Plan du chapitre

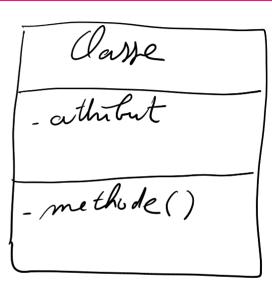
Le paradigme objet

2 Les objets Les classes

Associations entre classes

Rappel

- Les classes
 - Attributs
 - Méthodes
- Représentation UML
- Règles de nommage
 - Classes
 - Attributs
 - Méthodes

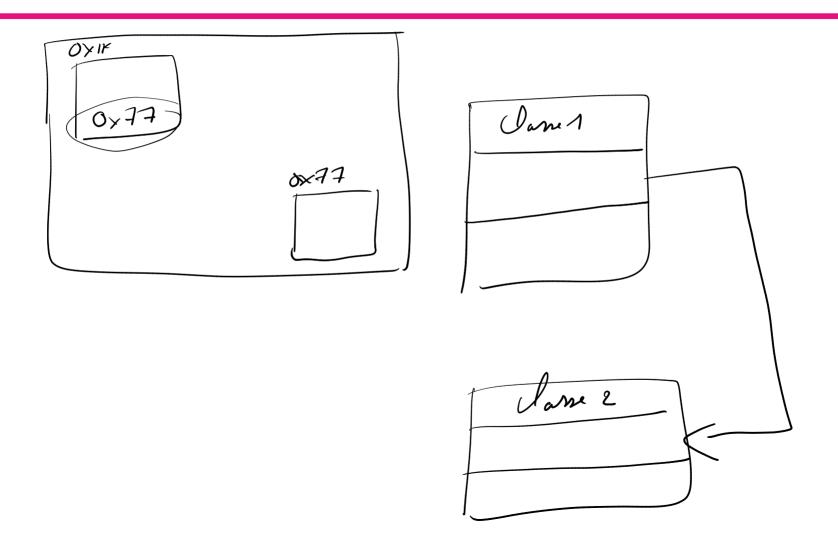


PascalCase camelCase camelCase snake_case

Les relations

- 3 types de relations entre classes
 - Association
 - Dépendance
 - Héritage

Association

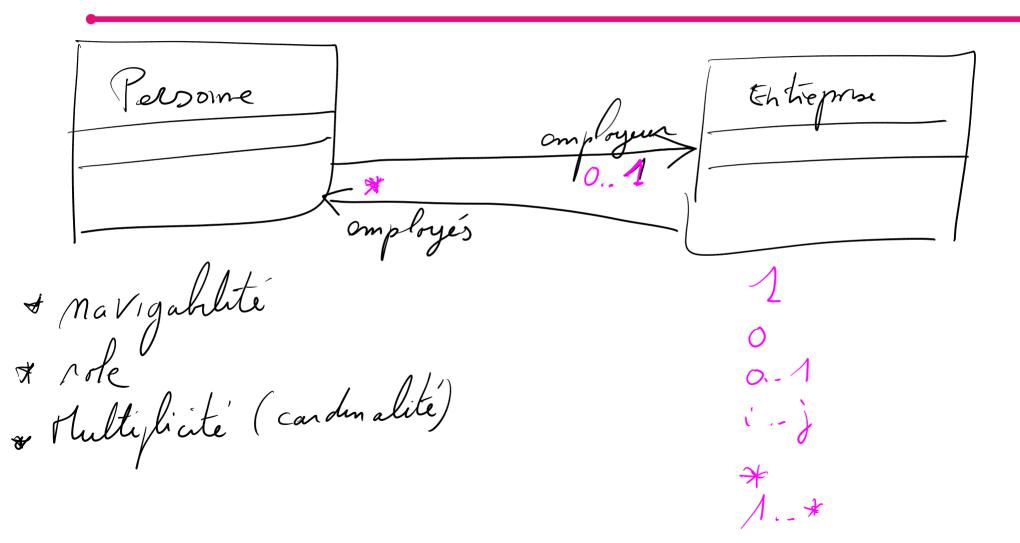


Code Java

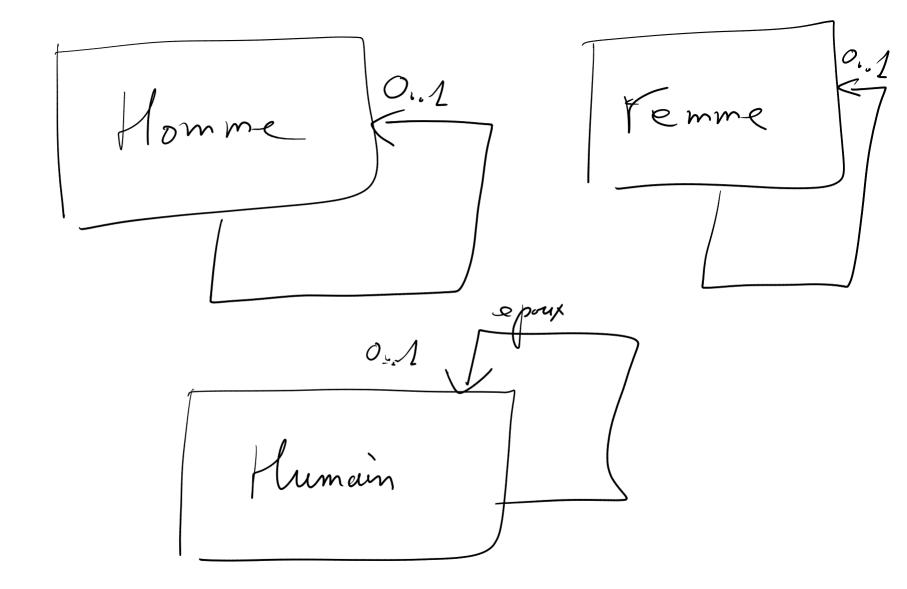
donnée membre

```
class Personne {
   Entreprise employeur;
                                     Personne
                                                         Entreprise
       Entreprise
class Personne
   Personne[] employes;
Personne toto = new Personne();
Entreprise airbus = new Entreprise();
toto.employeur = airbus;
airbus.employe[0] = toto;
```

Décoration des associations



