

BUT Informatique
1A - Semestre 1
Introduction aux bases de données
(R1.05)

R. Fleurquin

Chapitre 3

Diagramme de classes et d'objets UML : objets et classes

Les objets du monde réel

- Dans le monde réel, notre esprit structure notre environnement selon le concept d'objet.
- Certains de ces objets sont statiques. Ils ne font rien par eux mêmes (chaise, table, etc.)
- D'autres sont animés. Ils répondent à des sollicitations (la télévision) et sont même capables d'agir par eux mêmes (votre voisin(e) d'amphi).

L'état d'un objet

- Tous ces objets ont des **caractéristiques** qui leur sont propres : taille, poids, couleur, forme, nom, etc.
- Certaines de ces caractéristiques peuvent éventuellement évoluer dans le temps : l'âge et le poids d'une personne par exemple ☹.
- A un instant donné les valeurs de ces caractéristiques reflètent l'**état** d'un objet au moment où on le considère.

Le comportement d'un objet

- Certains de ces objets offrent aux autres objets des **services** : un téléviseur vous permet par exemple de capter et regarder la télévision, baisser le son ou de changer de chaîne.
- Les objets interagissent entre eux en sollicitant les services des uns et des autres.
- De cette collaboration naît tout ce qui se passe dans notre bel univers!

L'identité d'un objet

- Chaque objet du mode réel est unique dans l'espace temps. Même deux chaises identiques placées l'une à côté de l'autre nous apparaissent comme unique chacune.
- Notre esprit associe donc à chaque objet une identité propre et unique. Il peut ainsi les percevoir comme indépendant.

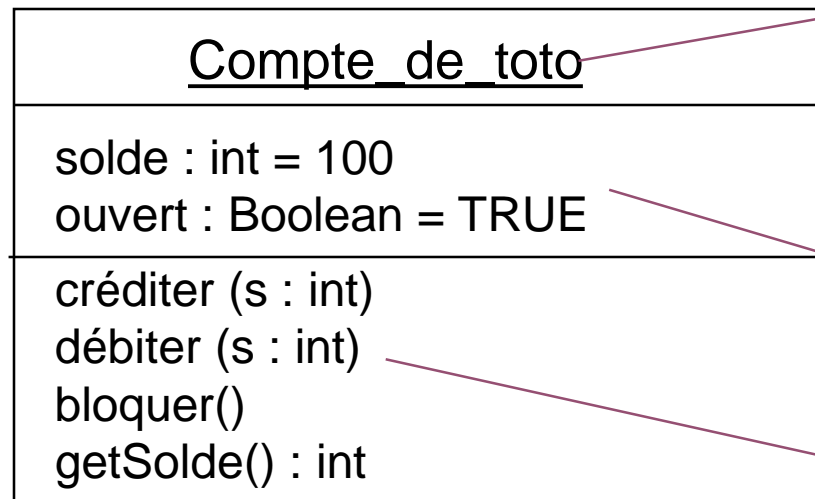
Les « classes » d'objets

- Certains objets du monde réel partagent des propriétés communes. Par exemple tous les téléviseurs possèdent un écran, captent et retransmettent des images et des sons, etc.
- Ces propriétés communes peuvent être décrites dans une représentation abstraite : le concept de téléviseur.
- Ce concept aide notre esprit à cataloguer des objets qui se ressemblent sur le plan de leur état et de leur comportement.
- C'est une façon pour notre esprit de gérer la complexité du monde qui nous entoure en raisonnant par « paquets » d'objets se ressemblant.

Les objets UML ressemblent aux objets du monde réel!

- Comme pour les objets du monde réel, les objets UML se voient associés 3 traits :
 - un état
 - une identité (unique pour chaque objet dessiné)
 - un comportement

Dessin d'un objet UML à un instant de sa vie



Nom de l'objet. Toujours souligné et toujours unique dans un diagramme = son identifiant dans le diagramme!

État = une liste d'attributs valués. L'état est toujours dessiné à un instant particulier.

