de type B liés au même objet
de type A

Association qualifiée en Java

```
class Échiquier {  
    ...  
    Case [] cases;  
    ...  
}
```

Quand on construit / modifie le tableau cases, vérifier qu'il n'y a pas de doublons (= deux cases avec la même lettre et le même chiffre).

Association qualifiée = contrainte à traduire par des vérifications dans le code.

Association qualifiée

Autres exemples

