

Exercices UML

Classification vs généralisation

Considérer les phrases suivantes avec "est un".

- A. Snoopy est un beagle.
- B. Un beagle est un chien.
- C. Les chiens sont des animaux.
- D. Un beagle est une race.
- E. Chien est une espèce.

Que peut-on déduire si l'on combine les phrases

- A et B
- B et C
- A et B et C
- A et D
- B et E

Pourquoi?

Classification vs généralisation

Considérer les phrases suivantes avec "est un".

- A. Snoopy est un beagle.
- B. Un beagle est un chien.
- C. Les chiens sont des animaux.
- D. Un beagle est une race.
- E. Chien est une espèce.

Que peut-on déduire si l'on combine les phrases

- A et B \rightarrow Snoopy est un chien.
- B et C \rightarrow Un beagle est un animal.
- A et B et C \rightarrow Snoopy est un animal.
- A et D \rightarrow ~~Snoopy est une race.~~
- B et E \rightarrow ~~Un beagle est une race.~~

Pourquoi?

Classification vs généralisation

Considérer les phrases suivantes avec "est un".

- | | |
|---------------------------------|----------------|
| A. Snoopy est un beagle. | classification |
| B. Un beagle est un chien. | généralisation |
| C. Les chiens sont des animaux. | généralisation |
| D. Un beagle est une race. | classification |
| E. Chien est une espèce. | classification |

Que peut-on déduire si l'on combine les phrases

- A et B → Snoopy est un chien.
- B et C → Un beagle est un animal.
- A et B et C → Snoopy est un animal.
- A et D → ~~Snoopy est une race.~~
- B et E → ~~Un beagle est une race.~~

Pourquoi?

Classification: relation instance — classe

Généralisation: relation classe — superclasse

Les généralisations sont transitives, les classifications ne le sont pas.

On peut combiner une classification en la suivant avec une généralisation, mais pas l'inverse.

Un bon signe pour détecter une généralisation est qu'il est possible de reformuler avec "tous les x sont des y". Par exemple on peut dire "tous les beagles sont des animaux", mais pas "tous les chiens sont des espèces".

Cardinalités

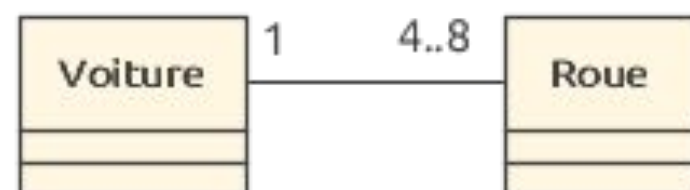
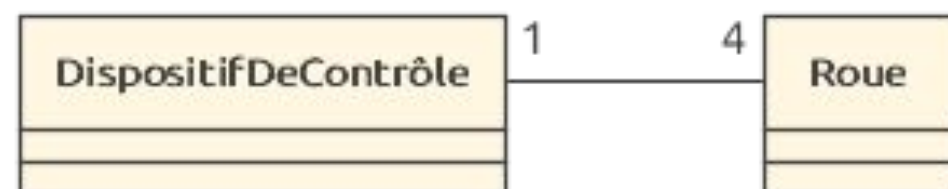
Modéliser les cas suivants (indépendants les uns des autres) en focalisant sur la cardinalité des roues.

- Un constructeur de voitures produit des voitures. Une application gère les pièces, produites à l'usine ou achetés chez un fournisseur. Certaines voitures sont vendues avec une roue de secours, d'autres pas.
- Le système antidérapage d'une voiture a un dispositif de contrôle qui est connecté aux roues.
- Un garage achète des voitures d'occasion aux particuliers. Il reprend aussi les roues d'hiver (ou été) si elles sont en bon état. Une application gère les achats.

Cardinalités

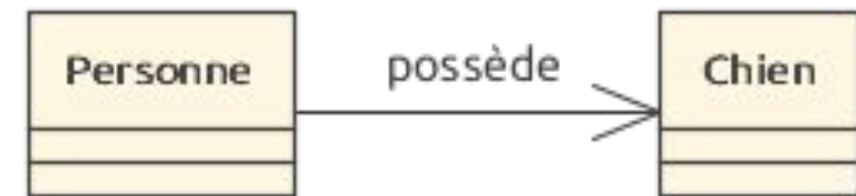
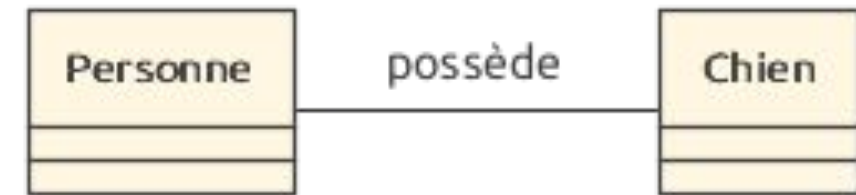
Modéliser les cas suivants (indépendants les uns des autres) en focalisant sur la cardinalité des roues.

