

QCM 5 – UML et classes

1. Quelle est la différence entre un diagramme de classes et un diagramme d'objets ?

Le diagramme de classes représente l'aspect statique d'un système tandis que le diagramme d'objets montre son aspect dynamique. Ce dernier représente les instances créées et leurs liens lorsque le système est actif. Pour un diagramme de classe, il peut y avoir une infinité de diagrammes d'objets.

DIAGRAMME DE CLASSE

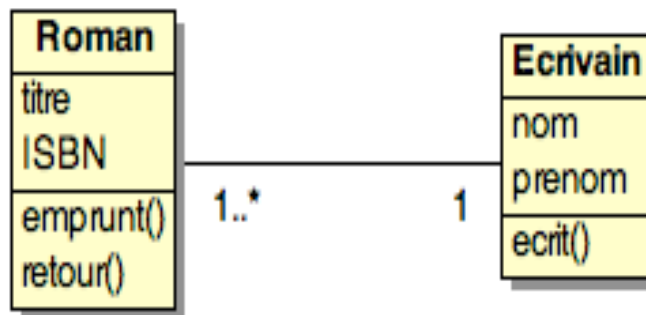
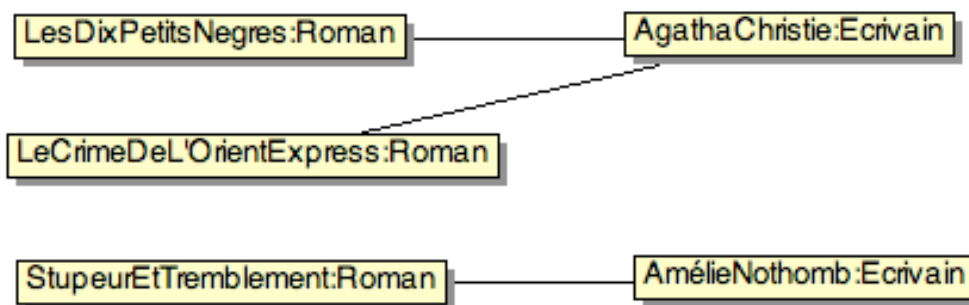


DIAGRAMME D'OBJETS



2. Une classe décrit uniquement les attribut d'un type d'objet

- a. Oui
- b. Non

Non, elle décrit aussi son comportement par les méthodes.

3. Qu'est-ce qu'une association entre deux classes ?

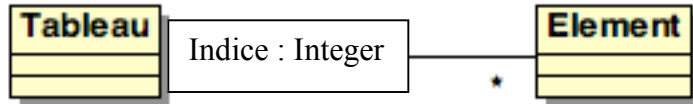
C'est un lien sémantique bidirectionnel, qui définit un lien entre les objets instances de chaque classe impliquée dans l'association.

4. Dans quel cas est-il possible d'utiliser un qualificateur dans une association et à quoi cela sert-il ?

Un qualificateur se place à l'extrémité d'une association de cardinalité non finie, pour identifier de façon unique l'instance participant à l'association (par exemple, le numéro de d'Elément, pour une association entre Tableau et Element). Les associations qualifiées spécifient un chemin précis pour trouver un objet cible (l'élément) à partir d'un objet source (le tableau).

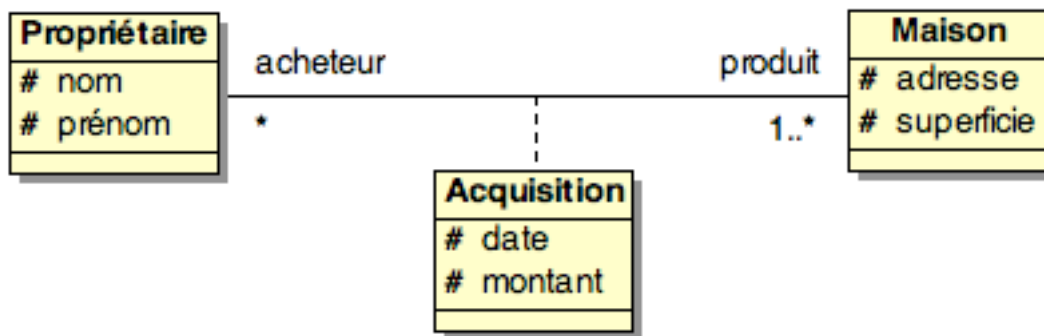


devient:



5. Quel est l'intérêt des classes d'association ?

Les classes d'association permettent de décrire les liens entre classes sous forme d'une classe. Elles sont dotées d'attributs et d'opérations spécifiques à chaque instance de lien.