l'om me mono gamie 0.1 Solggamie Joly gyme Joly andrie celibat

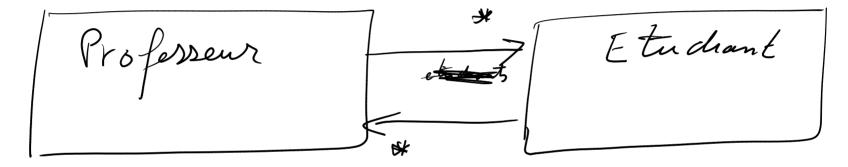


Type d'association

- Intention
 - Ajouter de la sémantique à la modélisation
 - Implication dans le code
- 3 types
 - Normale —
 - Agrégation
 - Composition

Association: normale

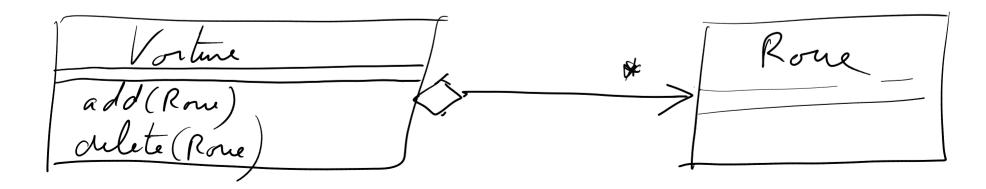
• Sémantique : connait



Association: agrégation

Sémantique : possède (ensemble)

| Relation | Exemple |
|----------------------|---------------------|
| Composé / Composant | Voiture / Roues |
| Collection / Élément | Forêt / Arbres |
| Espace / Position | Désert / Oasis |
| Événement / Étape | Document / Chapitre |



Association: composition

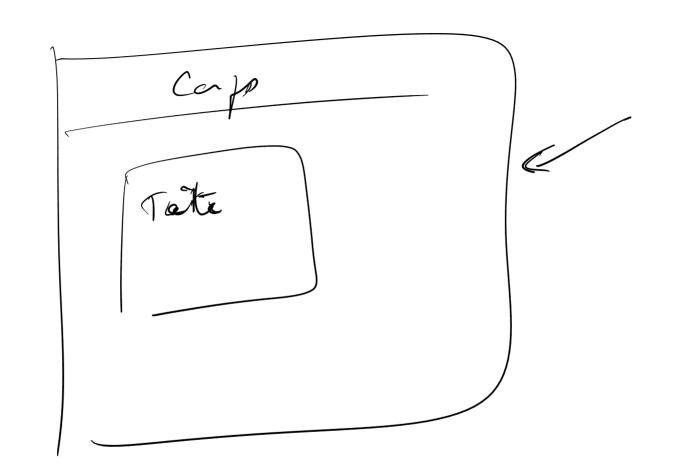
Sémantique : est constitué de

| Relation | Exemple |
|---------------------|------------------|
| Corps / Portion | Corps / Tête |
| Matière / Substance | Eau / Hydrogène |
| Activité / Phase | Achat / Paiement |