- Un constructeur de voitures produit des voitures. Une application gère les pièces, produites à l'usine ou achetées chez un fournisseur. Certaines voitures sont vendues avec une roue de secours, d'autres pas.
- Le système antidérapage d'une voiture a un dispositif de contrôle qui est connecté aux roues.
- Un garage achète des voitures d'occasion aux particuliers. Il reprend aussi les roues d'hiver (ou été) si elles sont en bon état. Une application gère les achats.



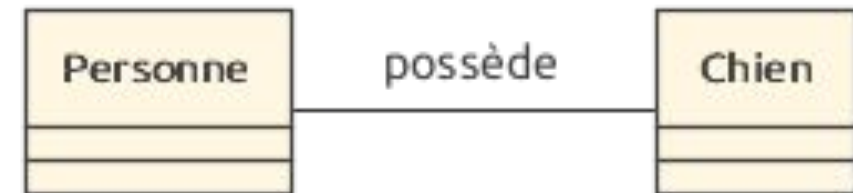
Flêches

- Vous créez le modèle suivant.
- Vous voulez faire la relation entre les chiens et leurs propriétaires explicite : qui possède qui. Est-ce que le diagramme suivant est correct?



Flèches

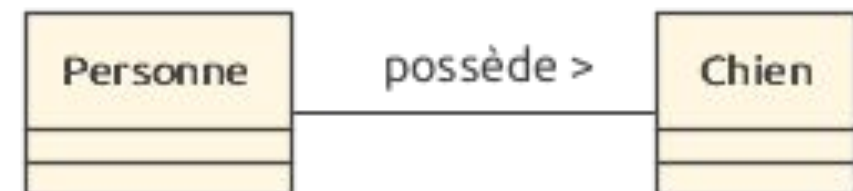
- Vous créez le modèle suivant.



- Vous voulez faire la relation entre les chiens et leurs propriétaires explicite. Est-ce que le diagramme suivant est correct?

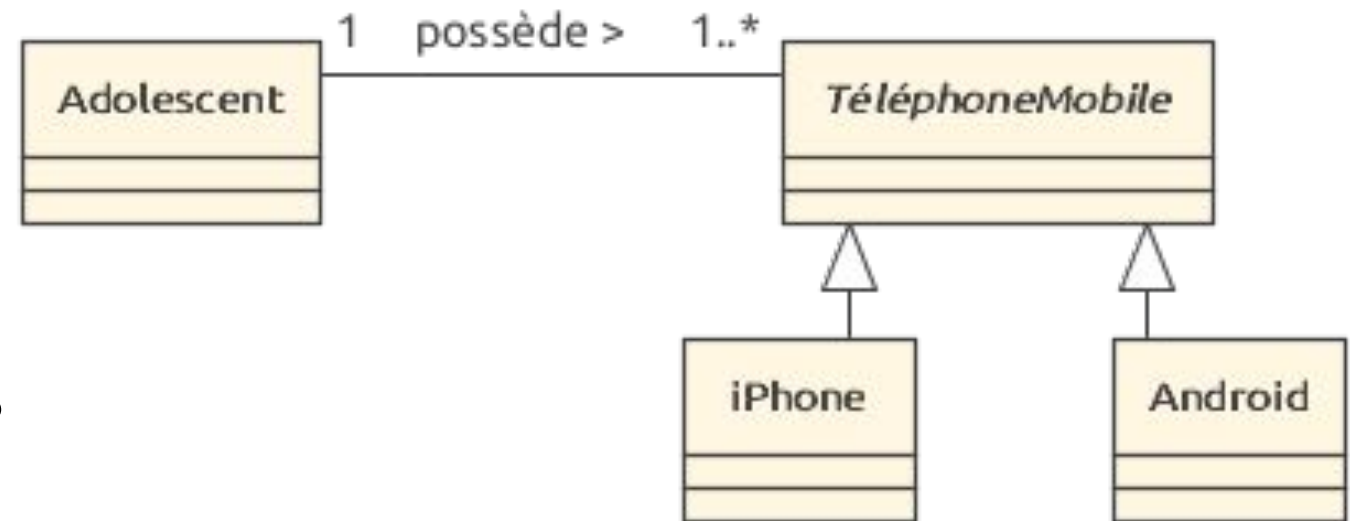


- Non, le diagramme proposé dit que l'on peut **naviguer** depuis un objet Personne à son objet Chien. Ce que vous vouliez faire était de préciser le **sens de lecture** de l'association.



