

# La Programmation Orientée Objet (POO) avec Java

## Module 4 – Les associations



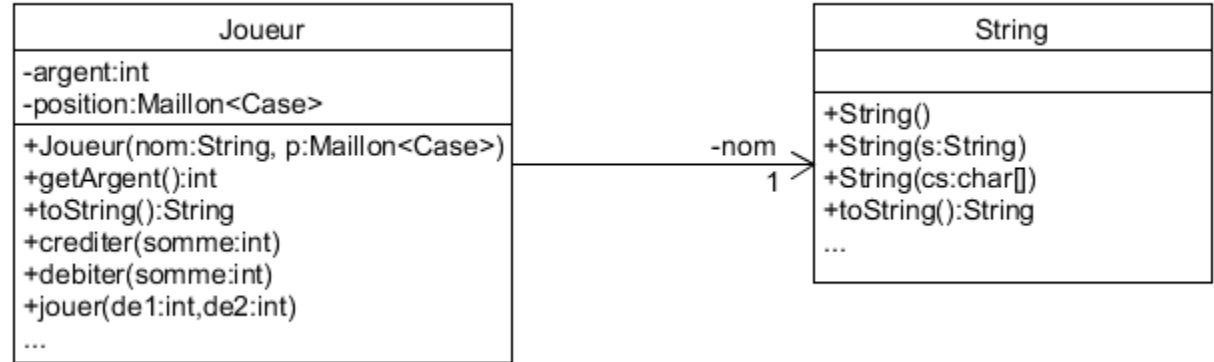
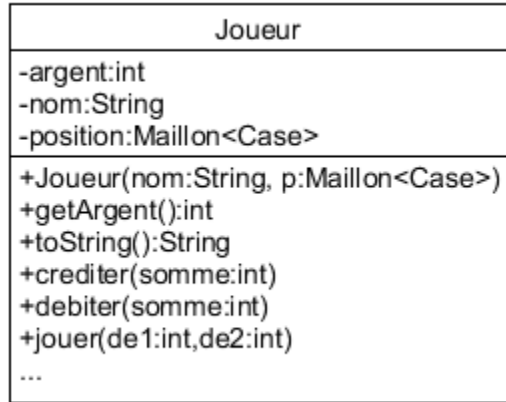
# Objectifs

- Comprendre ce qu'est une association entre deux classe
- Créer des associations unidirectionnelles et bidirectionnelles

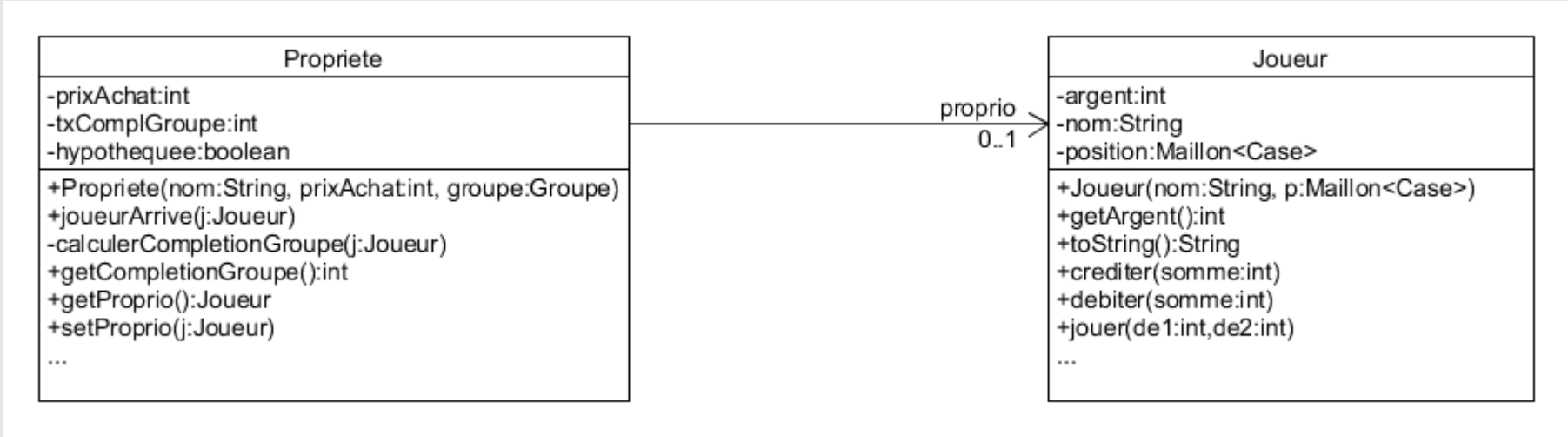
# Les associations unidirectionnelles

```
public class Joueur {  
  
    private int argent;  
    private String nom;  
    ...  
}
```