Comportement = une liste de méthodes (les services offerts par cet objet aux autres objets par un mécanisme appelé « envoi de message » entre objets)

Des représentations UML des objets dont on peut varier le niveau de détail

Compte_de_toto

(Un objet dont on ne donne que le nom)

Compte_de_toto

solde = 100
état = TRUE

(L'objet est décrit par son nom et son état complet ou non à l'instant où on le considère)

Compte_de_toto

créditer ()
débitier ()
bloquer()

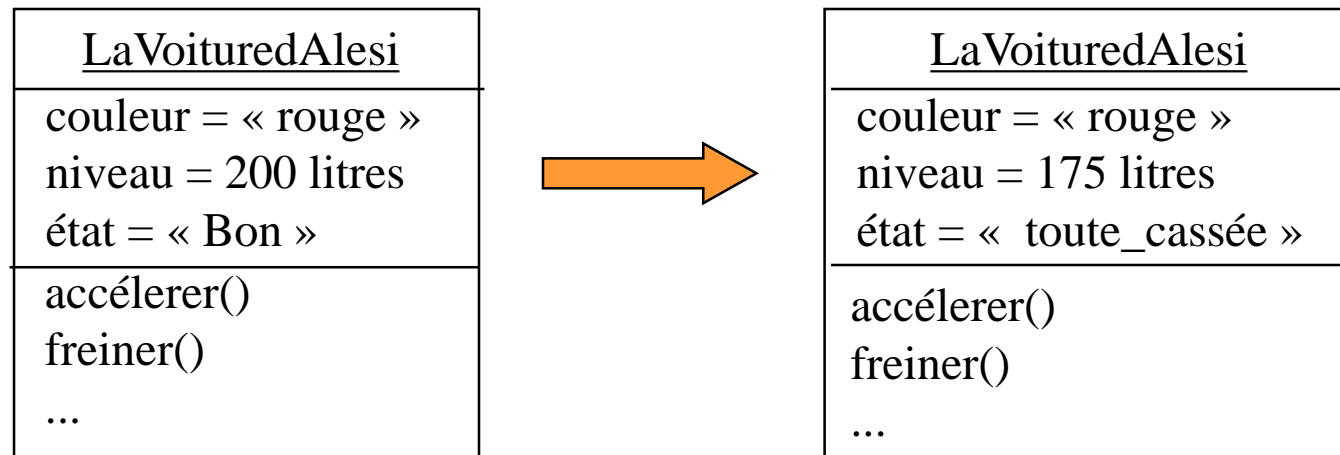
(On ne donne ici que le nom de l'objet et son comportement = les services qu'il offre aux autres objets)

L'État d'un objet UML

- L'état d'un objet regroupe les valeurs à un instant particulier de tous ses attributs (et les liens qu'il entretient avec d'autres objets cf. cours 3).
- Un attribut est une information atomique particulière sur l'objet du monde réel modélisé (une variable au sens informatique portée par l'objet).
- Chaque attribut peut prendre une valeur dans un domaine de définition donné et fixé à vie.
- Ce domaine est un type au sens informatique (type entier, type booléen, etc.).

L'état d'un objet UML peut évoluer au cours du temps (de sa naissance, jusqu'à sa mort).