6. Une classe d'association joue le rôle d'une classe classique et elle peut être liée à d'autres classes

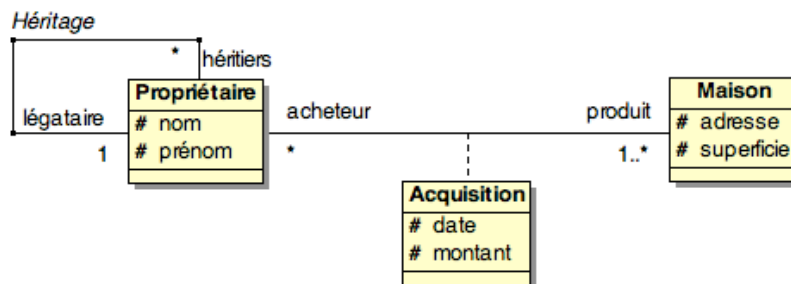
- Oui
- Non

Oui, une classe-association a le statut d'une classe décrivant les occurrences d'une association. Elle peut-être dotée d'attributs, d'opérations, et être reliée à d'autres classes par des associations

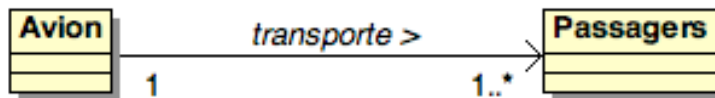
7. Il est possible de lier une classe à elle-même par association ?

- Oui
- Non

Oui, c'est une association réflexive



8. Les flèches dans une association représentent
- Des références
 - Des dépendences
 - La navigation
- c. Les flèches représentent le sens de la navigation de l'association



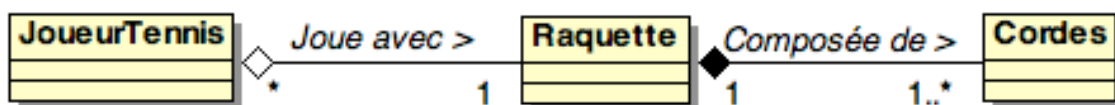
9. Quel est l'intérêt de la navigation ?
- C'est utile pour le passage au développement : on détermine ainsi quel objet possède la référence sur l'autre, et c'est moins complexe que de coder une association bidirectionnelle.

10. Quelle est l'utilité de nommer les extrémités d'une association ?
- Les noms spécifiés pour les extrémités d'une association représentent le rôle joué par les instances de classes dans l'association. Un rôle a la même nature qu'un attribut dont le type serait la classe située à l'autre extrémité de l'association.

11. L'aggrégation est un type d'association ?
- Oui
 - Non
- Oui, l'aggrégation est un cas particulier d'association associant un objet complexe aux objets qui le constituent.

12. Une composition est un type d'aggrégation ?
- Oui
 - Non
- Oui, avec des contraintes supplémentaires.

13. Quelles contraintes supplémentaires doivent vérifier une composition en comparaison avec une aggrégation ?
- Le composant n'appartient qu'à un seul objet composé (une roue de voiture n'est pas partagée entre plusieurs voiture)
 - La destruction de l'objet composé entraîne la destruction de ses composants (l'incendie de la voiture détruit aussi ses roues)



14. Quelle phrase détermine une relation d'héritage ?
- Un cheval est un type d'animal
 - Un animal est un type de cheval
- Un cheval est un type d'animal.

15. Un attribut protégé est visible uniquement dans sa classe ?
- Oui
 - Non
- Non, il est aussi visible de ses sous-classes (par héritage).