Équivalent à



# Les associations unidirectionnelles



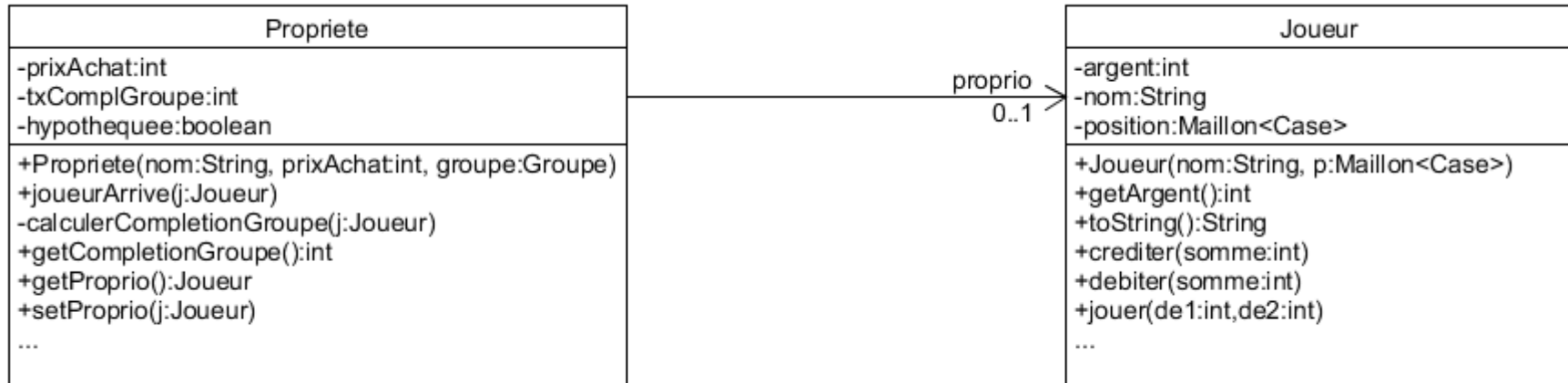
# Les associations unidirectionnelles

```
public class Joueur {  
  
    private int argent;  
    private String nom;  
    ...  
}
```

```
public class Propriete {  
  
    private int prixAchat;  
    private Joueur proprio;  
    private int txComplGroupe;  
    private boolean hypothee;  
  
    public Joueur getProprio() {  
        return proprio;  
    }  
  
    public void setProprio(Joueur j) {  
        Joueur ancienProprio = this.proprio;  
        this.proprio = j;  
        if(ancienProprio != null)  
            this.calculerCompletionGroupe(ancienProprio);  
        this.calculerCompletionGroupe(j);  
    }  
    ...  
}
```

