

Génie logiciel

Notes du cours de 02/12

L3 Informatique appliquée 2022-2023

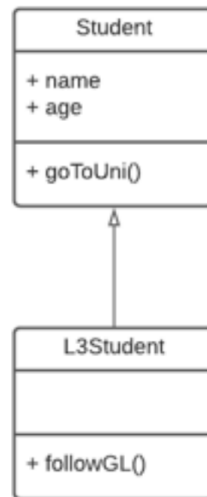
MABROUK Fayez

2 décembre 2022

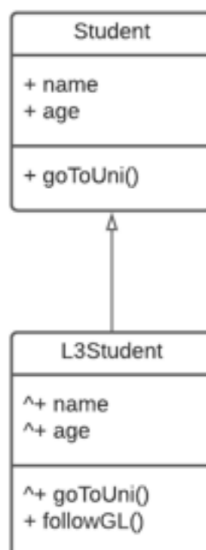
1 UML pour modéliser la structure

1.1 Héritage

- * Les instances d'une sous-classe sont également des instances de la superclasse.
- * Par conséquent, elles héritent des méthodes définies dans la superclasse.
- * Exemple :

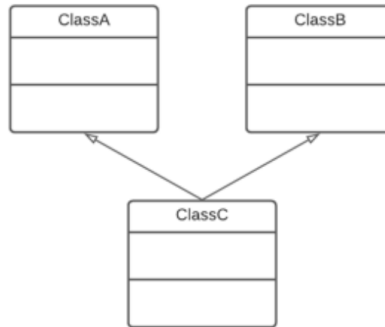


- * Notez que vous pouvez explicitement montrer des éléments hérités en les préfixant avec "chapeau".
- * Enfin, notez que les associations entre une classe et une superclasse est héritée par ses sous-classes.



1.2 Héritage multiple

- * Il est possible qu'une classe soit une spécialisation de plus d'une classe.
- * Exemple : La classe C est une spécialisation de la classe A et de la classe B.

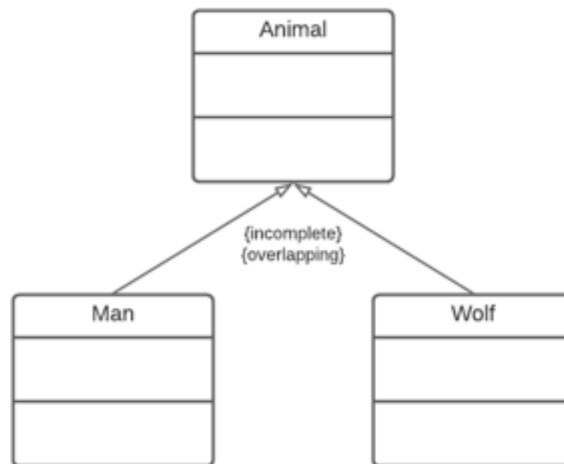


- * L'héritage multiple peut être problématique si un attribut de même nom/type ou un nom/-type ou un code mathématique avec la même signature est défini dans plus d'une superclasse.
- * Pas toujours possible en pratique : pas d'héritage multiple en Java.

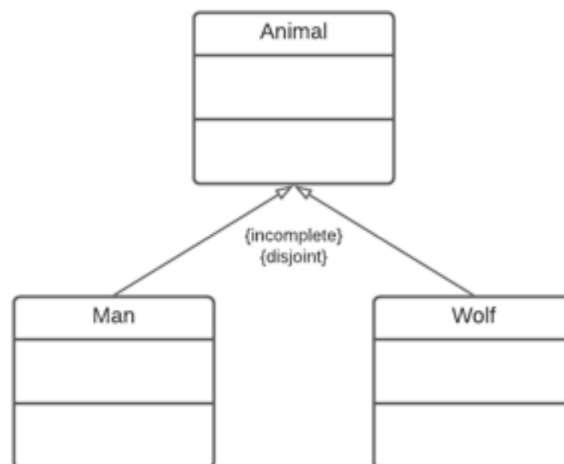
1.3 Contraintes

- * Il est possible d'ajouter des contraintes sur la relation, soit sur :
 - * la complétude : la spécialisation peut être complète ou incomplète. Si elle est complète, elle indique que l'ensemble des domaines des sous-classes couvre le domaine de la super-classe.
 - * superposition : la spécialisation peut être soit disjointe (elle n'a pas d'instances communes), soit superposée. n'ont pas d'instances communes) ou superposées (elles peuvent avoir des instances communes)
- * Syntaxe : contrainte.

- * Exemple :



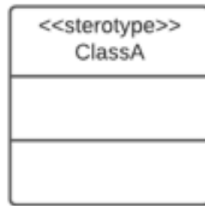
- * il y a d'autres animaux que les hommes et les loups, donc la relation est incomplète
- * si vous croyez aux loups-garous, une instance peut être à la fois un homme et un loup, donc la relation se chevauche.



- * Probablement, un homme ne peut pas être un loup.