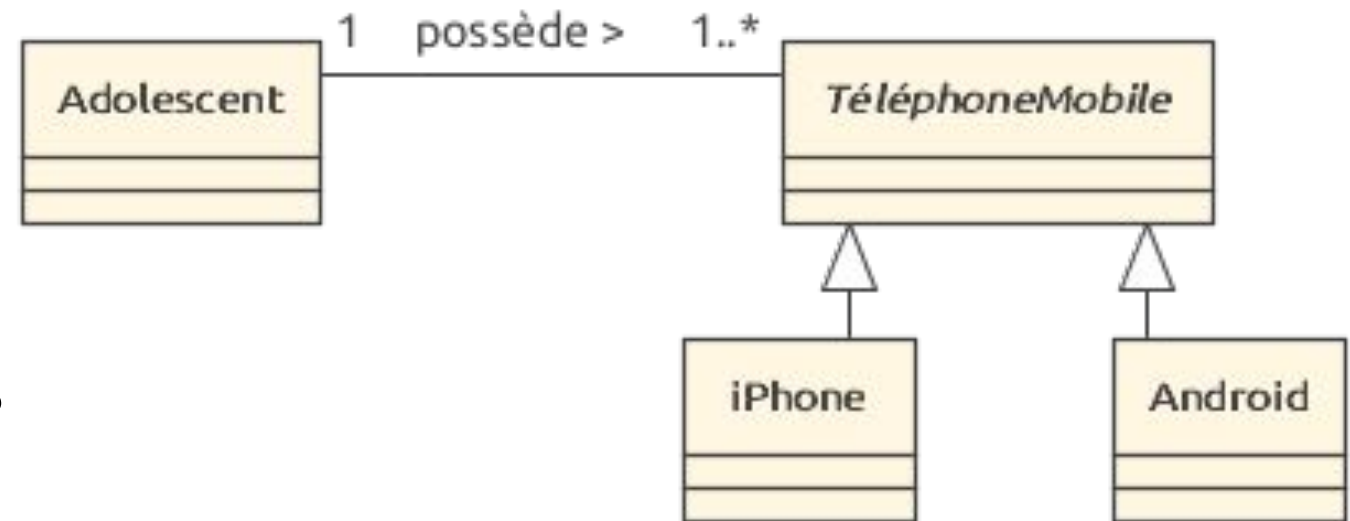
Cardinalités

- Utiliser le diagramme pour répondre aux questions.
 - Combien de téléphones possède chaque adolescent?
 - Peut-il y avoir de téléphones sans propriétaire?
 - Peut-il y avoir un iPhone sans propriétaire?