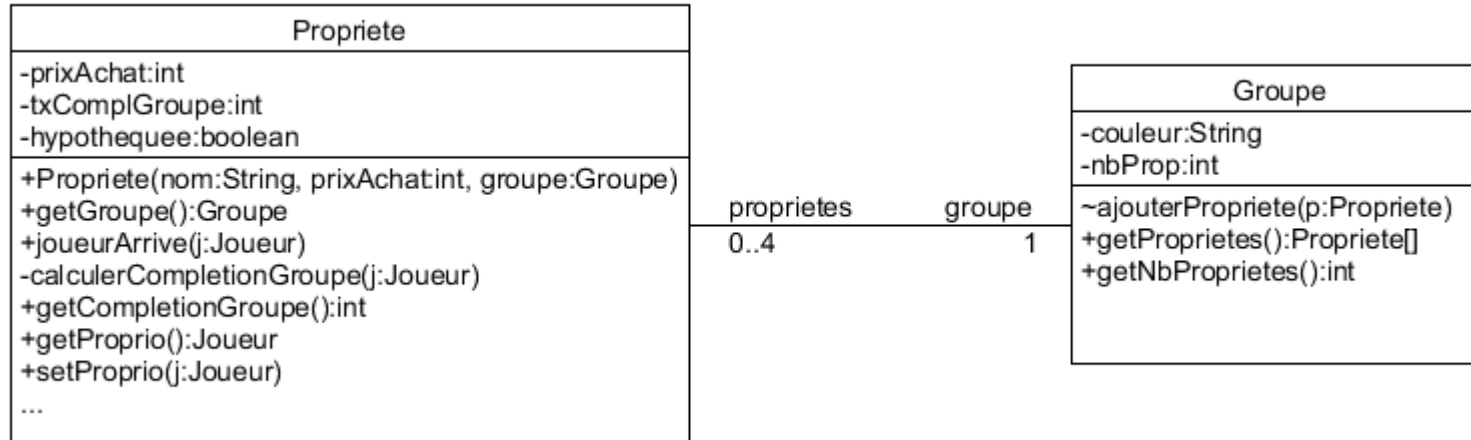
# Les associations unidirectionnelles

- Une association unidirectionnelle n'est navigable que dans un seul sens
  - Dans l'exemple précédent :
    - Une propriété sait quel est son propriétaire
    - Un joueur ne connaît pas les propriétés qu'il possède



# Les associations bidirectionnelles

- La navigation est possible dans les deux sens
  - Exemple :
    - Une propriété sait à quel groupe de propriétés elle appartient
    - Un groupe de propriété sait quelles sont les propriétés qui la compose