1.4 Stéréotypes

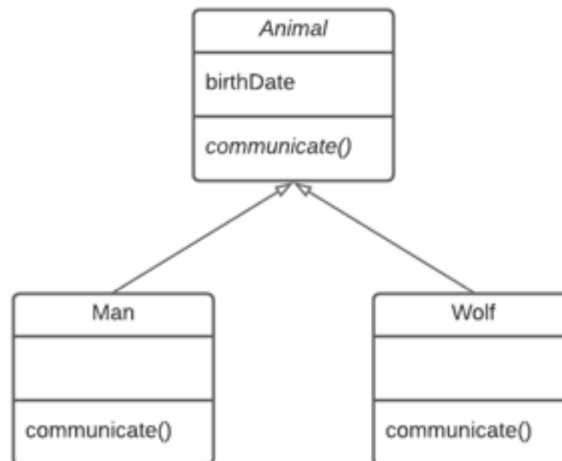
- * Les stéréotypes peuvent être utilisés pour spécialiser un élément en UML.
- * Syntaxe : «stéréotype» au-dessus du nom de la classe.



- * Stéréotypes possibles :
 - * énumération : classe présentant un type avec une liste de valeurs constantes.
 - * auxiliaire : pour indiquer une classe secondaire.
 - * abstrait.
 - * interface.

1.5 Abstract classes

- * Rappel : Abstrait et concret classes abstraites : les classes abstraites sont des classes qui n'ont pas d'instances (par exemple Mammal). Les classes concrètes en ont (par exemple Human).
- * Les classes abstraites permettent de hiérarchiser les classes de classes et de regrouper les attributs et les méthodes. Elles doivent avoir des sous-classes.
- * Exemple : la méthode communicate de classe Animal est abstraite (indiquée en italique). Elle n'est pas définie pour un animal, mais elle l'est pour les classes concrètes.
- * Note : peut aussi être indiquée en italique le nom de la classe.



1.6 Interface

- * Définition : une interface est une classe totalement abstraite : elle ne possède aucun attribut et ses méthodes sont toutes publiques. n'a aucun attribut et ses méthodes sont toutes publiques et abstraites. et abstraites.

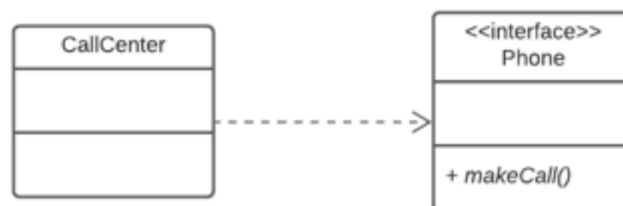
- * Syntaxe : stéréotype + flèche vide en pointillés.



- * Alternative : sucette.



- * Lorsqu'il y a une dépendance sur une interface, on peut être notée "classiquement".

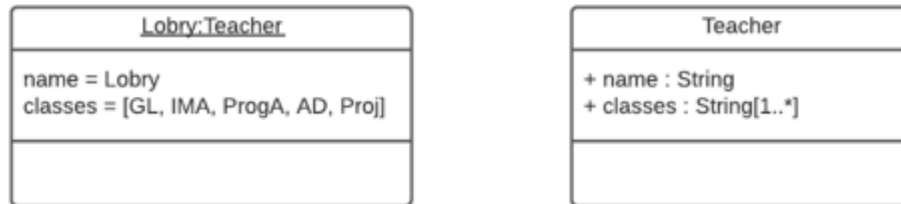


- * Ou à travers une sucette(a lollipop).



1.7 Représentation d'un objet

- * Le diagramme de classes représente une vue statique de la structure.
- * Le diagramme d'objets peut montrer un instantané du système :
- * Le diagramme d'objets montre les instances et les valeurs de leurs attributs.
- * Syntaxe : nom de l'instance :nom de la classe.

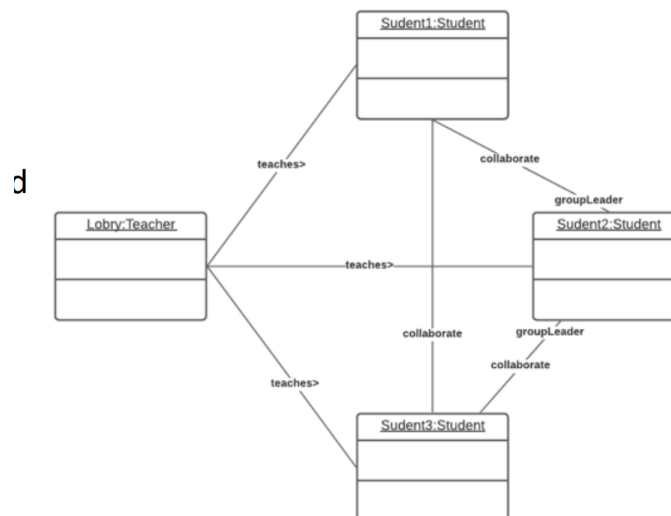


- * Lien "instanceOf" facultatif.



1.8 Relation entre les instances

- * Enfin, il est possible de représenter les interactions entre les instances avec une ligne pleine.
- * Facultatif : nom de la relation et rôles.



1.9 Conclusion

- * Les diagrammes de classes permettent d'ajouter des informations sur la structure de notre modèle.
- * L'ajout des bons liens entre les classes améliore la sémantique et rend le diagramme plus léger.
- * Comme toujours avec la modélisation :
 - * Faites attention à la cible du modèle : qu'ont-ils besoin de savoir ?
 - * Pas seulement un diagramme, il doit être accompagné d'une documentation (en particulier : vos choix)
 - * Pas une bonne solution unique.
- * Nécessite de la pratique.