

DUT MMI – IUT de Marne-la-Vallée

02/05/2017

M2203 – Bases de données

Cours 4

Méthodes de modélisation

Sources

- Cours de Tony Grandame à l'IUT de Marne-la-Vallée en 2010-2011

- Cours de Mathieu Mangeot, IUT de Savoie

<http://jibiki.univ-savoie.fr/~mangeot/Cours/BasesDeDonnees.pdf>

- Cours de Fabrice Meuzeret, IUT de Troyes

<http://195.83.128.55/~fmeuzeret/vrac/>

- Livre de Laurent Audibert : *Bases de données - de la modélisation au SQL*

Version partielle sur :

<http://laurent-audibert.developpez.com/Cours-BD/html/index.php>

Plan du cours 4 – Modélisation, SQL avancé (suite)

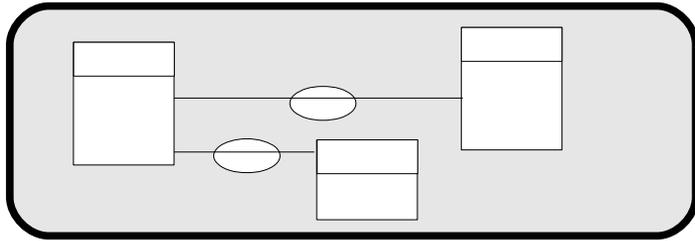
- Résumé des épisodes précédents
- Modélisation MERISE et UML

Plan

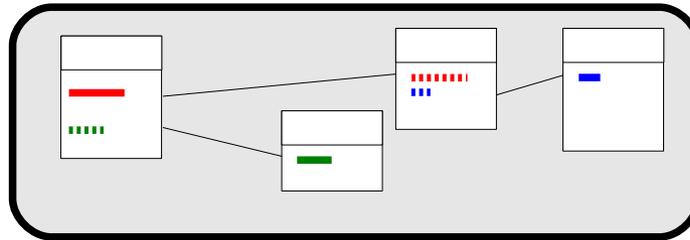
- Résumé des épisodes précédents
- Modélisation MERISE et UML

Épisodes précédents

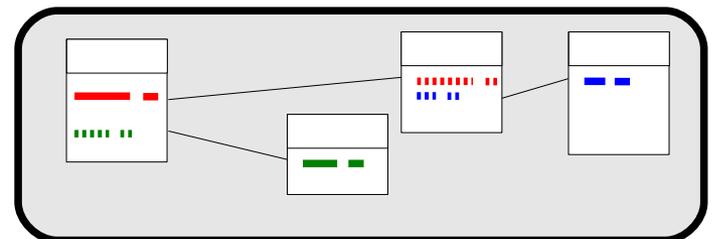
Modèle entité-association
(modèle conceptuel des données)



Modèle logique des données

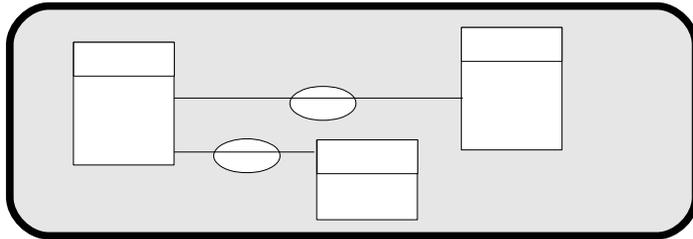


Modèle physique des données



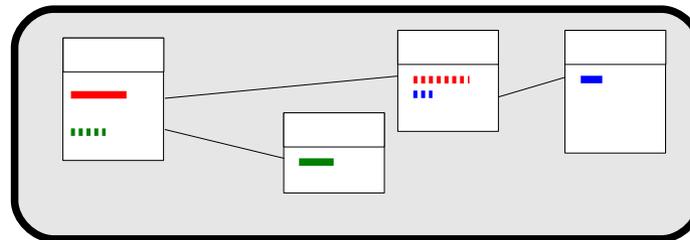
Épisodes précédents

Modèle entité-association
(modèle conceptuel des données)

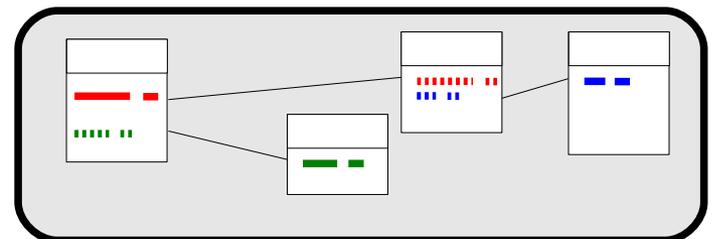


Méthode MERISE :
méthode d'analyse, de
conception et de réalisation
de systèmes d'informations.

Modèle logique des données

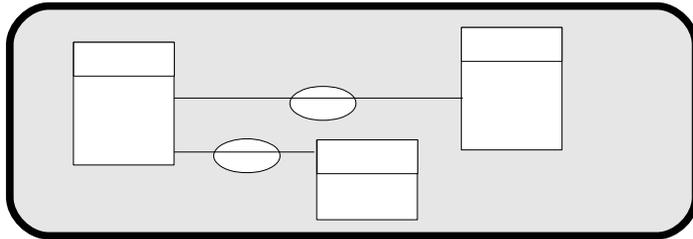


Modèle physique des données



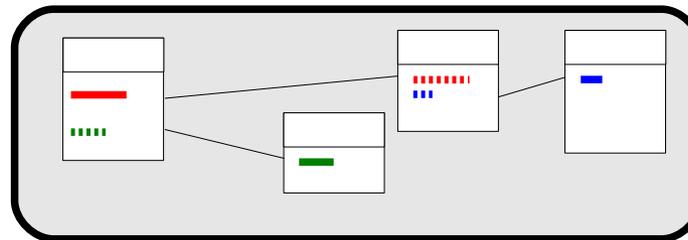
Épisodes précédents

Modèle entité-association
(modèle conceptuel des données)



Méthode MERISE :
méthode d'analyse, de
conception et de réalisation
de systèmes d'informations.

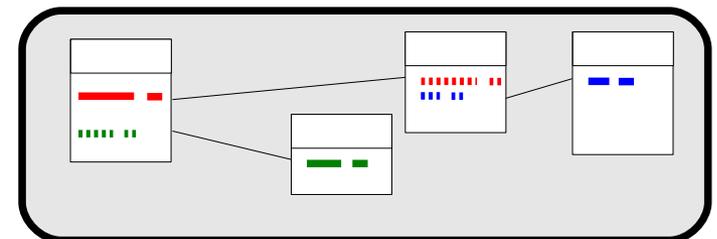
Modèle logique des données



Méthode MERISE **pas seulement**
pour les bases de données :

- Exprimer le besoin
- Créer les modèles conceptuels
- Créer les modèles logiques
- Créer les modèles physiques