16. Mapper l'adjectif avec le niveau de visibilité

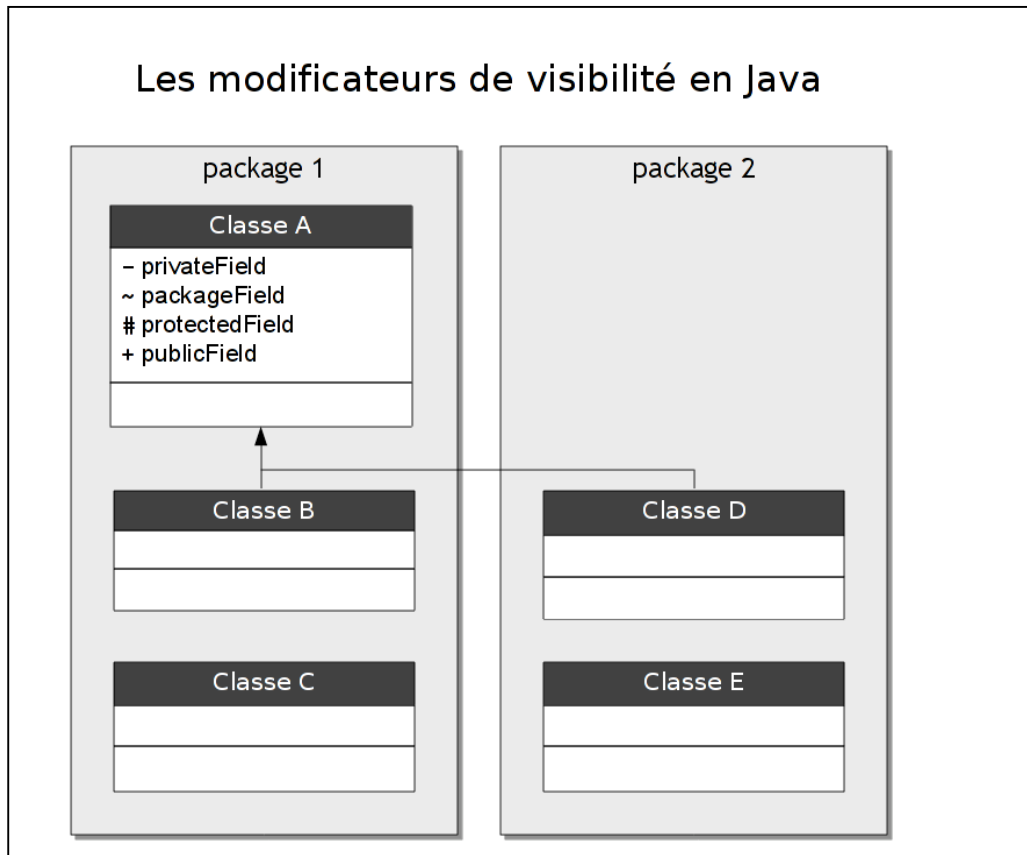
- | | |
|--------------|------------------------|
| 1. Private | a. Paquetage |
| 2. Protected | b. Classe |
| 3. Public | c. Classe et héritiers |

1.b

2.c

3 : tout

Que se passe-t-il si rien n'est spécifié ? C'est une visibilité de niveau Paquetage par défaut.



Source: <http://thecodersbreakfast.net/index.php?post/2009/06/06/Java-%3A-les-modificateurs-de-visibilit%C3%A9>

	Classe A	Classe B	Classe C	Classe D	Classe E
privateField	✓				
packageField	✓	✓	✓		
protectedField	✓	✓	✓	✓	
publicField	✓	✓	✓	✓	✓

<http://thecodersbreakfast.net>

Public ou + :

tout élément qui peut voir le conteneur peut également voir l'élément indiqué.

Protected ou # :

seul un élément situé dans le conteneur ou un de ses descendants peut voir l'élément indiqué.

Private ou - :

seul un élément situé dans le conteneur peut voir l'élément.

Package ou ~ ou rien :

seul un élément déclaré dans le même paquetage peut voir l'élément.

17. Quelle est la différence entre une classe concrète et une classe abstraite ?

Une classe concrète possède des instances, alors qu'une classe abstraite n'en possède pas.