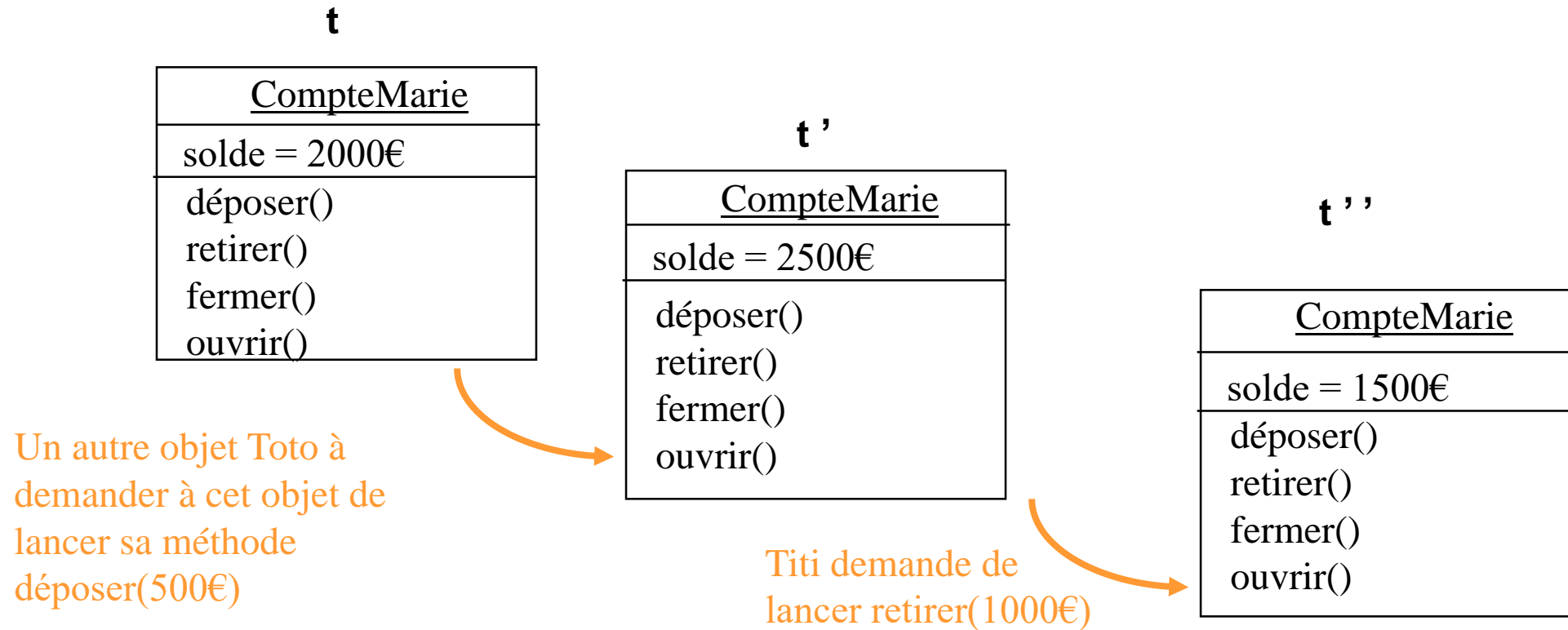
Un même objet dessiné à deux instants différents peut afficher deux états différents



Dessin à l'instant t

Dessin à l'instant t'

L'état d'un objet a un instant donné est la conséquence de ses comportements passés => donc des sollicitations d'autres objets qui lui ont réclamé l'exécution de telle ou telle méthode.