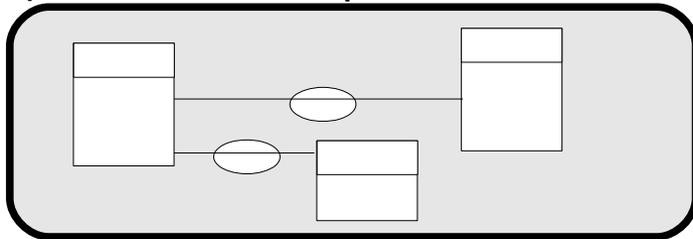
Modèle physique des données



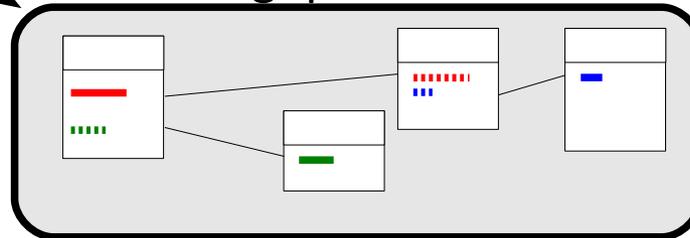
Épisodes précédents

Cahier des charges

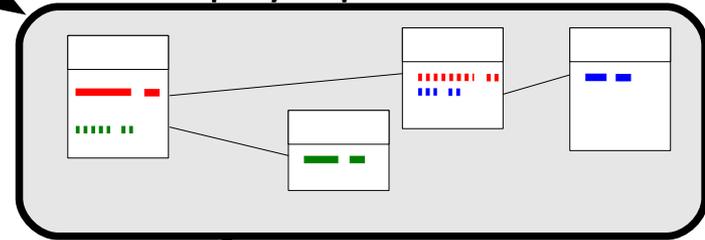
Modèle entité-association
(modèle conceptuel des données)



Modèle logique des données



Modèle physique des données



Langage SQL

Méthode MERISE :
méthode d'analyse, de
conception et de réalisation
de systèmes d'informations.

Méthode MERISE **pas seulement**
pour les bases de données :

- Exprimer le besoin
- Créer les modèles conceptuels
- Créer les modèles logiques
- Créer les modèles physiques

Plan

- Résumé des épisodes précédents
- Modélisation MERISE et UML

Modélisation MERISE et UML

Modélisation des données :

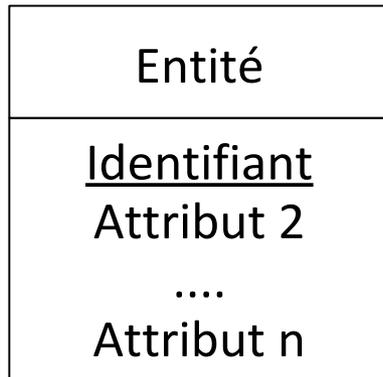
- Modèle conceptuel des données de MERISE
- **UML** (Unified Modeling Language) :
 - autre langage de modélisation
 - langage dédié à l'objet
 - plusieurs types de diagramme, dont un utile en bases de données :
le **diagramme de classes**

Lien / traduction entre :

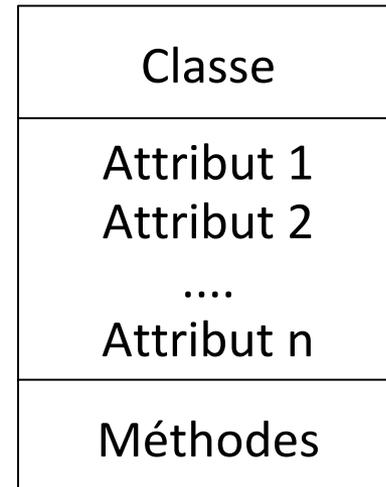
- Modèle conceptuel des données de MERISE
- Diagramme de classes UML