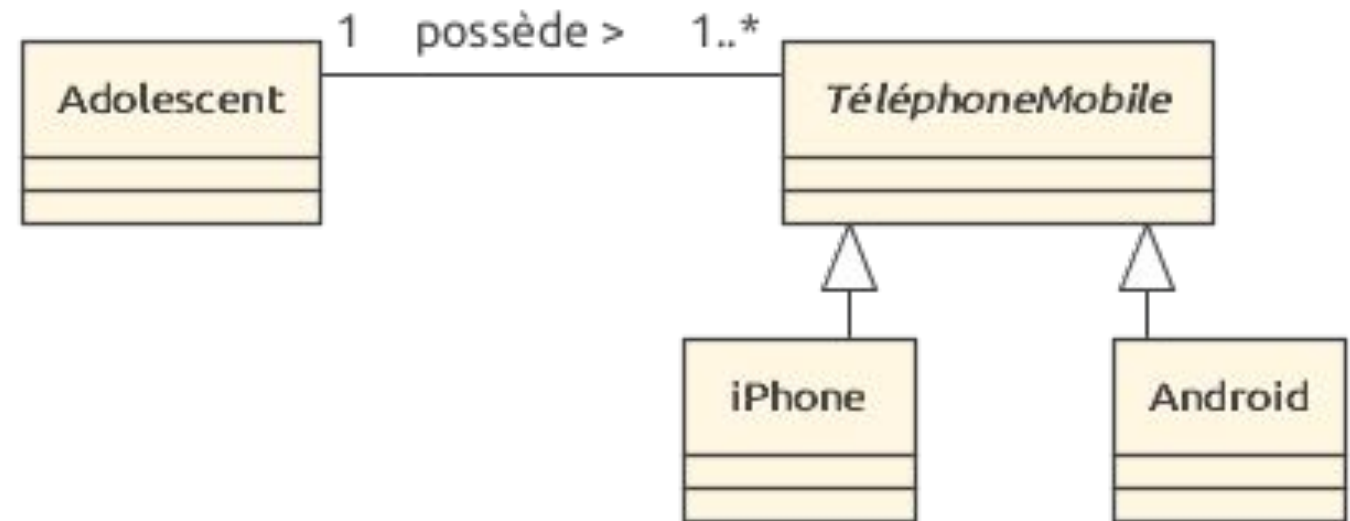
Cardinalités

- Utiliser le diagramme pour répondre aux questions.
 - Combien de téléphones possède chaque adolescent?
 - 1 ou plus
 - Peut-il y avoir de téléphones sans propriétaire?
 - Non
 - Peut-il y avoir un iPhone sans propriétaire?
 - Non. Les objets iPhone héritent l'association avec Adolescent de TéléphoneMobile.



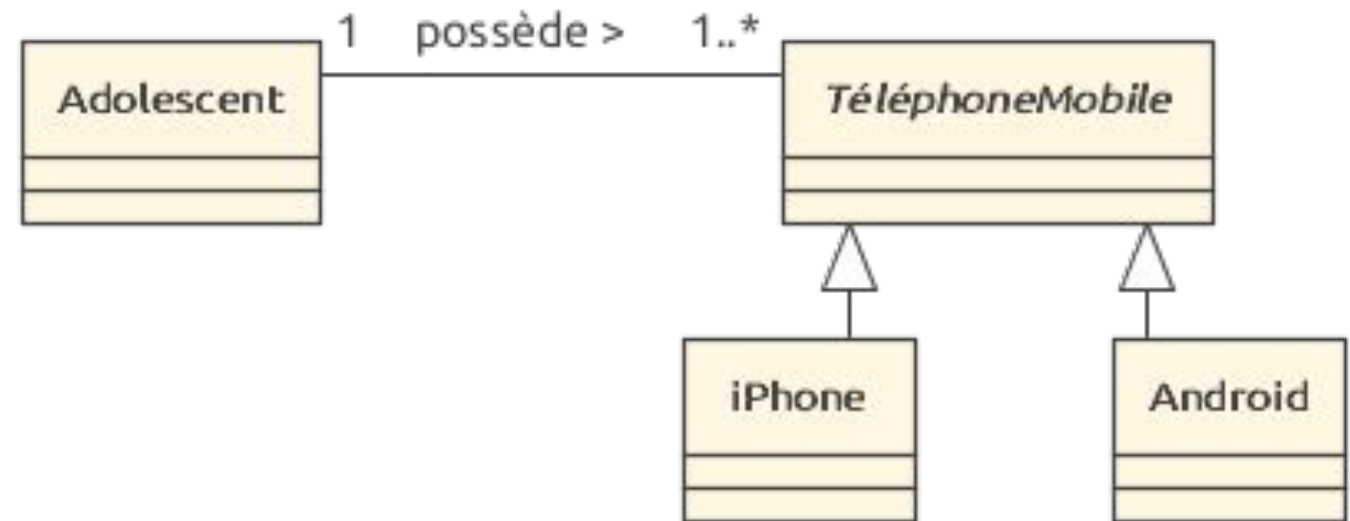
Cardinalités

- Vrai ou faux? Le modèle stipule qu'il y a au moins un adolescent et au moins un téléphone.