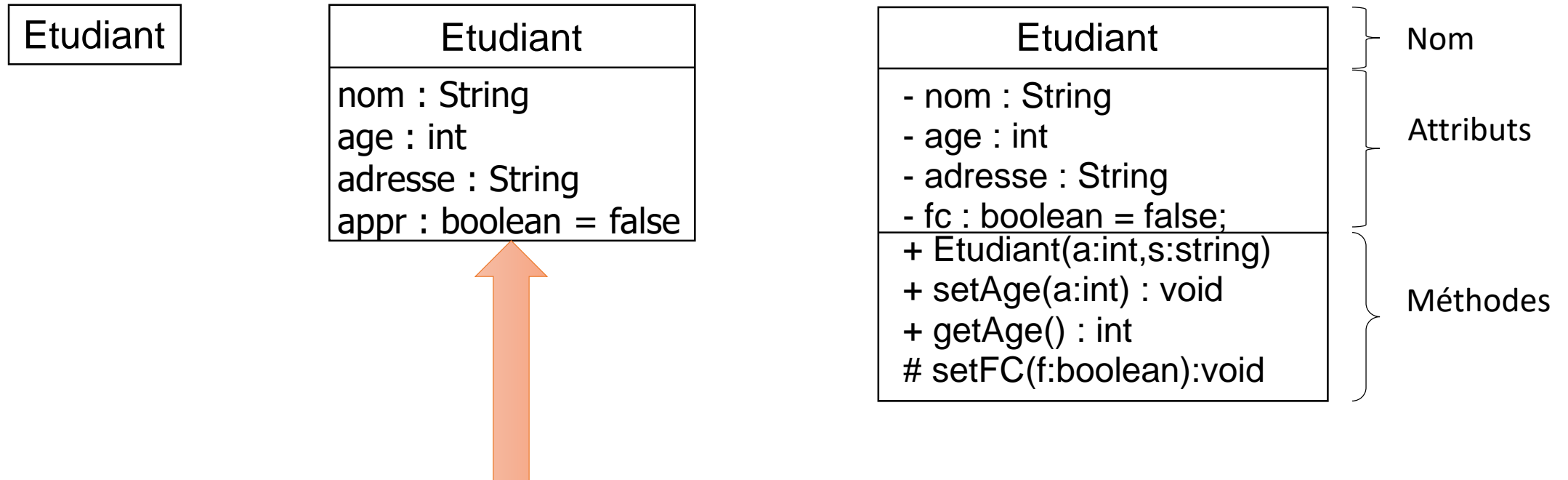


La classe

- Dans le domaine ciblé par l'application que l'on veut développer certains types « d'objets » apparaissent en plusieurs exemplaire : des centaines d'étudiants, des milliers de notes, etc.
- Une **classe UML** représente une abstraction (ne retenant que les informations utiles pour le logiciel à développer) d'un de ces types d'objet utile à l'application.
- La classe UML est la représentation explicite en UML de la notion de *représentation abstraite* du monde réel.
 - Une classe UML sera souvent implémenter plus tard dans un code prenant la forme par exemple d'une classe JAVA ou d'une table relationnelle.
- Une classe UML est un « moule » à partir duquel il est possible de créer des **objets** UML qui présenteront les mêmes caractéristiques (attributs) et offriront les mêmes services (méthodes).
- On évite ainsi de décrire plusieurs fois des objets qui se ressemblent. On les “factorise” dans le concept de classe.
- Le seul processus permettant d'obtenir un objet (depuis une classe) se nomme **instanciation**.
- Tout objet est nécessairement instance d'une classe et d'une seule.
 - On dit en désignant un objet de nom X : « l'objet X » ou encore « l'instance X de la classe C ».

Représentation d'une classe en UML

(On peut varier à loisir le niveau de détail selon l'étape de développement!)

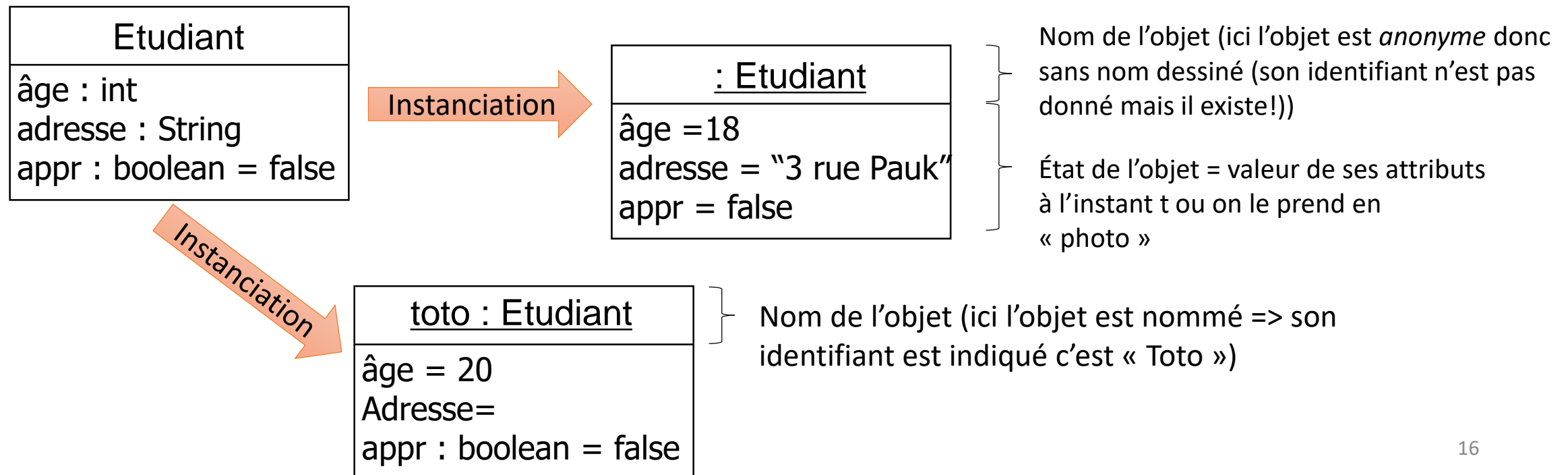


Niveau de détail utilisé dans R1.05

On oublie le comportement (les méthodes) car on ne s'intéresse qu'aux données...

