2 minutes – Modélisation objets

Création d'un diagramme de classe à partir d'un cahier des charges

1- Lister toutes les classes potentielles
2- Trouver les classes sœurs (ajout de classe possible)
3- Abstraction (ajout de classe possible)
4- Héritage
5- Association
6- Interface

Héritage

Règle d'héritage : A est_un (type particulier de) B

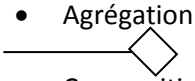
Anti-règles

1- A est_un(e instance de B) – Exemple : Sarkozy/Président
2- Héritage avec restriction
   ➔ Une classe concrète ne peut pas hériter d'une autre classe concrète
3- B est_un (type de donnée) de A
   ➔ Il faut utiliser l'association dans ce cas
4- Héritage multiple et B est_un (A-able)
   ➔ Il faut utiliser des interfaces

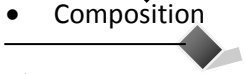
Association

- Simple association

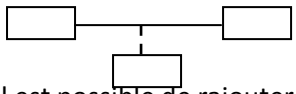
- Agrégation



- Composition



L'agrégation et la composition sont des relations dissymétriques de type conteneur/contenu, maitre/esclave.
Lors de l'agrégation, la destruction n'est pas systématique, alors qu'elle est systématique dans le cas de la composition.
Il existe aussi des classes d'association



Il est possible de rajouter des flèches pour indiquer plus de clarté dans le dessin.

Interface

Type abstrait qui illustre des méthodes abstraites (⇔l'ensemble des fonctionnalités)

MISC

Une **modélisation** est à une classe ce qu'une classe est à un objet.
Un itérateur est un objet utilitaire pour manipuler des conteneurs. **VRAI**
Une classe abstraite est instanciable. **FAUX**
UML = **Unified Modeling Language**
Type d'attribut ou méthode : public, private, protected
L'héritage a une notion « a un » **FAUX**
Une méthode constante pour être appelé sur un objet non constant. **VRAI**
Les trois axes d'un logiciel sont l'axe statique, l'axe dynamique et **l'axe fonctionnel**
Un diagramme de cas d'utilisation est comme un film : il y a des acteurs. **VRAI**