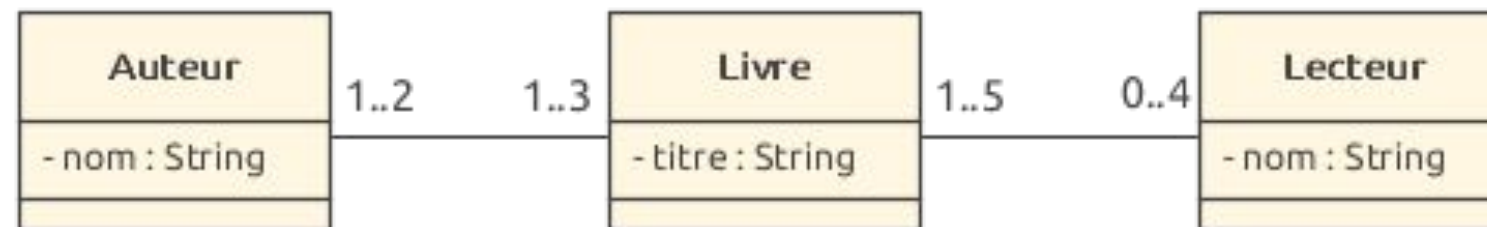


Cardinalités

- Vrai ou faux? Le modèle stipule qu'il y a au moins un adolescent et au moins un téléphone.
 - Non. Le diagramme dit seulement que **si** un adolescent existe il a au moins un téléphone, et **si** il y a un téléphone, il appartient à un adolescent.



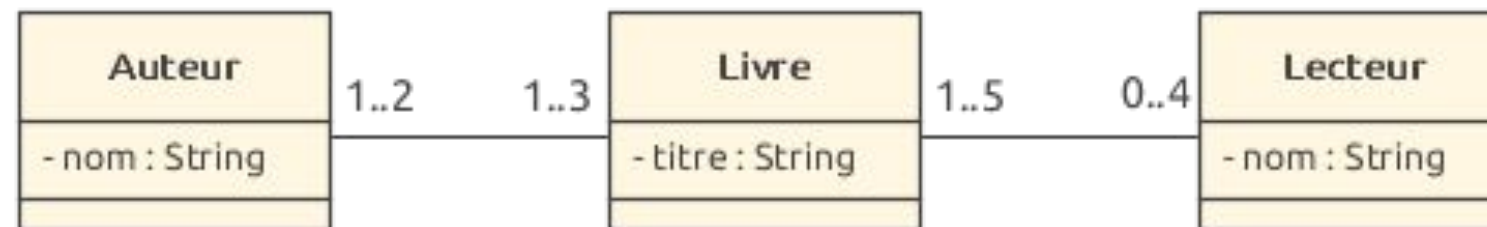
Cardinalités



■ Avec le modèle ci-avant

- S'il y a 6 auteurs, quel est le nombre minimum et maximum de livres et de lecteurs?
- S'il y a 6 lecteurs, quel est le nombre minimum et maximum de livres et d'auteurs?

Cardinalités



■ Avec le modèle ci-avant

- S'il y a 6 auteurs, quel est le nombre minimum et maximum de livres et de lecteurs?
 - Livres 3..18
 - Lecteurs 0..72
- S'il y a 6 lecteurs, quel est le nombre minimum et maximum de livres et d'auteurs?
 - Livres 2..*
 - Auteurs 1..*

Diagramme de classe

■ Vrai ou faux?

- CheckingAccount implémente BankAccount
- CheckingAccount et SavingsAccount sont BankAccount
- CheckingAccount et SavingsAccount sont associés
- BankAccount est associé avec CheckingAccount
- Les objets SavingsAccount ont une méthode processCheck
- Les objets CheckingAccount ont un attribut balance