Corps

Corps

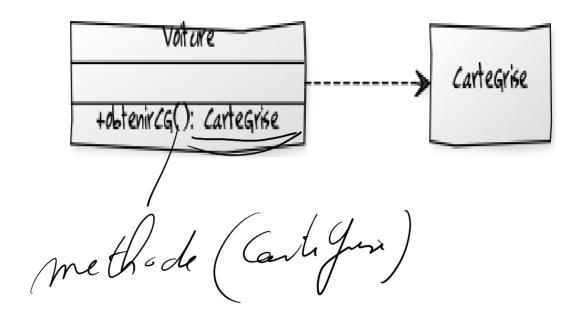
Tete



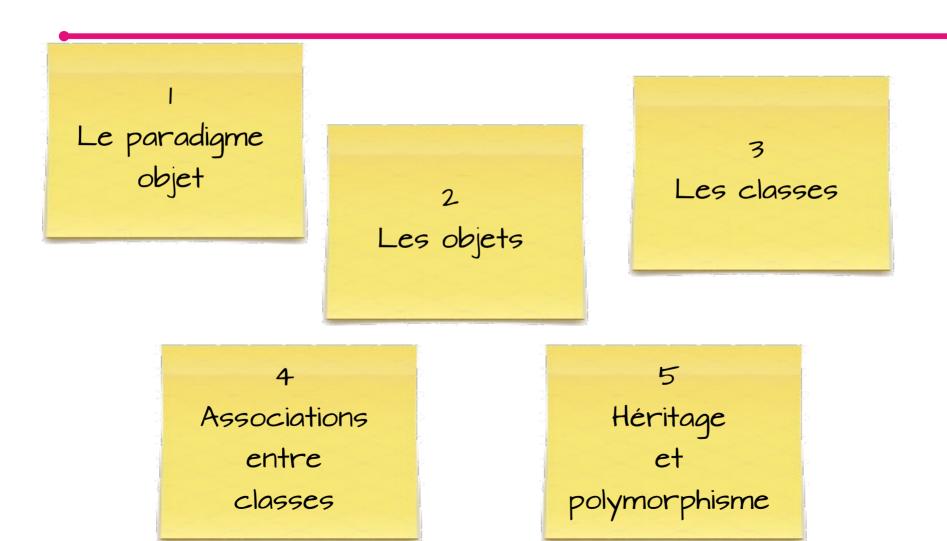
- Joker → Cheval
- Éleveur → Cheval
- Éleveur → Tête
- Cheval → Cœur
- Cheval \rightarrow Selle
- École → Étudiant
- Carte mère → microprocesseur
- GAB → Billet