Modélisation MERISE et UML : entité / classe

MERISE

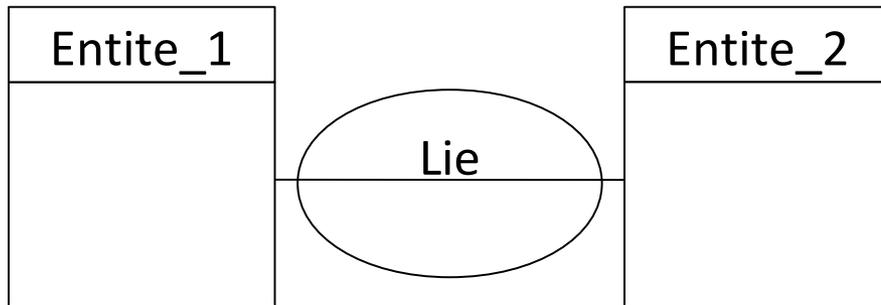


UML

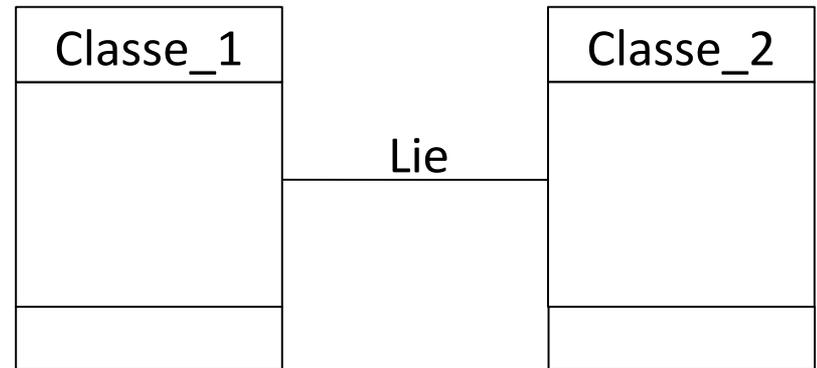


Modélisation MERISE et UML : association

MERISE



UML



Modélisation MERISE et UML : cardinalités

MERISE

Lien vers 0 ou 1 : **0,1**

Lien vers 1 : **1,1**

Lien vers 0 ou plusieurs : **0,n**

Lien vers 1 ou plusieurs : **1,n**

UML

Lien vers 0 ou 1 : **0..1**

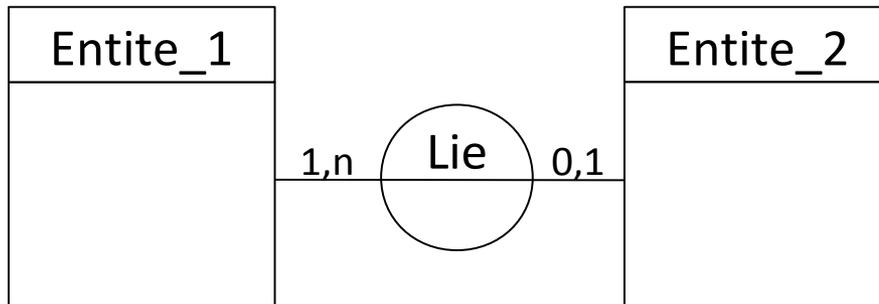
Lien vers 1 : **1**

Lien vers 0 ou plusieurs : *****

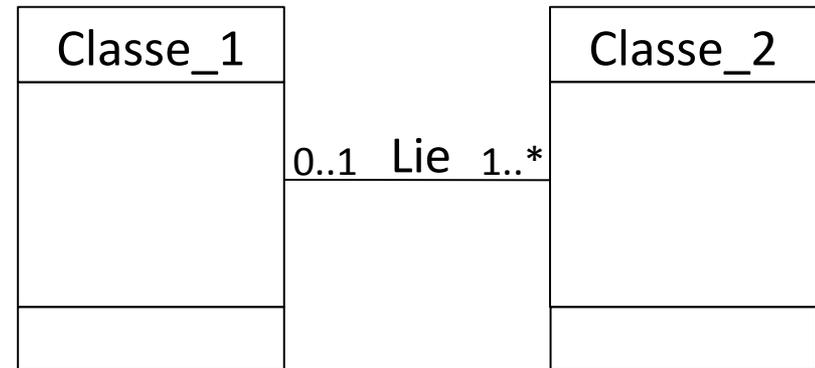
Lien vers 1 ou plusieurs : **1..***

Modélisation MERISE et UML : association & cardinalités

MERISE

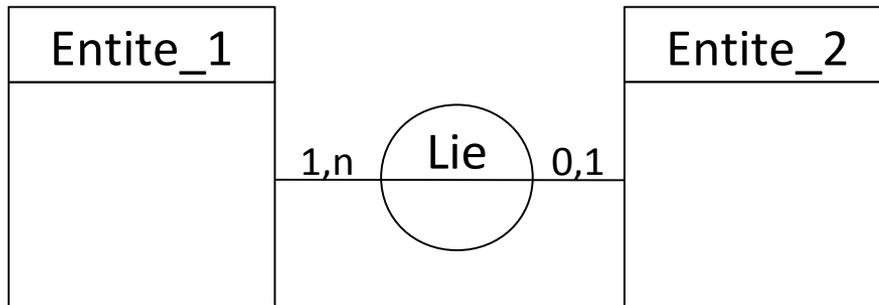


UML

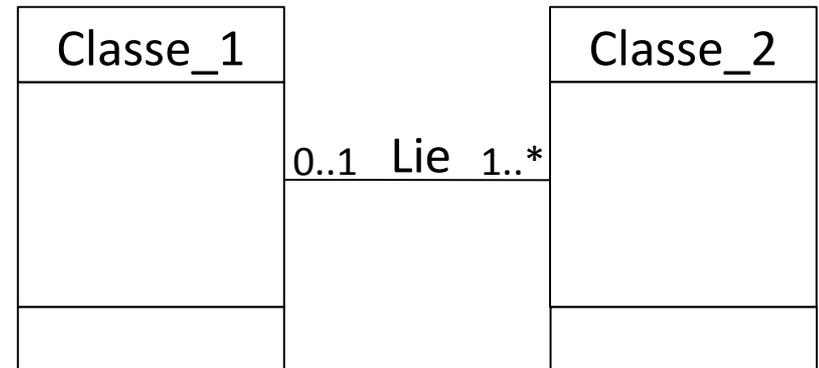


Modélisation MERISE et UML : association & cardinalités

MERISE



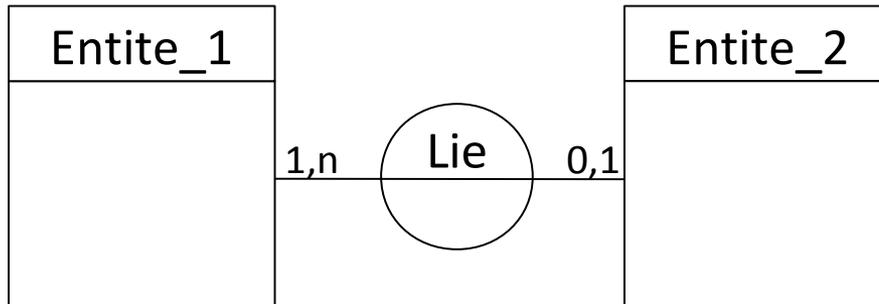
UML



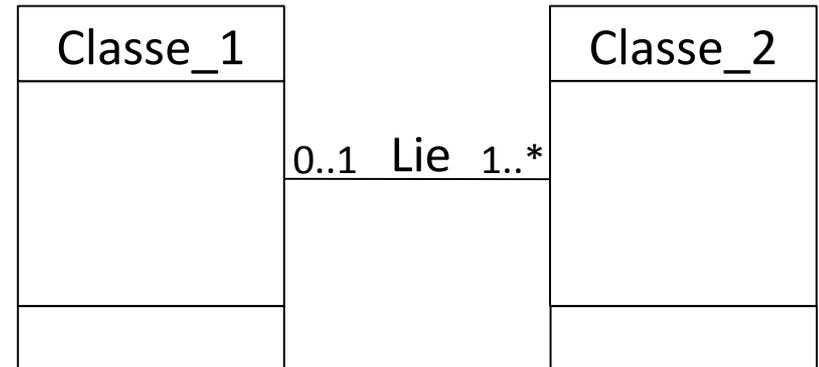
inversion de sens des
cardinalités !

Modélisation MERISE et UML : association & cardinalités

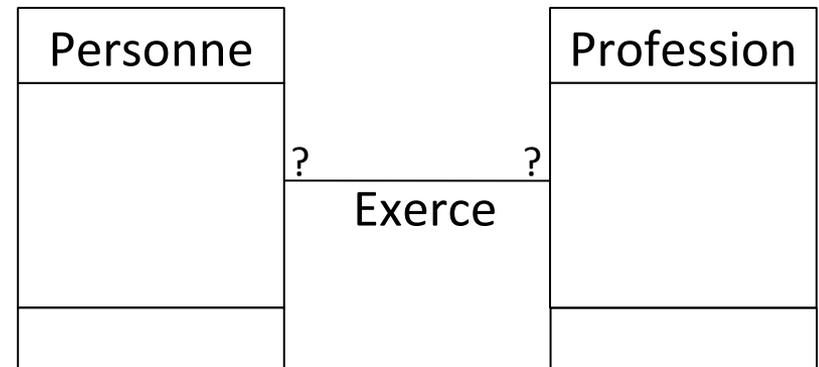
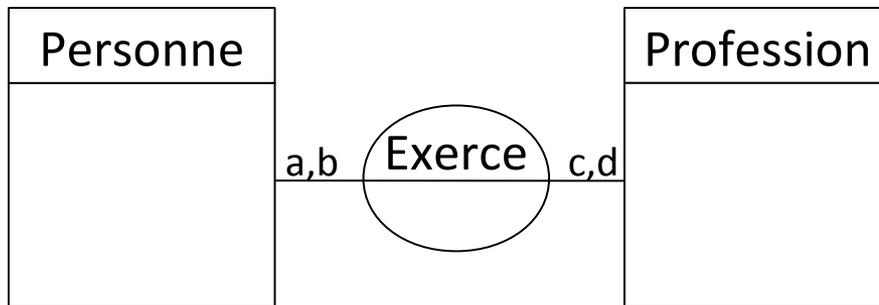
MERISE