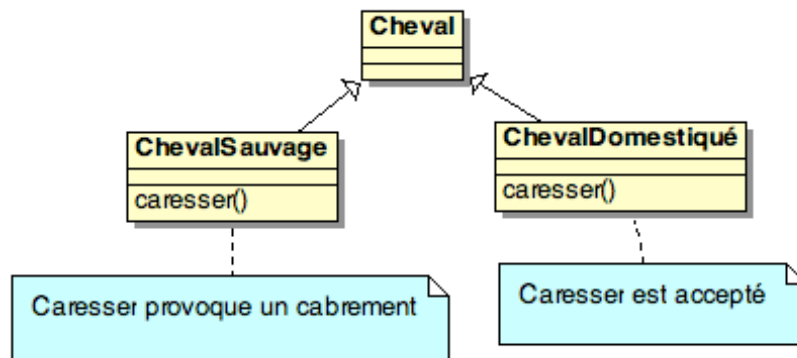
18. A quoi sert de définir des classes abstraites dans un modèle ?

Elle a pour objet de définir un comportement commun à un sous-ensemble de classes (ses héritières), et de factoriser les méthodes et les attributs communs à ses sous-classes.

19. Qu'est-ce que le polymorphisme

Le polymorphisme signifie que la même méthode peut avoir des comportements très différents dans des classes différents. Lors de l'appel d'une méthode de même nom, cette différence se traduit par des comportements différents sauf dans le cas où la méthode est commune et héritée de la surclasse dans les sous-classes).

Exemple : si l'on considère la classe Cheval dans son intégralité, on a donc un ensemble de chevaux qui ne réagissent pas de la même façon lors de l'activation de la méthode caresser.



20. Qu'est-ce que les stéréotypes UML ?

Les stéréotypes permettent de définir de nouveaux éléments de modélisation par l'extension d'éléments existant en UML. Certains sont prédéfinis en UML, et les autres peuvent être définis par l'utilisateur (par exemple <<table>> pour indiquer que l'objet décrit une table dans une BD).

21. Quel stéréotype permet de définir une classe abstraite ?

<<abstract>> définit une classe abstraite. La notation du nom de la classe en italique suffit aussi.

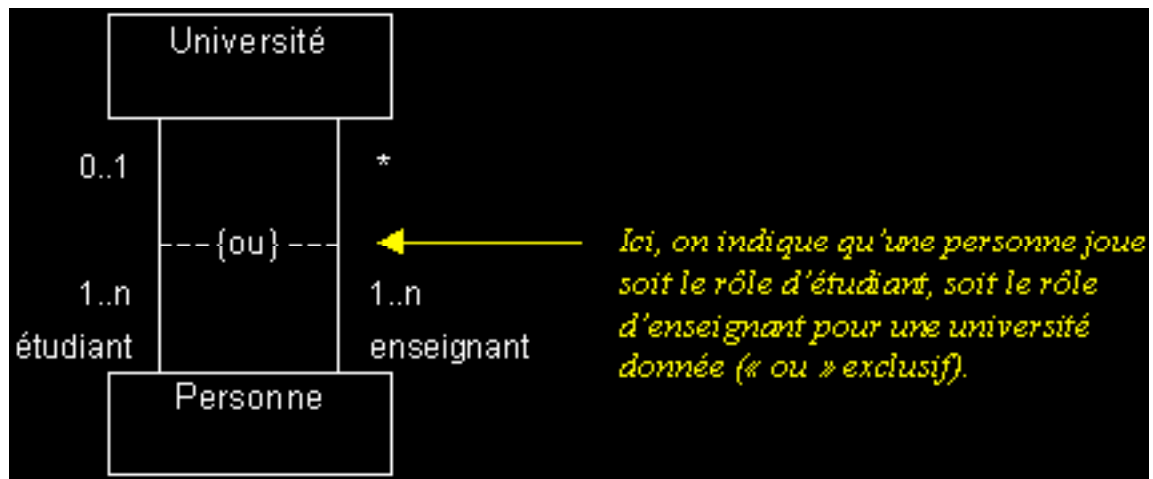
22. Les contraintes sont représentées dans un diagramme de classe par :