A: Normale B: Agrégation C: Composition

Dépendance



Plan du chapitre

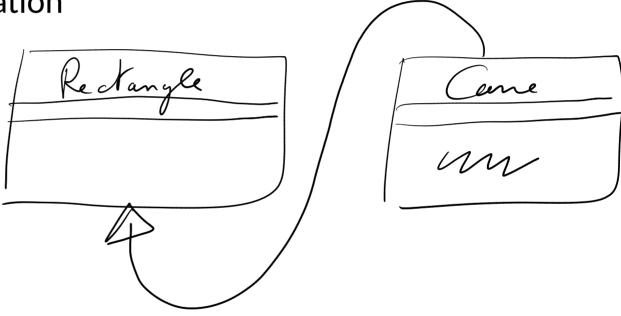


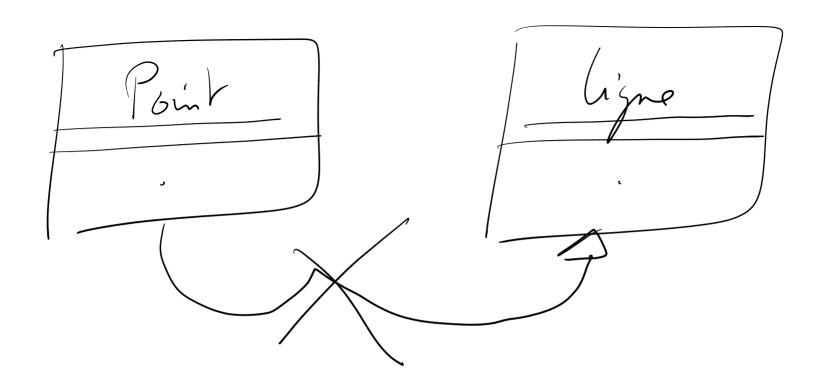
Héritage

Intention 1 : factorisation Carré Cercle dimension = 2 dimension = 2 / couleur= "blue" / couleur= "vert" < position = (54,88)position = $(54,88)_{7}$ largeur= 10 / largeur = 10 hauteur=10 ~ Forme hauteur=10 Cane me thode new (arrel) forme

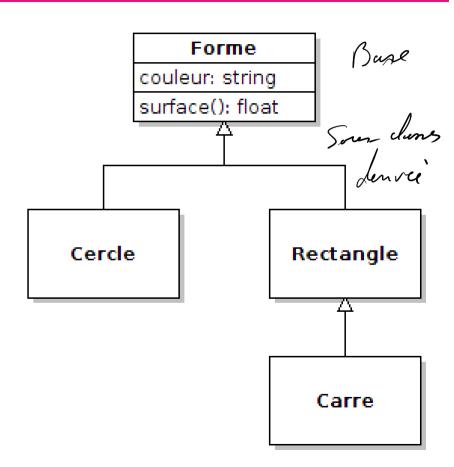
Héritage

Intention 2 : spécialisation



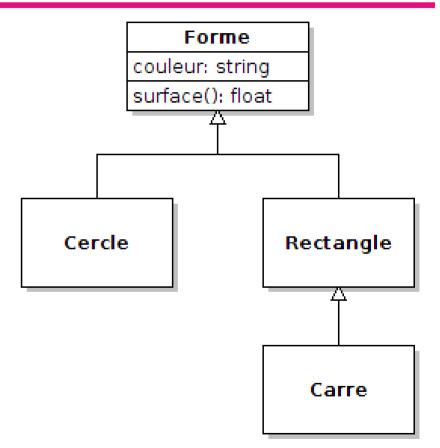


Héritage



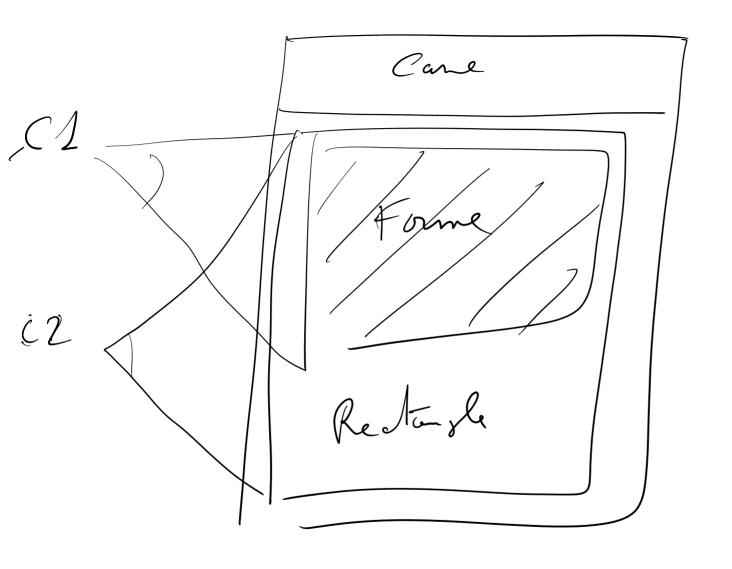
Héritage en Java

```
class Forme {
                               String couleur;
                       properties a filter of the properties of the pr
 class Cercle extends Forme {
class Rectangle extends Forme {
class Carre extends Rectangle {
```



Transtypage

Forme (1= new (ane()) C1. surfacel) Rectangle chimmet) Rectangle c2= new Came(); (1. surface() (arre (37 new (anel) Cl. converse () c 1. 20m ()



Form rz new Rectangle (); R. Surface () A. invent Rectangle) Ti) de danement. R. Boom () Rectangle 22 = 17. surfacul) Cane 132 (Cane) 22.