UML

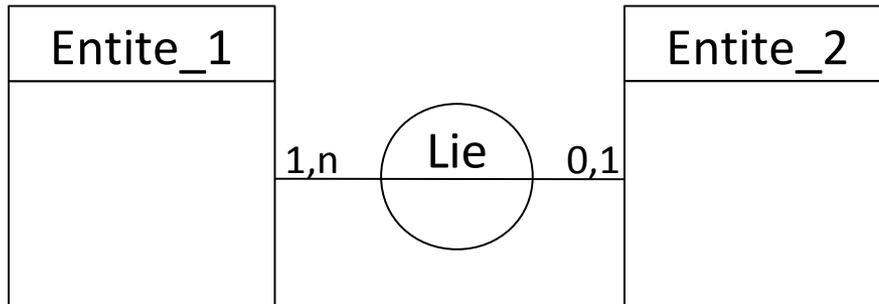


Exemple :

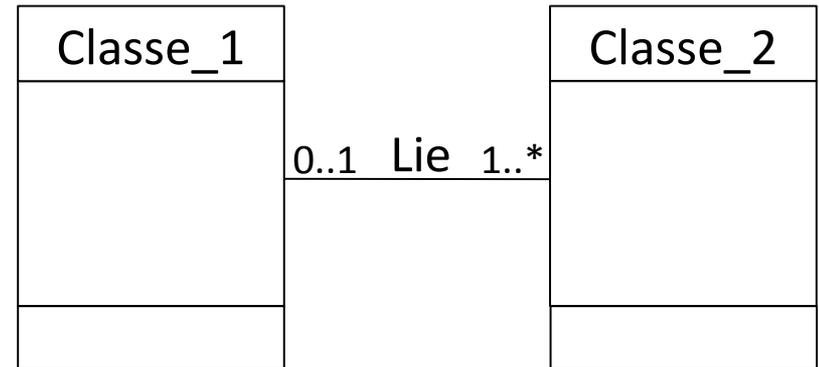


Modélisation MERISE et UML : association & cardinalités

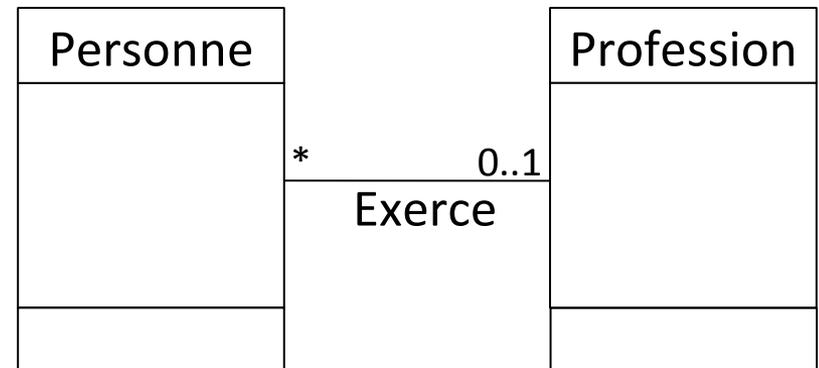
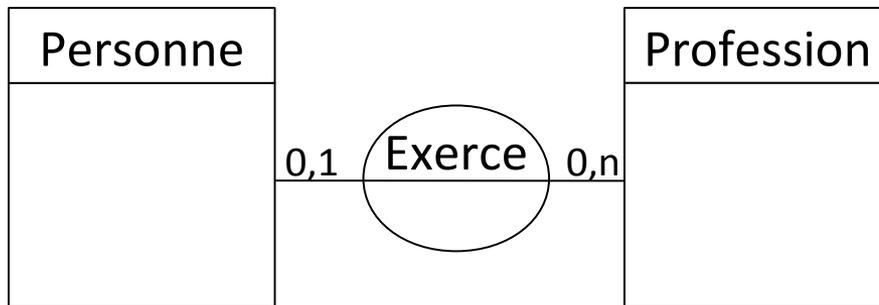
MERISE



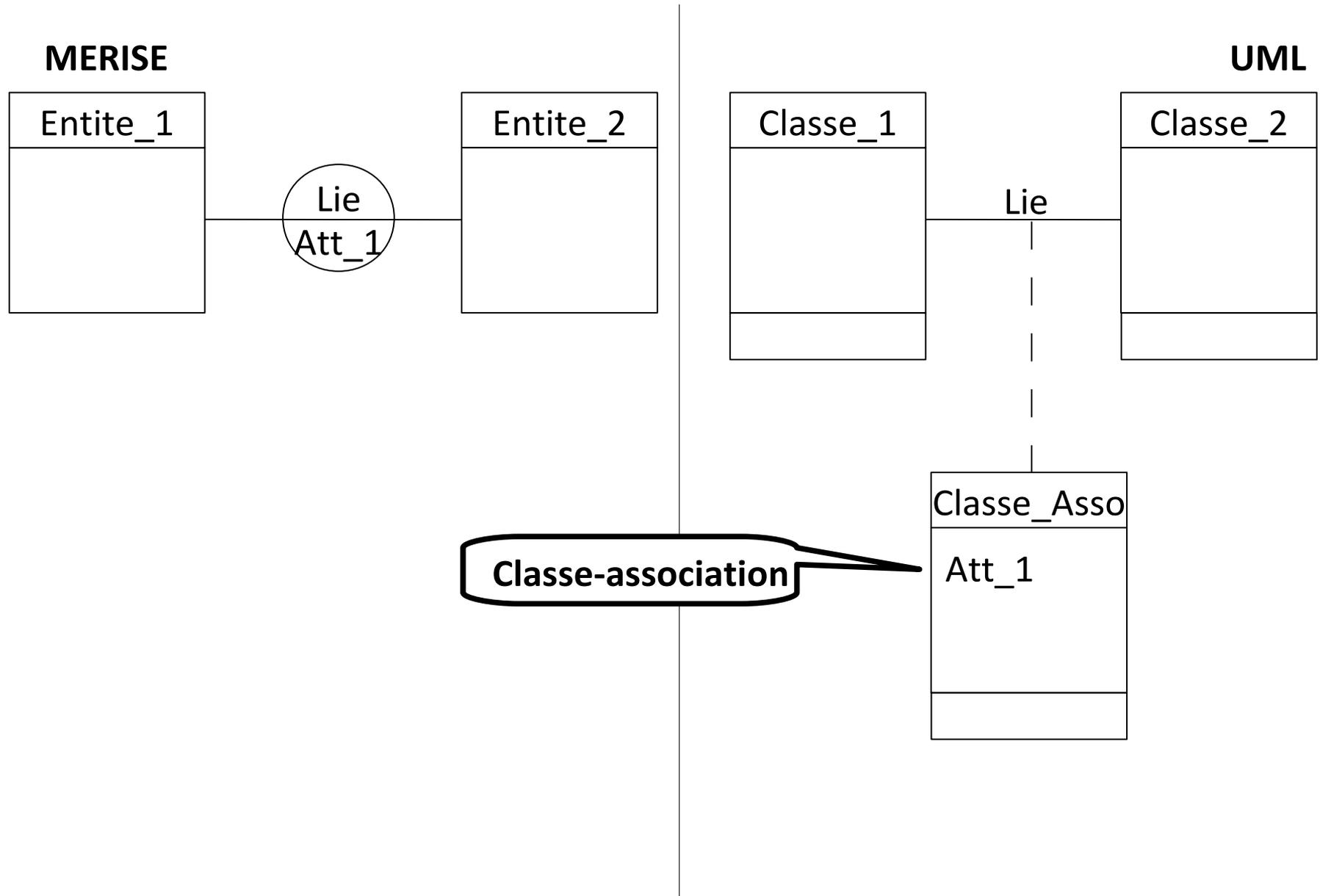
UML



Exemple :



