

Programmation orientée objet

Association

BTS SIO 1 – SLAM2

Programme

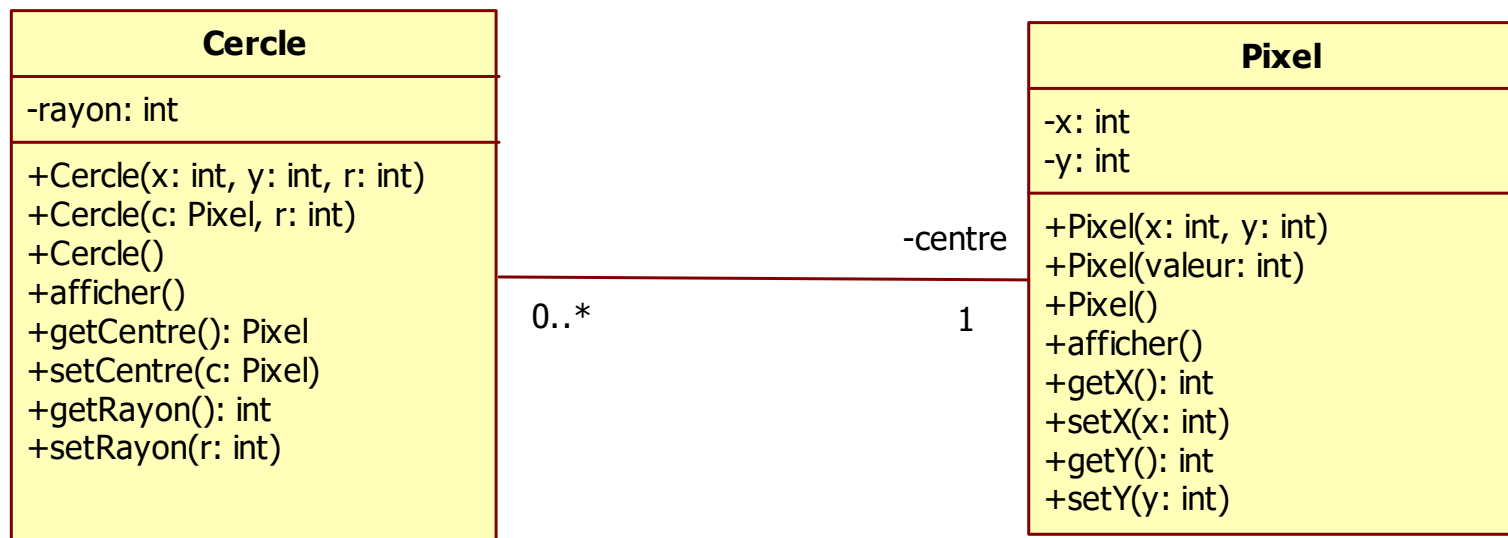
- Programmation Orientée Objet (POO)
 - Association entre classes en UML
 - Association entre classes en java

Association entre classes

- Dans une situation de modélisation, il est souvent nécessaire de relier deux objets ensemble, pour :
 - Compléter la description d'un premier objet complexe par un deuxième objet complexe
 - Associer deux objets complexes qui ont un lien hiérarchique ou sémantique