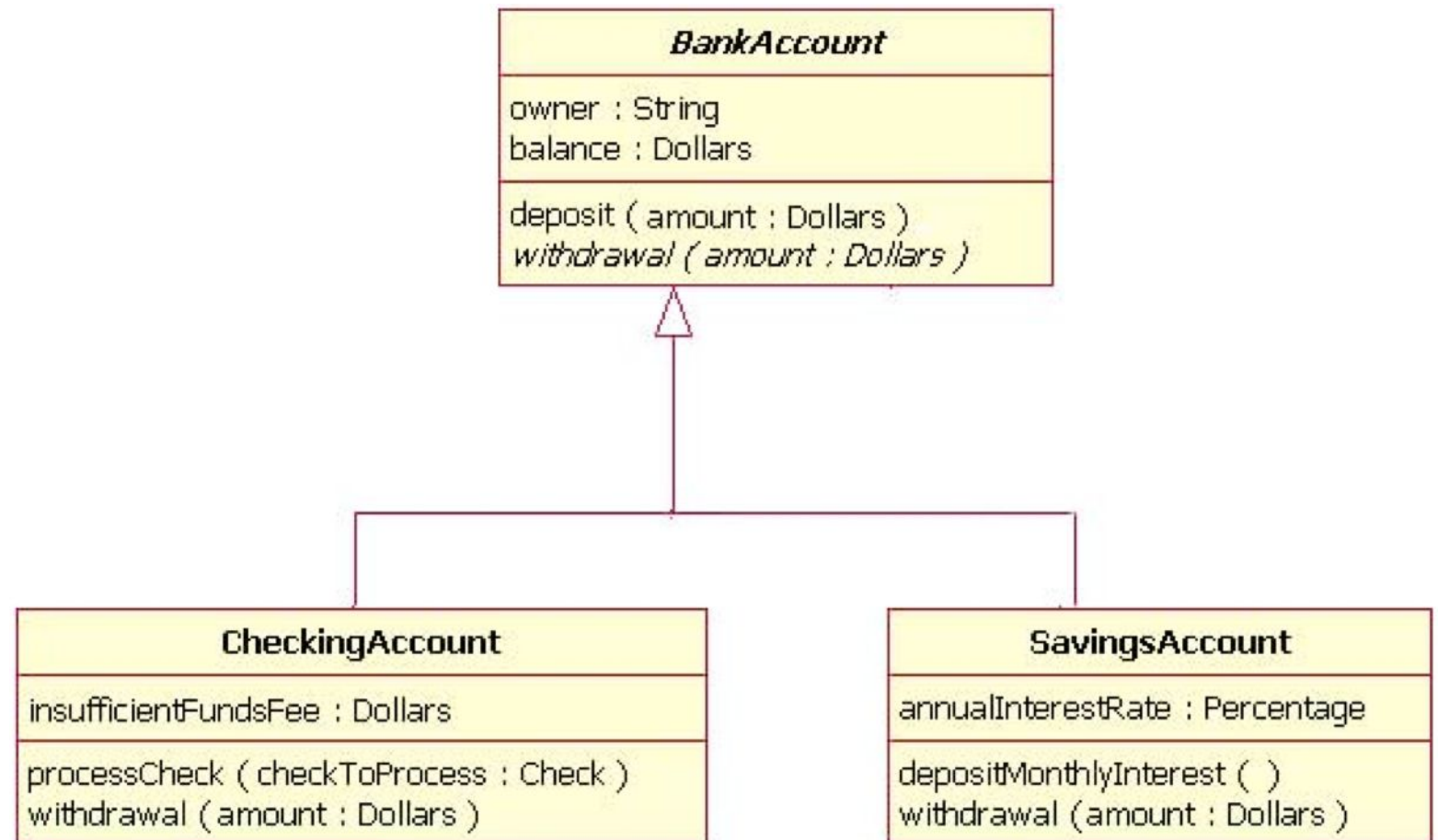
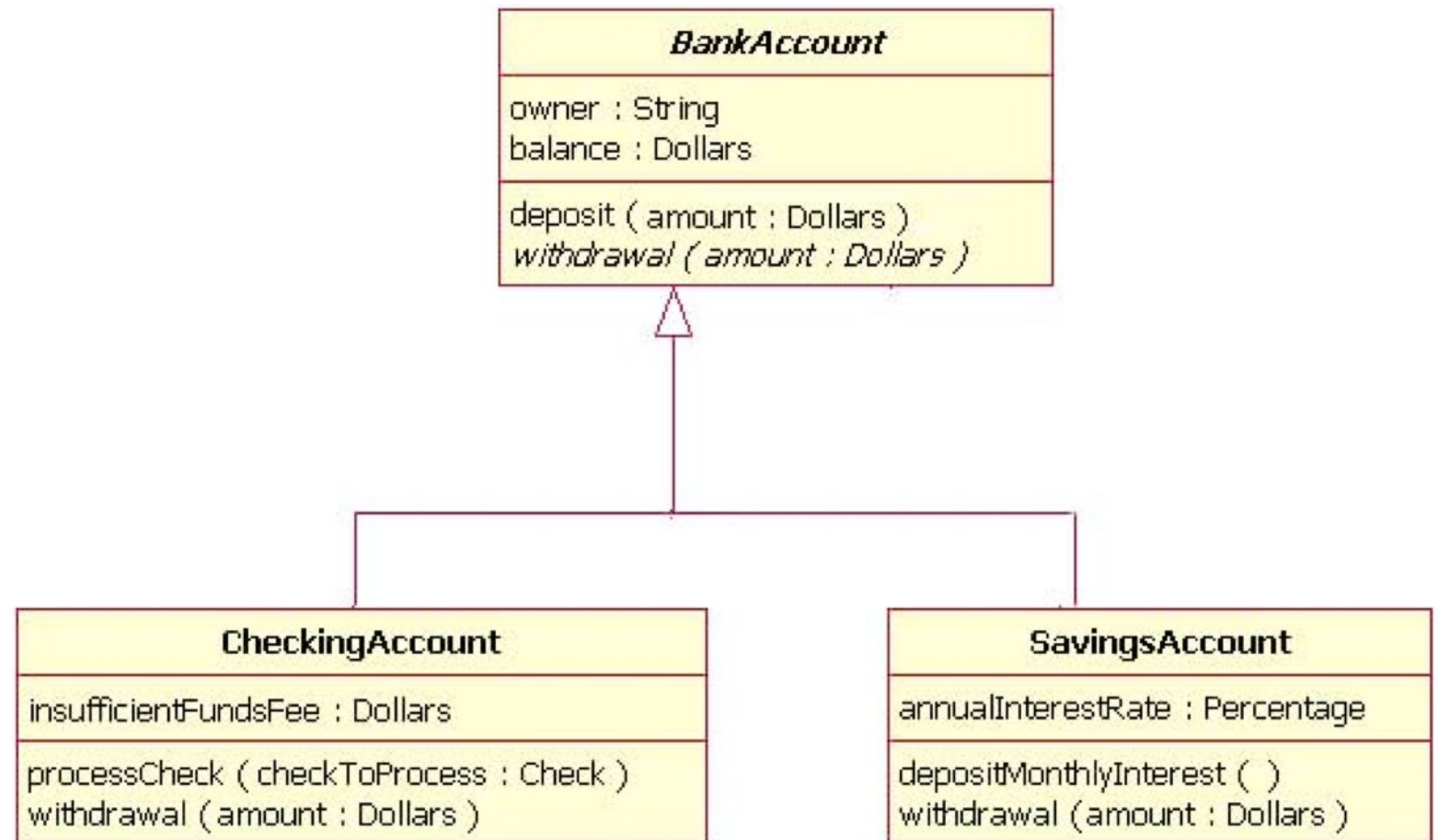