Modélisation MERISE et UML : association avec attributs



Les “plus” d'UML - Agrégation

Agrégation :

- Associations **non symétriques**
- Une classe joue un **rôle prépondérant** par rapport à l'autre

Deux formes d'agrégation :

- composition
- agrégation partagée

Les “plus” d'UML - Agrégation

Agrégation :

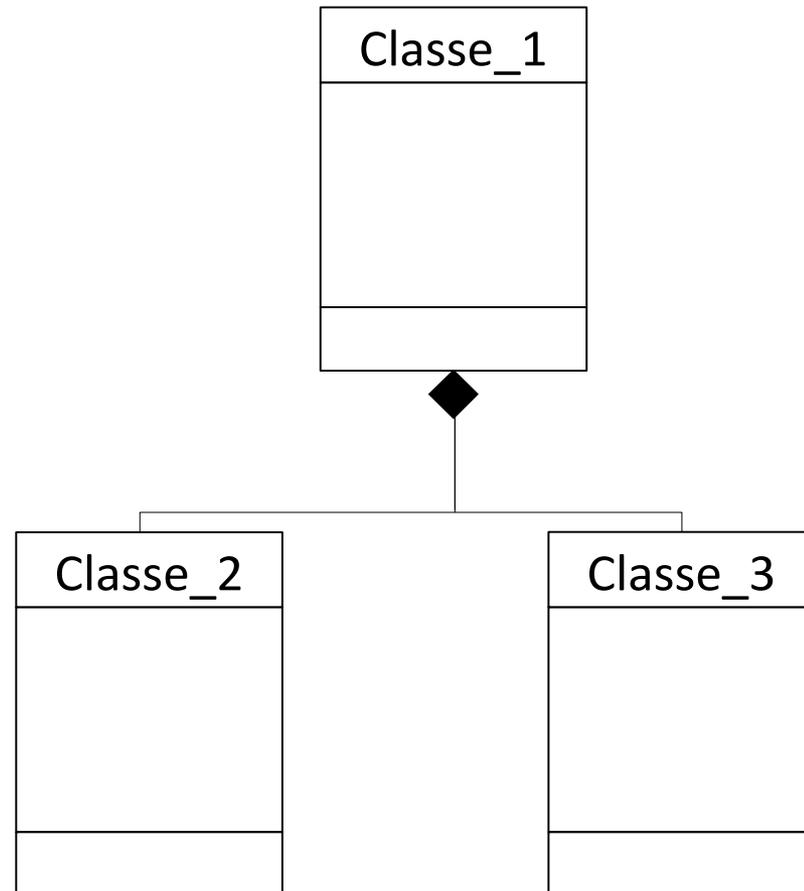
- Associations **non symétriques**
- Une classe joue un **rôle prépondérant** par rapport à l'autre

Deux formes d'agrégation :

- **composition**
- agrégation partagée

Composition : une classe Classe_2 est sous-ensemble d'une autre, Classe_1

→ **losange plein**



Les “plus” d'UML - Agrégation

Agrégation :

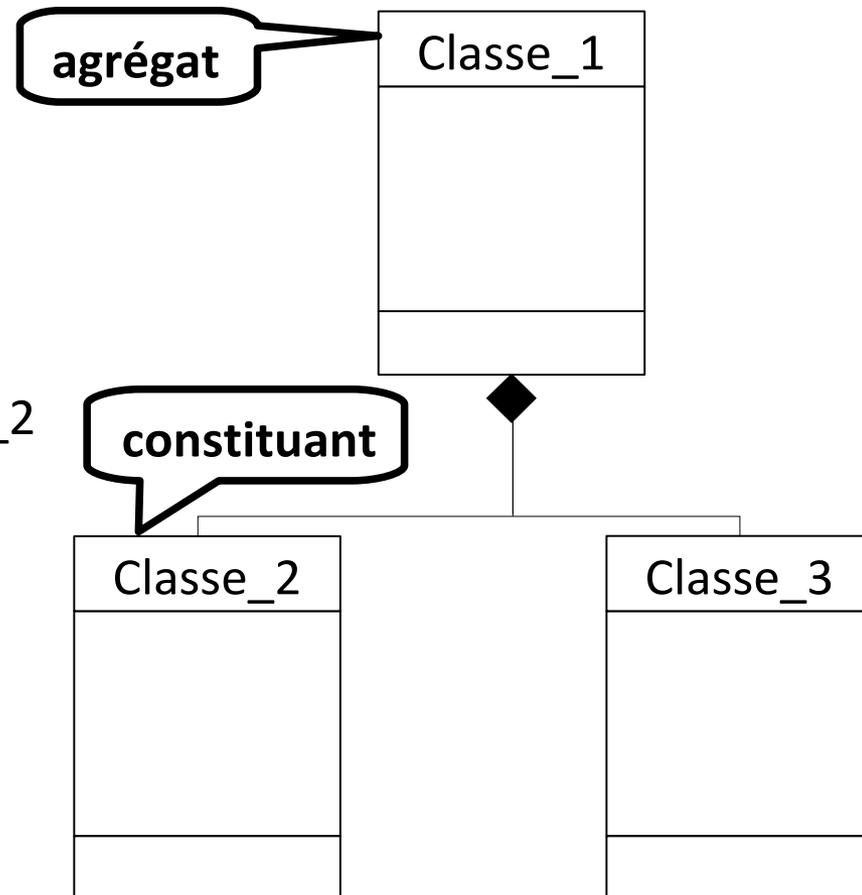
- Associations **non symétriques**
- Une classe joue un **rôle prépondérant** par rapport à l'autre

Deux formes d'agrégation :

- **composition**
- agrégation partagée

Composition : une classe Classe_2 est sous-ensemble d'une autre, Classe_1

→ **losange plein**



Les “plus” d'UML - Agrégation

Agrégation :

- Associations **non symétriques**
- Une classe joue un **rôle prépondérant** par rapport à l'autre

Deux formes d'agrégation :