# Les associations bidirectionnelles

```
public class Propriete {
    private int prixAchat;
    private Joueur proprio;
    private Groupe groupe;
    private int txComplGroupe;
    private boolean hypothecuee;

    public Propriete(int prixAchat, Groupe groupe) {
        this.groupe = groupe;
        groupe.ajouterPropriete(this);
        this.prixAchat = prixAchat;
        this.txComplGroupe = 0;
        this.hypothecuee = false;
    }

    public Groupe getGroupe() {
        return groupe;
    }

    ...
}
```

```
public class Groupe {
    private Propriete[] proprietes
                        = new Propriete[4];
    private int nbProp = 0;

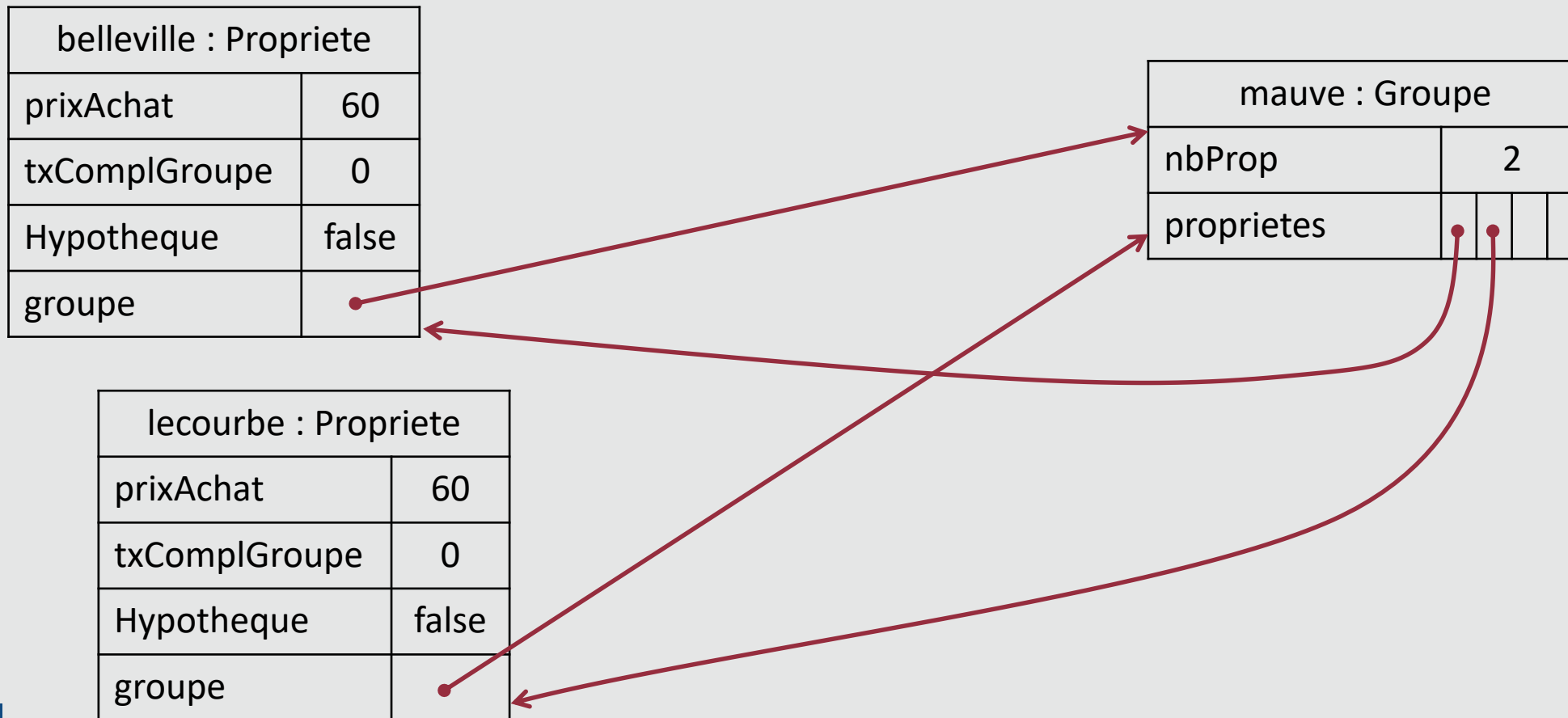
    void ajouterPropriete(Propriete p) {
        if(this.equals(p.getGroupe())) {
            this.proprietes[this.nbProp]=p;
            this.nbProp++;
        }
    }

    public Propriete[] getProprietes() {
        return this.proprietes;
    }

    public int getNbProprietes() {
        return this.nbProp;
    }
}
```



# Les associations bidirectionnelles



# Les associations bidirectionnelles

belleville : Propriete	
prixAchat	60
txComplGroupe	0
Hypothèque	false
groupe	

mauve : Groupe	
nbProp	1
proprietes	

 Incohérences possibles !

lecourbe : Propriete	
prixAchat	60
txComplGroupe	0
Hypothèque	false
groupe	

bleu : Groupe	
nbProp	1
proprietes	

# Les associations bidirectionnelles

```
public class Propriete {
    private int prixAchat;
    private Joueur proprio;
    private Groupe groupe;
    private int txComplGroupe;
    private boolean hypothecuee;

    public Propriete(int prixAchat, Groupe groupe) {
        this.groupe = groupe;
        groupe.ajouterPropriete(this);
        this.prixAchat = prixAchat;
        this.txComplGroupe = 0;
        this.hypothecuee = false;
    }

    public Groupe getGroupe() {
        return groupe;
    }
    ...
}
```

```
public class Groupe {
    private Propriete[] proprietes
        = new Propriete[4];
    private int nbProp = 0;

    void ajouterPropriete(Propriete p) {
        if(this.equals(p.getGroupe())) {
            this.proprietes[this.nbProp]=p;
            this.nbProp++;
        }
    }

    public Propriete[] getProprietes() {
        return this.proprietes;
    }

    public int getNbProprietes() {
        return this.proprietes.size();
    }
}
```

Les associations

TP