- Le nom de la contrainte entre parenthèse (*nom de la contrainte*)
- Le nom de la contrainte entre accolades {*nom de la contrainte*}
- Le nom de la contrainte précédé par le mot clé CONSTRAINT :
CONSTRAINT <nom de la contrainte>

Entre accolades (b)

23. Quelle contrainte permet d'exprimer l'exclusion entre les ensembles d'occurrences de 2 associations

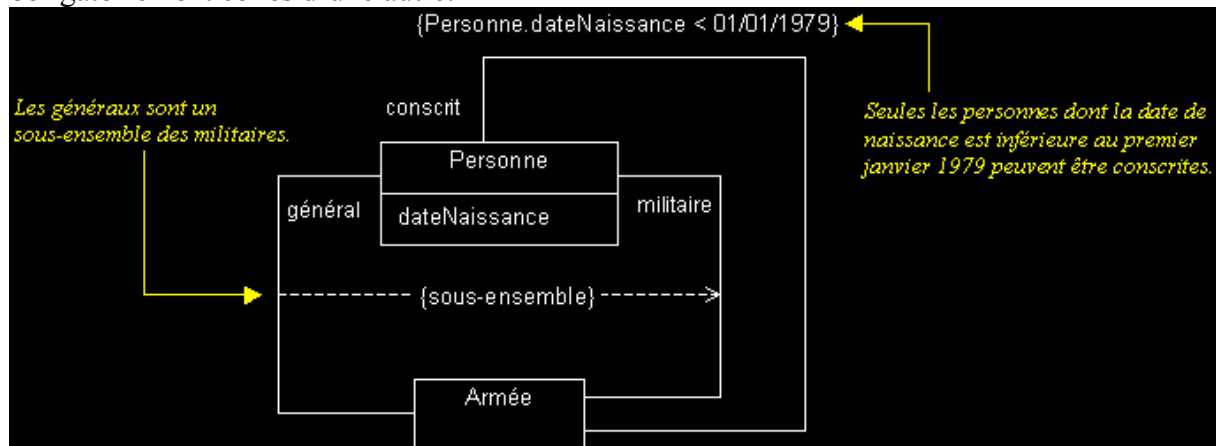
La contrainte {xor}. Elle exprime le fait qu'un objet ne peut être lié que par l'une ou l'autre des associations référencées par la contrainte.



Source : <http://uml.free.fr/cours/p16.html>

24. La notion d'inclusion entre les occurrences de deux associations peut-elle être exprimée à l'aide d'une contrainte

L'inclusion se note {subset}. Elle exprime le fait que les occurrences d'une association sont obligatoirement celles d'une autre.

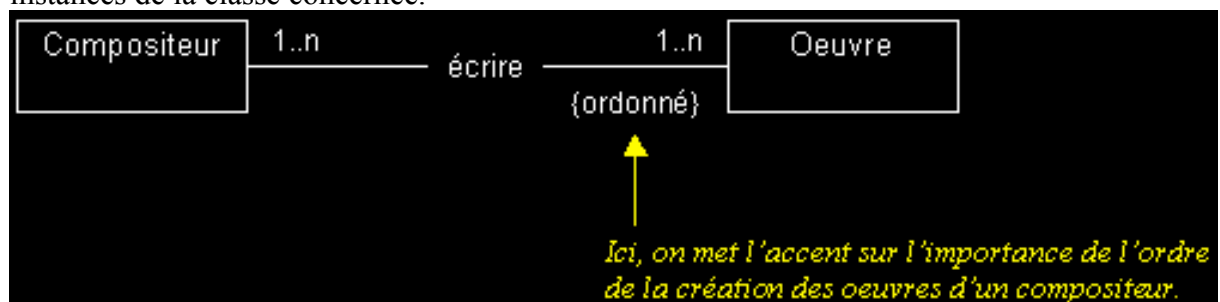


Source : <http://uml.free.fr/cours/p16.html>

25. Parmi les contraintes suivantes, laquelle est utilisée pour exprimer un ordre dans une association ?

- a. Arranged
- b. Ordered
- c. Numbered

B, {ordered}, est utilisé à l'extrémité d'une association pour exprimer un ordre sur les instances de la classe concernée.



Source : <http://uml.free.fr/cours/p16.html>