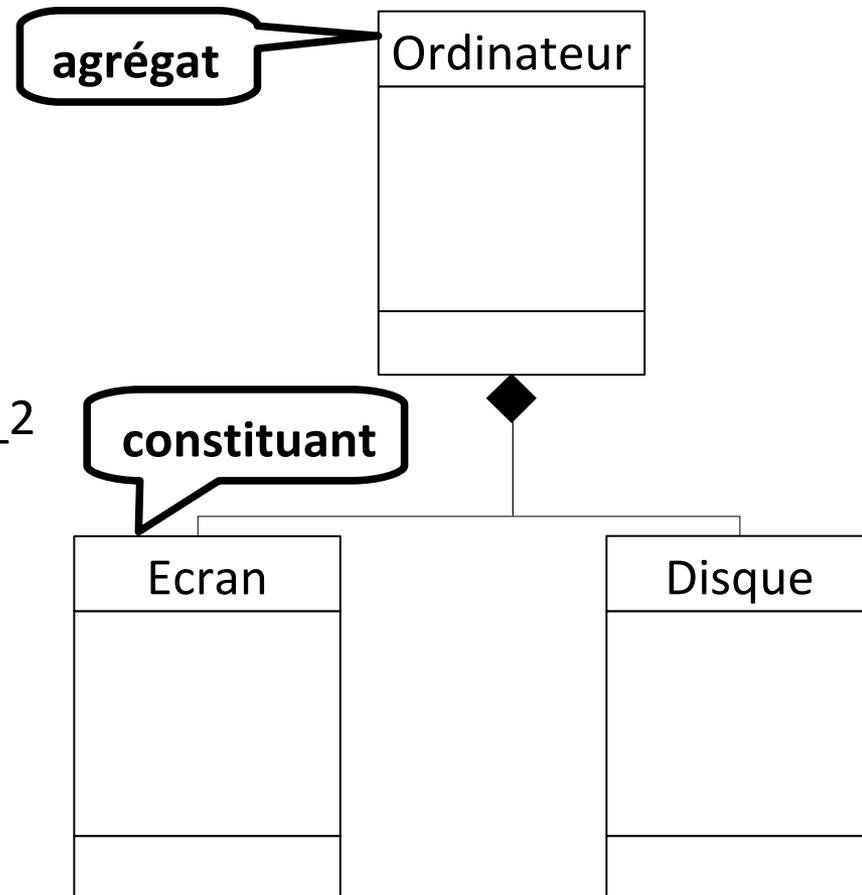
- **composition**
- agrégation partagée

Composition : une classe Classe_2 est sous-ensemble d'une autre, Classe_1

→ **losange plein**

Ex. :

Si un ordinateur est supprimé, son écran et son disque aussi



Les “plus” d'UML - Agrégation

Agrégation :

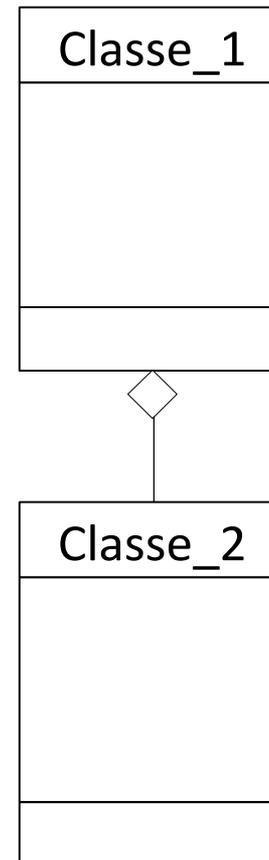
- Associations **non symétriques**
- Une classe joue un **rôle prépondérant** par rapport à l'autre

Deux formes d'agrégation :

- composition
- **agrégation partagée**

Agrégation partagée : une classe Classe_2 est dépendante d'une autre, Classe_1

→ **losange vide**



Les “plus” d'UML - Agrégation

Agrégation :

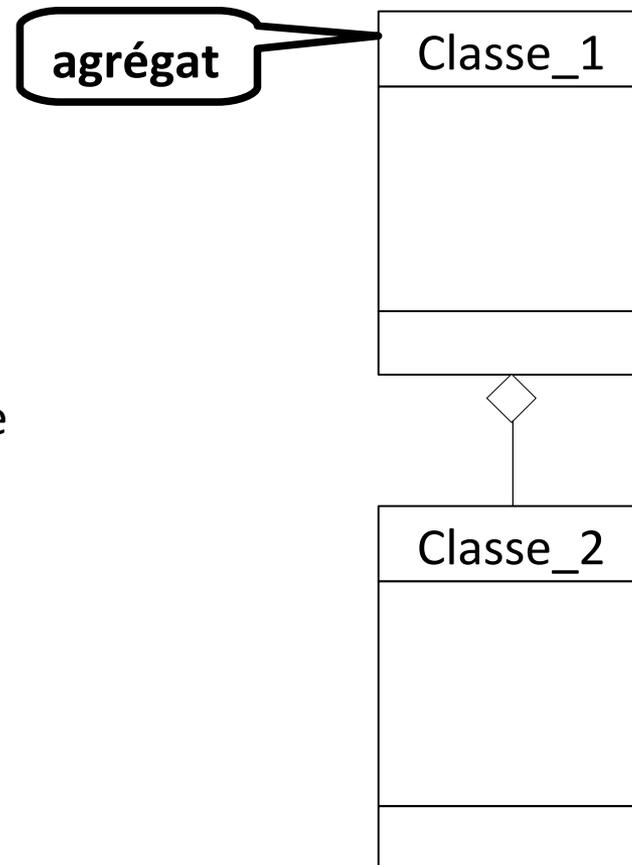
- Associations **non symétriques**
- Une classe joue un **rôle prépondérant** par rapport à l'autre

Deux formes d'agrégation :

- composition
- **agrégation partagée**

Agrégation partagée : une classe Classe_2 est dépendante d'une autre, Classe_1

→ **losange vide**



Les “plus” d'UML - Agrégation

Agrégation :

- Associations **non symétriques**
- Une classe joue un **rôle prépondérant** par rapport à l'autre

Deux formes d'agrégation :

