

Attributs

Une classe a des **attributs**, qui associent à chaque objet de la classe certaines **valeurs**.

Les valeurs d'un attribut *A* sont des éléments d'un ensemble appelé le **domaine** de *A*.

Ce domaine doit être **primitif** (ou **atomique**) : Nombre, Chaîne, ou Date.

Le fait que les domaines doivent être primitifs est dû aux contraintes de stockage des bases de données.

Clés

Un **identifiant** ou une **clé** *K* pour une classe *C* est un ensemble d'attributs de *C*

- dont les valeurs **déterminent les valeurs de tous les attributs**,
- et qui est **minimal** pour cette propriété : si on enlève un attribut à *K* alors cette propriété n'est plus vérifiée.

Chaque classe a une clé appelée **clé primaire**.

Aucun attribut d'une clé primaire ne peut être nul !

Parfois une classe a d'autres clés, appelées **clés secondaires**, qui jouent un rôle beaucoup moins important que la clé primaire.

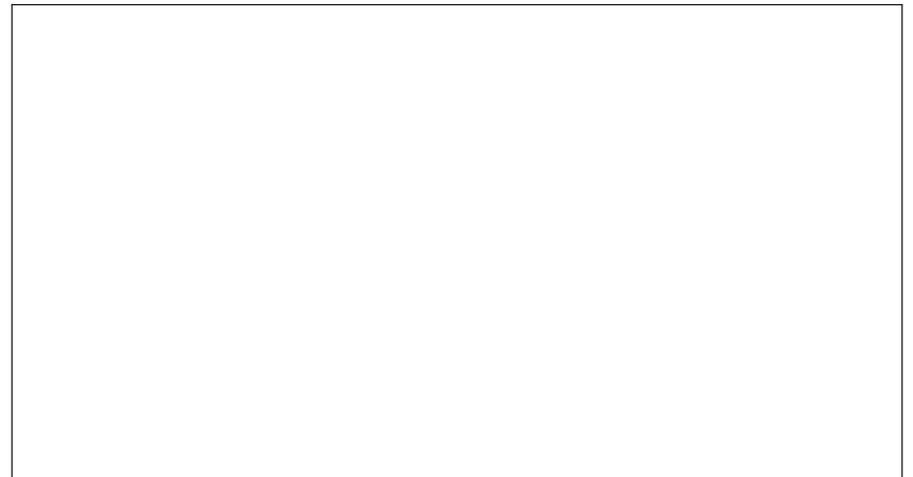
Représentation graphique



Remarques.

- Les **clés secondaires** ne sont pas représentées, elles peuvent être indiquées en commentaires.
- Contrairement aux diagrammes de classes pour la programmation, il n'y a pas de troisième partie pour les méthodes.
- Pour alléger, on omet souvent les domaines des attributs. Un **dictionnaire de données** peut être fourni pour définir les domaines et les contraintes des attributs.

Représentation graphique : exemples



Cardinalités : représentation graphique



Cardinalités : mono/multi (1/2)

Remarques.

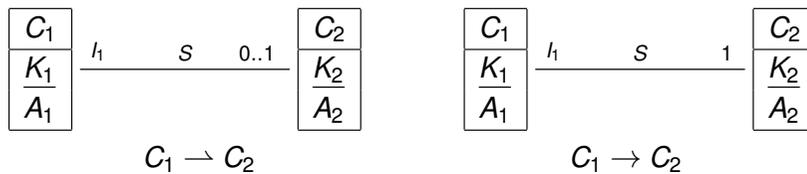
- Les cardinalités les plus courantes sont 0..*, 1..*, 0..1, 1. On note aussi * pour 0..*.
- Les cardinalités 0..1 et 1 sont appelées **mono** et les cardinalités 0..*, 1..*, 0..x et 1..x, avec $x > 1$, sont appelées **multi**. Donc une association peut être **mono-mono** (rare), ou **multi-mono**, ou **multi-multi**.

Cardinalités : mono/multi (2/2)

Remarques.

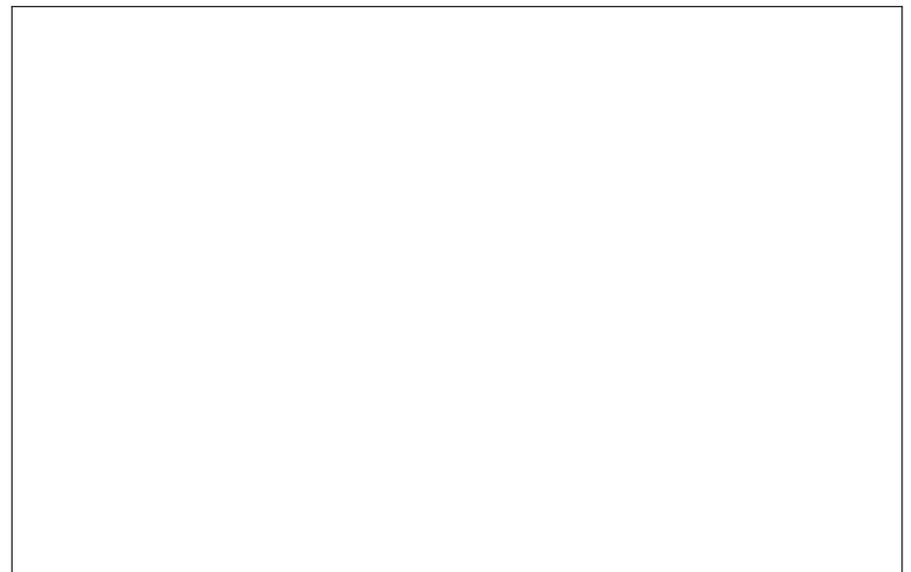
- Le cas **multi-mono**, avec l_1 **multi** et l_2 **mono**, signifie que chaque $e_1 \in C_1$ est associé à **au plus un** $e_2 \in C_2$.