A partir d'une classe on peut créer des objets par instantiation

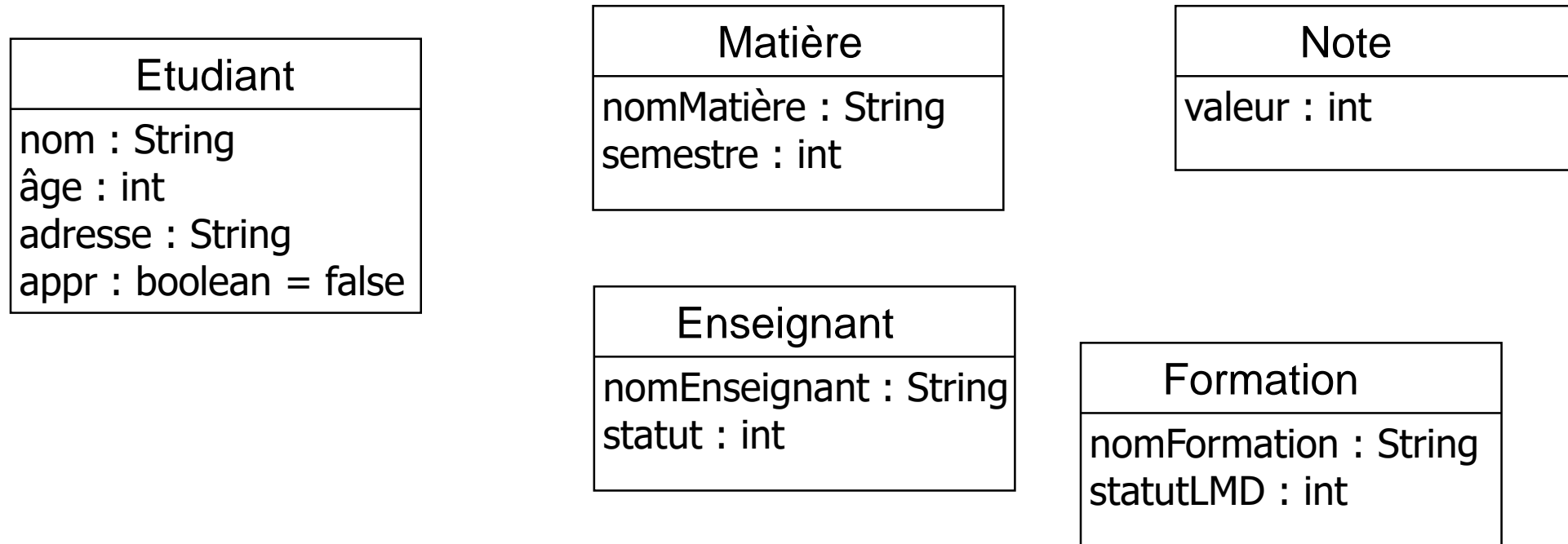
- Toutes les instances d'une même classe partagent des propriétés communes (les mêmes attributs et offrent les mêmes méthodes)
- Le dessin d'un objet représente cet objet à un instant t particulier.



Comment utiliser les classes pour indiquer les données avec lesquelles va travailler un logiciel? Un exemple!

- On souhaite développer une application de gestion de notes à l'université.
- Dans le domaine de ce logiciel, on va bien sûr traiter des étudiants, des matières, des formations, les notes des étudiants, etc.
- Pour cataloguer l'information que l'on aura à utiliser (et donc la conserver dans une base de données) dans ce logiciel on va utiliser des classes UML.
 - Par exemple on va dessiner une classe UML Étudiant avec les attributs : nom, age, adresse, etc. Cette classe fait état de la nécessité de pouvoir manipuler des informations de cette nature pour plusieurs étudiants dans le logiciel.

On produit un diagramme de classes qui indique les informations que l'on devra manipuler. En R1.05 les classes ne comportent jamais de méthodes...



Ce diagramme donne déjà de nombreuses indications sur les données que l'on souhaite manipuler, mais il est encore très incomplet : de nombreuses informations sont manquantes. Exemple une note est associée à quelle matière et à quel étudiant? => vite RDV au cours 3 sur les associations binaires UML!!