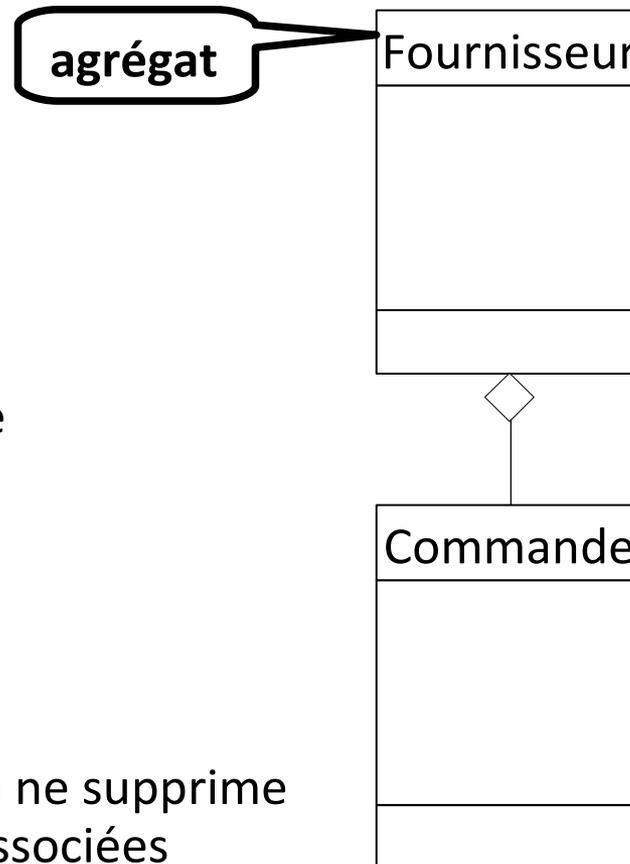
- composition
- **agrégation partagée**

Agrégation partagée : une classe Classe_2 est dépendante d'une autre, Classe_1

→ **losange vide**

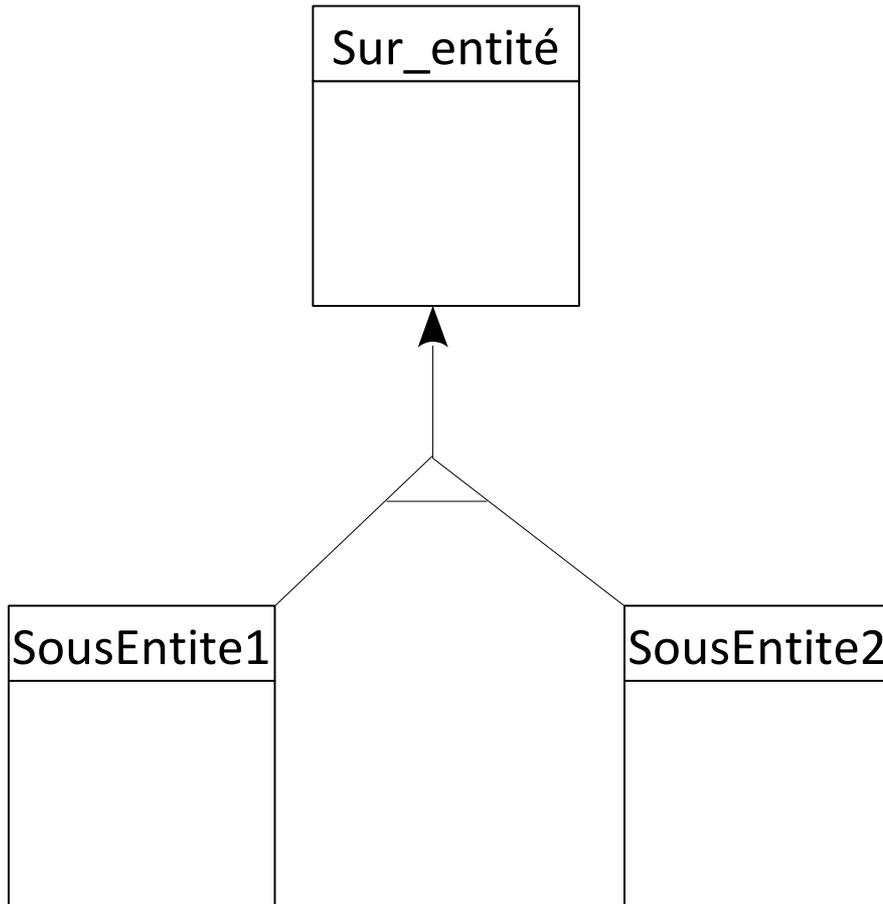
Ex :

Si on supprime le fournisseur, on ne supprime pas forcément les commandes associées

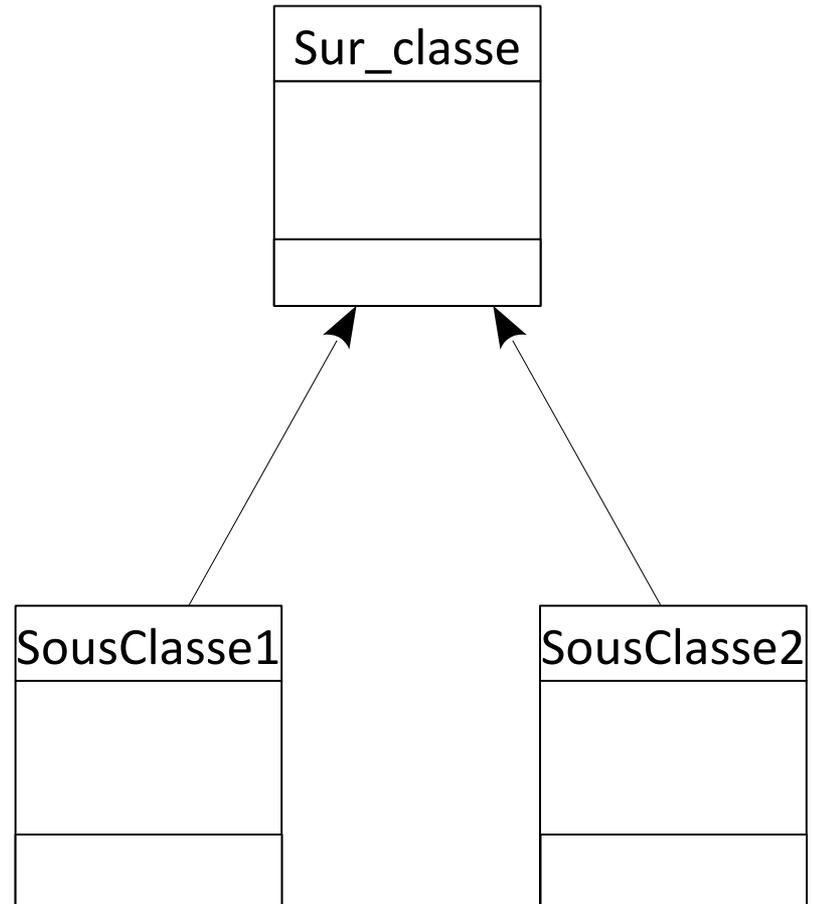


Modélisation MERISE et UML : héritage

MERISE

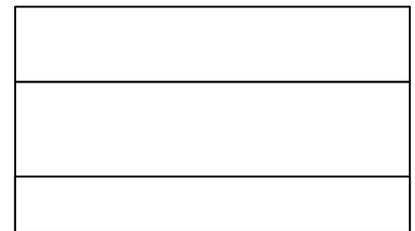
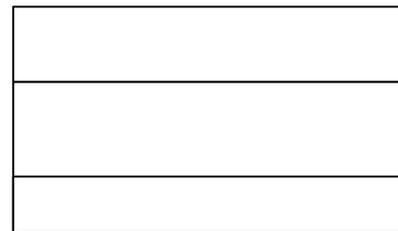
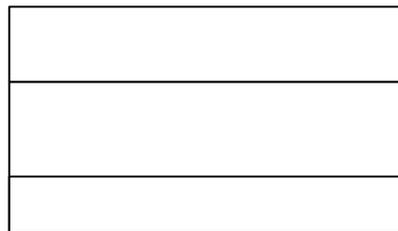
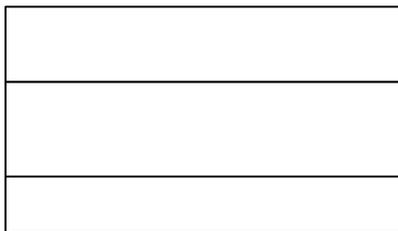
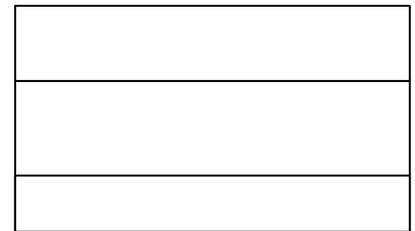
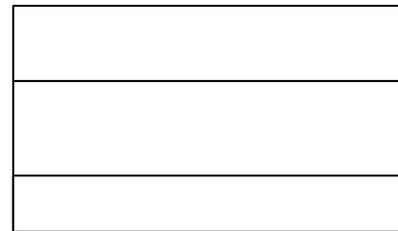
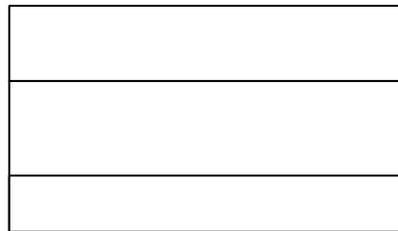
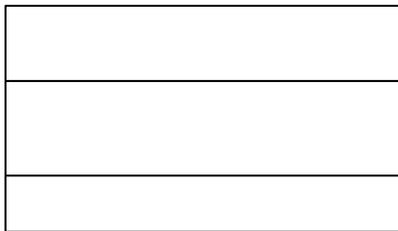
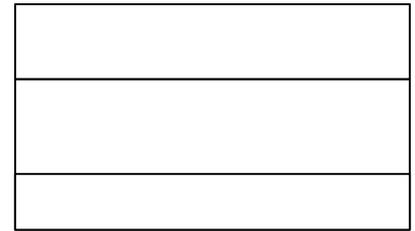
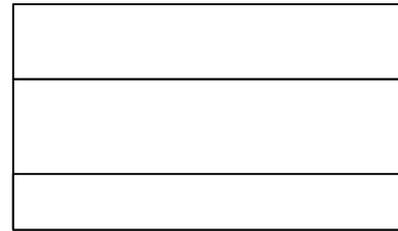
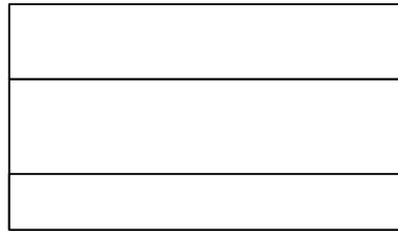
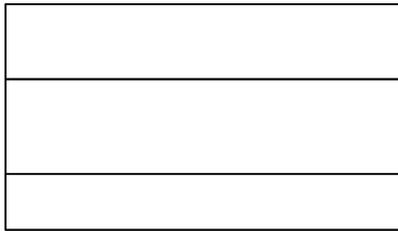