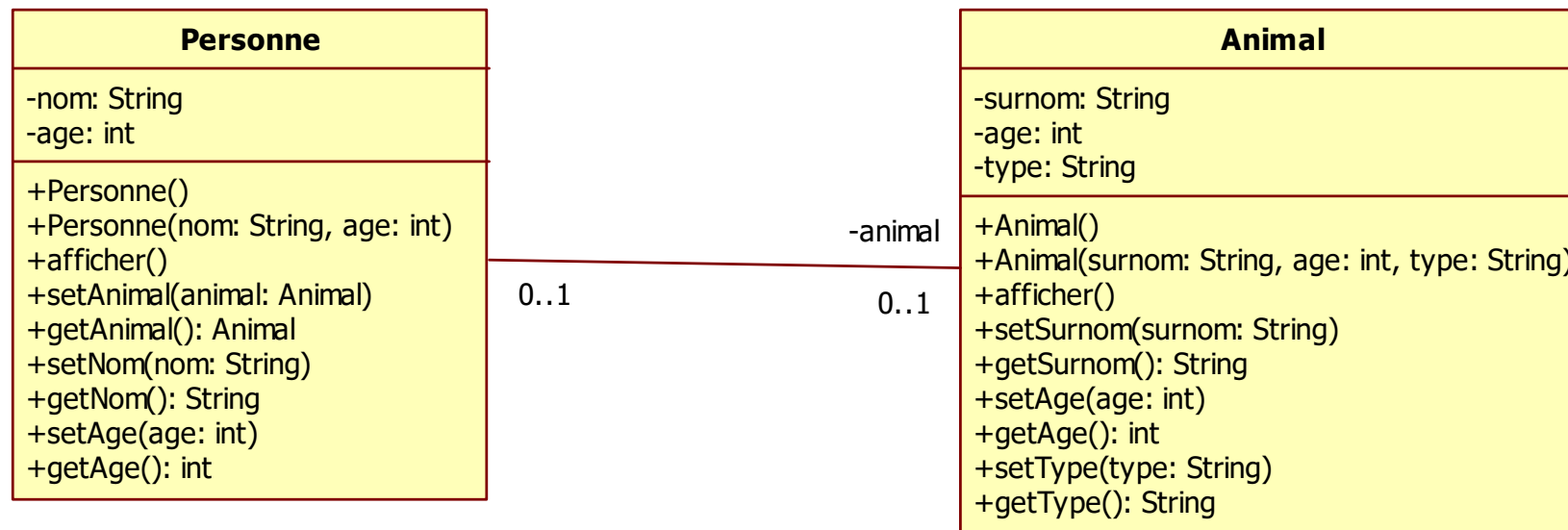
Exemple 1

- Un cercle est décrit par son centre (de type Pixel) et un rayon (de type entier)



Exemple 2

- Une personne est décrite par un nom (type String) et un âge (type entier) et elle peut être associée à un animal (type Animal)



Association en UML

- Le plus souvent une association ne relie que deux classes.
- Une association peut être identifiée par un nom et chacune de ses extrémités définit le nombre d'instances des classes reliées qui sont impliquées dans cette association (multiplicité).