Au sens mathématique, c'est une **fonction partielle** de C_1 vers C_2 si l_2 est 0..1 et c'est une **fonction totale** de C_1 vers C_2 si l_2 est 1.



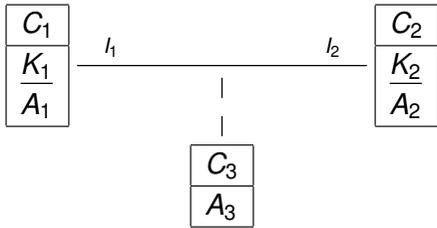
- Lorsqu'on hésite entre deux cardinalités il vaut mieux, en général, choisir la moins contraignante : choisir 0..* plutôt que 1..*, et choisir 0..1 plutôt que 1.

Cardinalités : exemple



Classes-associations

Une **classe-association** est une **association** qui de plus se comporte comme une classe : elle a des attributs et elle peut être associée à d'autres classes.



Remarque.

- Une classe-association n'a pas besoin d'identifiant : chaque objet de la classe C_3 est identifié par le couple (c_1, c_2) d'objets qu'il relie (avec c_1 dans C_1 et c_2 dans C_2).

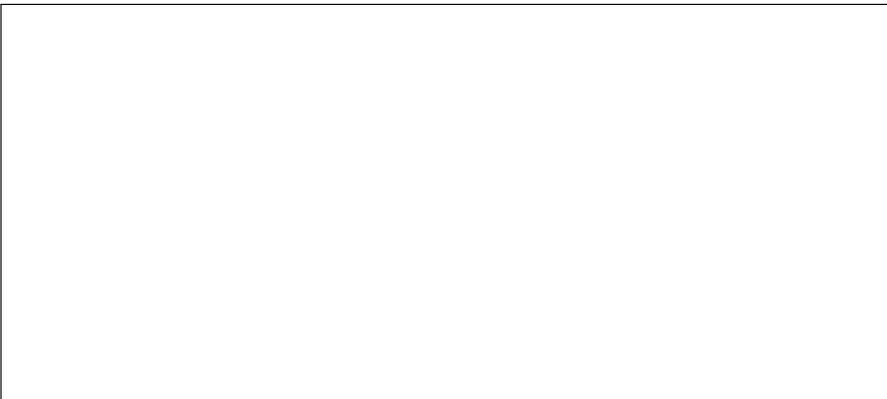
Donc l'identifiant de C_3 est formé de l'identifiant de C_1 et de l'identifiant de C_2 .

Classes-associations : exemple mono-multi (rare) (2/2)

On souhaite maintenant noter la date de l'emprunt.

L'association « a emprunté » devient une **classe-association** Emprunts avec un attribut dateE.

Rappel : pas de clé primaire pour Emprunts, chaque emprunt est identifié par le livre et l'adhérent. Cependant la cardinalité 0..1 du côté des Adhérents permet de simplifier : chaque emprunt est déterminé par le livre emprunté (cf. cours 7).



Classes-associations : exemple multi-multi (1/2)

L'inscription d'étudiants à des cours. D'abord sans date : c'est une **association multi-multi**.



Puis **avec une date d'inscription** : c'est une **classe-association**. Chaque inscription est identifiée par l'étudiant ET le cours, la date d'inscription est un autre attribut.



Classes faibles

Une **classe faible** est une classe C_2 qui dépend d'une autre classe C_1 pour déterminer son identifiant.

La classe faible C_2 est associée à la classe C_1 par une association S avec toujours une cardinalité 1 du côté de C_1 , donc on l'**omet**.

Dans ce cas on dit que l'association S est une **référence de C_2 vers C_1** .

Puisque tout objet de C_2 est lié à exactement un objet de C_1 , il est inutile de répéter dans C_2 les attributs de C_1 .

Les attributs soulignés dans C_2 forment le **complément d'identifiant** de C_2 : du point de vue logique l'identifiant de C_2 est formé de l'identifiant de C_1 auquel on ajoute ce complément d'identifiant.