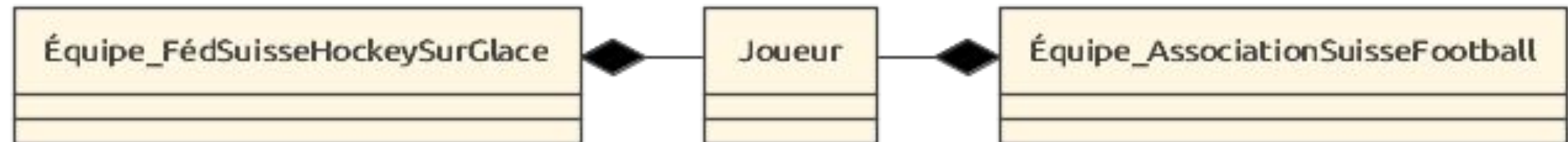
Diagramme de classe

■ Vrai ou faux?

- CheckingAccount implémente BankAccount
→ Il *hérite* de BankAccount
- CheckingAccount et SavingsAccount sont BankAccount
→ Vrai
- CheckingAccount et SavingsAccount sont associés
→ Faux
- BankAccount est associé avec CheckingAccount
→ Faux
- Les objets SavingsAccount ont une méthode processCheck
→ Faux
- Les objets CheckingAccount ont un attribut balance
→ Vrai



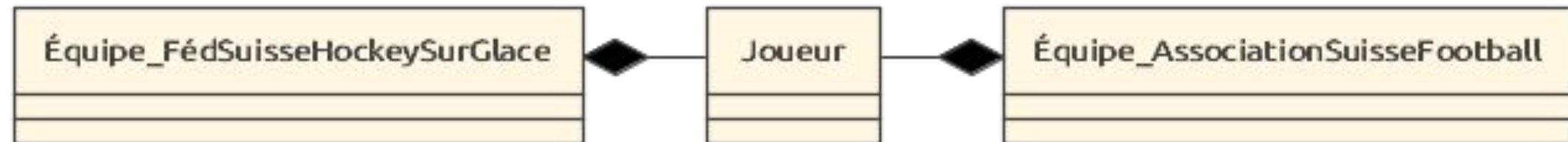
Composition



■ Vrai ou faux?