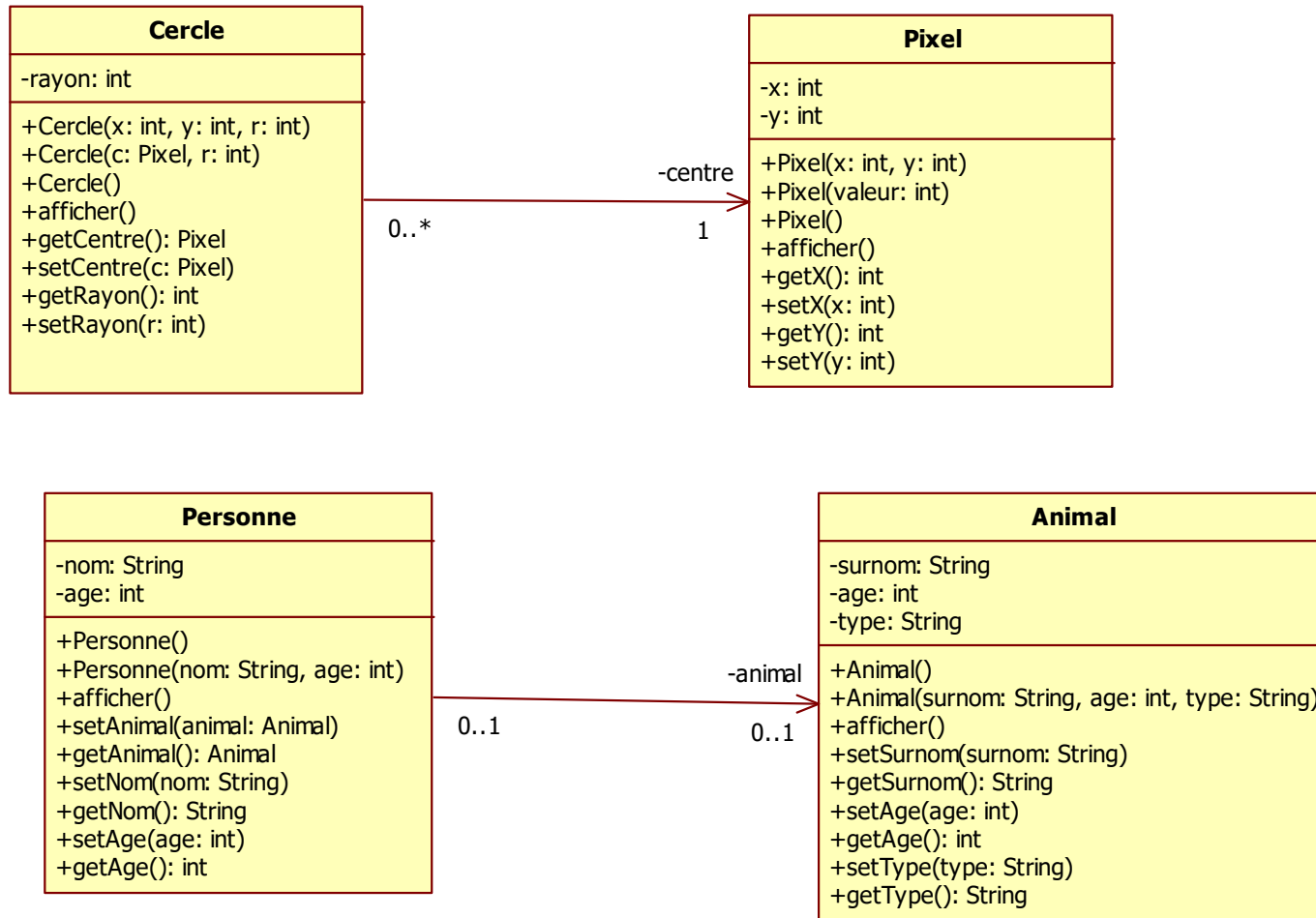
Multiplicité

- A chaque extrémité d'une association est indiqué la multiplicité :
 - 1 : un et un seul
 - 0..1 : 0 ou un seul
 - 0..* ou * : 0 ou plusieurs
 - 1..* : 1 ou plusieurs
 - N : exactement N
 - N..M : de N à M
- Par défaut, une extrémité est à 1

Unidirectionnelle ou Bidirectionnelle

- Les associations peuvent être dirigées, ce qui contraint la visibilité et la navigation dans le modèle.
- La direction se représente par une flèche classique.
- Par défaut, s'il n'y a pas de flèche, l'association est bidirectionnelle (comme s'il y avait une flèche à chaque extrémité de l'association).

Exemples modifiés



L'association en Java

- Mise en place de l'association à l'aide d'attributs
- Les attributs sont créés dans l'une ou l'autre des classes selon la navigabilité de l'association
- Matérialisation de l'association au moment de la construction des instances
- Quand l'association est multiple, elle est représentée à l'aide d'un tableau ou d'une collection (Tableau dynamique)

Classe Cercle

```
class Cercle{
    int rayon; // attribut rayon
    Pixel centre; //attribut centre (voir association !!)
    // Constructeur
    Cercle( int x, int y, int r) {
        this.rayon = r;
        this.centre = new Pixel(x,y);
    }
    //méthode pour afficher les infos de l'objet
    void afficher () {
        System.out.println ("Cercle avec un rayon de " +
            this.rayon + "et un centre au(" + this.centre.getX() +
            ";" + this.centre.getY() + ")");
    }
    ...
}
```