UML



Héritage : l'exemple des formes géométriques

Dessinez les **relations d'héritage** du modèle UML pour les classes d'objet suivantes :
Triangle, Rectangle, Cercle, Polygone, Carre, Triangle_isocele, Triangle_rectangle,
Triangle_equilateral, Hexagone, Quadrilatere, Rectangle, Parallelogramme, Losange.



Héritage : l'exemple des formes géométriques

Dessinez les **relations d'héritage** du modèle UML pour les classes d'objet suivantes : Triangle, Rectangle, Cercle, Polygone, Carre, Triangle_isocele, Triangle_rectangle, Triangle_equilateral, Hexagone, Quadrilatere, Rectangle, Parallelogramme, Losange.

Cercle

Triangle

Triangle_isocele

Triangle_equilateral

Polygone

Quadrilatere

Triangle_rectangle

Carre

Hexagone

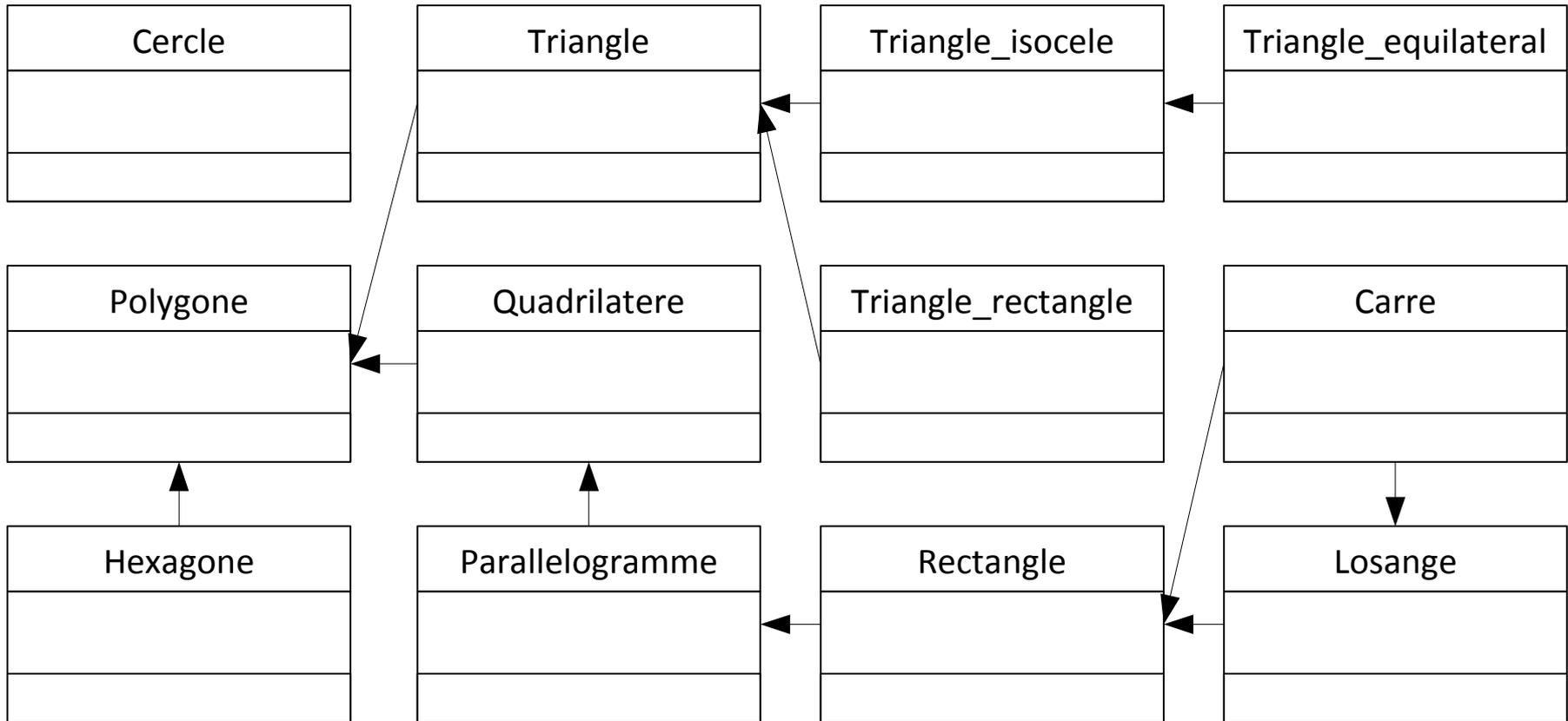
Parallelogramme

Rectangle

Losange

Héritage : l'exemple des formes géométriques

Dessinez les **relations d'héritage** du modèle UML pour les classes d'objet suivantes : Triangle, Rectangle, Cercle, Polygone, Carre, Triangle_isocele, Triangle_rectangle, Triangle_equilateral, Hexagone, Quadrilatere, Rectangle, Parallelogramme, Losange.



Héritage : l'exemple des formes géométriques

Dessinez les **relations d'héritage** du modèle UML pour les classes d'objet suivantes : Triangle, Rectangle, Cercle, Polygone, Carre, Triangle_isocele, Triangle_rectangle, Triangle_equilateral, Hexagone, Quadrilatere, Rectangle, Parallelogramme, Losange.