- Un joueur peut jouer pour une équipe ou de la Fédération Suisse de Hockey sur Glace, ou de l'Association Suisse de Football, mais pas pour les deux.
- Un joueur peut jouer simultanément pour deux équipes FSHG.
- Un joueur peut être transféré d'une équipe à une autre.
- Un joueur doit toujours être membre d'une équipe.

Composition



■ Vrai ou faux?

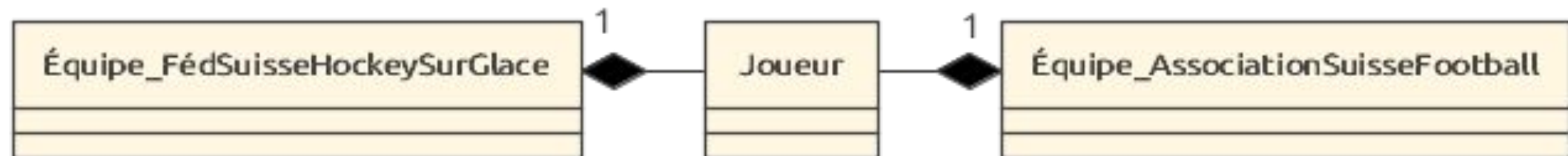
- Un joueur peut jouer pour une équipe ou de la Fédération Suisse de Hockey sur Glace, ou de l'Association Suisse de Football, mais pas pour les deux. → Vrai
- Un joueur peut jouer simultanément pour deux équipes FSHG. → Faux
- Un joueur peut être transféré d'une équipe à une autre. → Vrai
- Un joueur doit toujours être membre d'une équipe. → Faux

Composition



- Qu'est-ce qui ne va pas dans ce modèle?

Composition



- Qu'est-ce qui ne va pas dans ce modèle?
 - Les cardinalités devraient être 0..1 au lieu de 1. Une composante peut faire partie d'au plus **une** composition. Le diagramme dit qu'un joueur fait partie de deux compositions.

Classes abstraites

- Si au moins une méthode d'une classe est abstraite, la classe doit aussi être abstraite. L'inverse est-il vrai, çàd. une classe abstraite doit-elle avoir une méthode abstraite?

Classes abstraites

- Si au moins une méthode d'une classe est abstraite, la classe doit aussi être abstraite. L'inverse est-il vrai, çàd. une classe abstraite doit-elle avoir une méthode abstraite?
 - Non. On peut avoir une classe abstraite qui n'a pas de méthode abstraite. Vous pouvez la spécifier abstraite pour prévenir que l'on puisse en créer des objets.

Contraintes d'intégrité

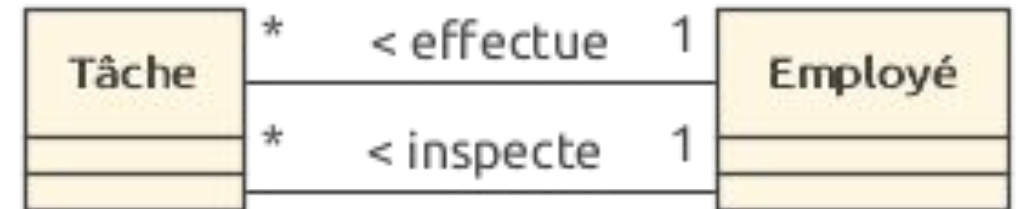
- Établir un modèle UML pour Tâche et Employé et spécifier en français les contraintes d'intégrité qui ne peuvent pas être exprimées dans le modèle:

Dans une entreprise qui produit des stimulateurs cardiaques la production d'un stimulateur est subdivisée en tâches. Chaque tâche est effectuée par un employé. Chaque tâche doit être inspectée par un autre employé pour assurer qu'il n'y a pas eu d'erreurs.

Contraintes d'intégrité

- Établir un modèle UML pour Tâche et Employé et spécifier en français les contraintes d'intégrité qui ne peuvent pas être exprimées dans le modèle:

Dans une entreprise qui produit des stimulateurs cardiaques la production d'un stimulateur est subdivisée en tâches. Chaque tâche est effectuée par un employé. Chaque tâche doit être inspectée par un autre employé pour assurer qu'il n'y a pas eu d'erreurs.



L'employé qui inspecte la tâche ne peut pas être celui qui effectue la tâche.