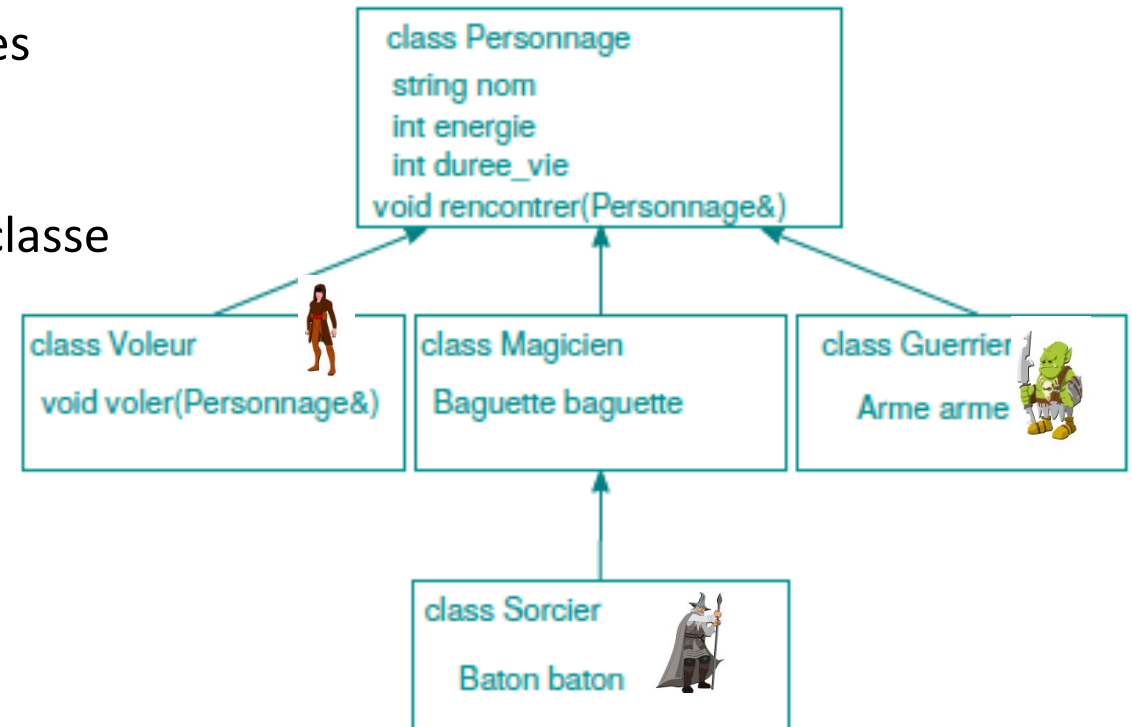


# Mise en œuvre du principe de ré-utilisation avec l'héritage

**Motivation: lorsqu'un ensemble d'entité présente un forte cohérence sémantique**

- Eviter de dupliquer du code manipulant des propriétés communes
- Partager un sous-ensemble de propriétés/méthodes communes dans une superclasse
- Définir des classes spécialisées à partir de la superclasse
- Environnement professionnel:  
Offrir une base de code cohérente et **validée** pouvant être seulement **étendue** par l'**ajout de sous-classes**

(exemple en Robotique dans présentation indépendante)

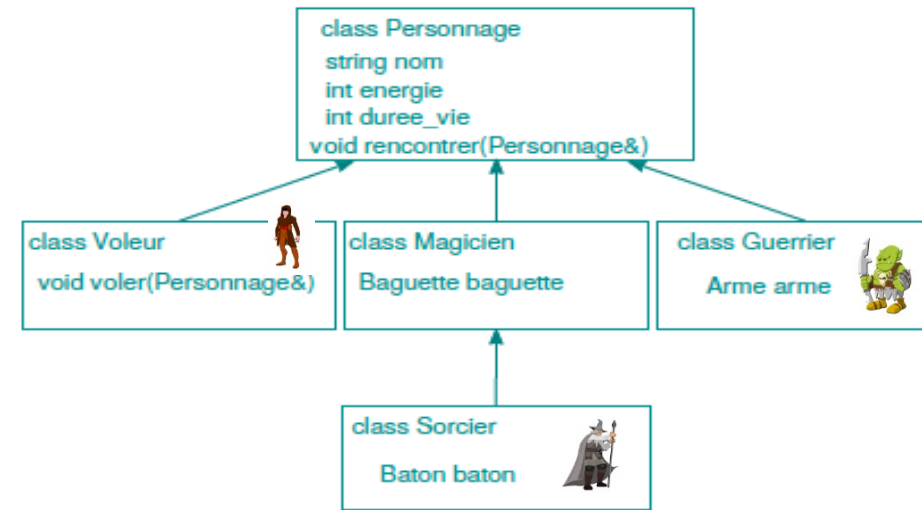


# Distinguer “Est un(e)” de “Possède un(e)”

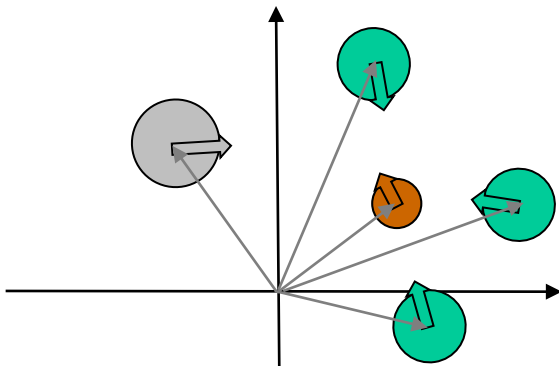
La relation « Possède un(e) » est la relation déjà connue entre **une instance d’une classe** et **ses attributs** :  
=> Un Personnage possède : nom, energie, duree\_vie

La relation « Est un(e) » est celle qui existe entre **une sous-classe** et **sa classe parente**:

- ⇒ Un Sorcier est un Magicien
- ⇒ Un Magicien est un Personnage



**Question SpeakUp:** supposons que pour les besoins d’une simulation de scénario de film, je veux pouvoir éviter des collisions et calculer des stratégies de déplacement. Pour cela l’information de **position**, **orientation** et de **rayon d’espace occupé** des personnages est nécessaire.



Quelle approche est la plus pertinente :

- A Créer des nouvelles sous-classes
- B Ajouter l’information à chaque sous-classe existante
- C Créer une nouvelle superclasse dont Personnage va hériter
- D Ajouter l’information à la classe Personnage

# Avant de vous lancer avec l'héritage, il faut :

- **Savoir *utiliser* une hiérarchie de classes (ex: GTKmm)**
- Maîtriser le **principe de ré-utilisation** sans héritage
  - Identifier les limitations de cette approche (gestion de la complexité)
- Maîtriser l'analyse des relations entre les éléments d'un modèle
  - pour distinguer deux types de relations (diagramme de classes):
    - les relations « **Possède un(e)** »
      - => conception classique = attribut d'une classe ou struct
    - Des relations « **Est un(e)** »
      - => conception d'une hiérarchie de classes
- Guide: [Principe de Parcimonie](#)