Java
Associations

Objectifs

- Comprendre la notion d’association
- Créer des associations unidirectionnelles et bidirectionnelles

Une association

- On appelle « association » toute utilisation d’une classe depuis une autre classe.

Une association

- On appelle « association » toute utilisation d'une classe depuis une autre classe.
  - Par exemple, utiliser la classe « LocaleDate » dans notre classe « Patient » est une association

```java
package bo;

import java.time.LocalDate;

public class Patient {
        private String nom;
        private String prenom;
        private String telephone;
        private char sexe;
        private long numeroSecu;
        private LocalDate dateDeNaissance;
        private String commentaires;
```

les attributs utilisent ici une autre classe:
-LocalDate
-String

Une association

- On appelle « association » toute utilisation d'une classe depuis une autre classe.
  - Par exemple, utiliser la classe « LocaleDate » dans notre classe « Patient » est une association
  - ou même utiliser la classe « String »…

```java
package bo;

import java.time.LocalDate;

public class Patient {
        private String nom;
        private String prenom;
        private String telephone;
        private char sexe;
        private long numeroSecu;
        private LocalDate dateDeNaissance;
        private String commentaires;
```

De patient on retrouve la date de naissance
A l'inverse date de naissance ne permet pas de retrouver le patient

Une association

- On appelle « association » toute utilisation d'une classe depuis une autre classe.
  - Par exemple, utiliser la classe « LocaleDate » dans notre classe « Patient » est une association
  - ou même utiliser la classe « String »…

```java
package bo;

import java.time.LocalDate;

public class Patient {
        private String nom;
        private String prenom;
        private String telephone;
        private char sexe;
        private long numeroSecu;
        private LocalDate dateDeNaissance;
        private String commentaires;
```

- Dans cet exemple, l'association est dite « unidirectionnelle ».

Les associations unidirectionnelles

- Une association unidirectionnelle n’est navigable que dans un seul sens.

Les associations unidirectionnelles

- Une association unidirectionnelle n’est navigable que dans un seul sens.

```
public class Avion {

    private String carburant;
    private int nbPlace;
```

Avion
- carburant : String - nbPlaces : int
getters et setters…

Les associations unidirectionnelles

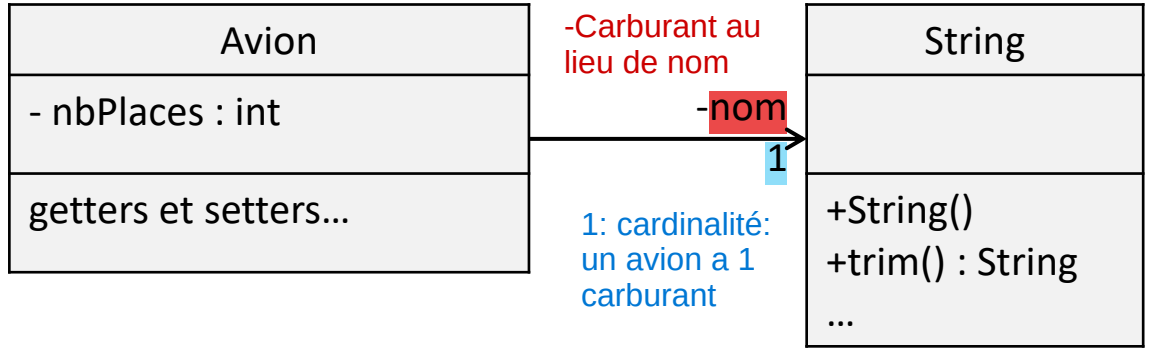
- Une association unidirectionnelle n'est navigable que dans un seul sens.

```
public class Avion {

    private String carburant;
    private int nbPlace;
```

Avion
- carburant : String - nbPlaces : int
getters et setters…

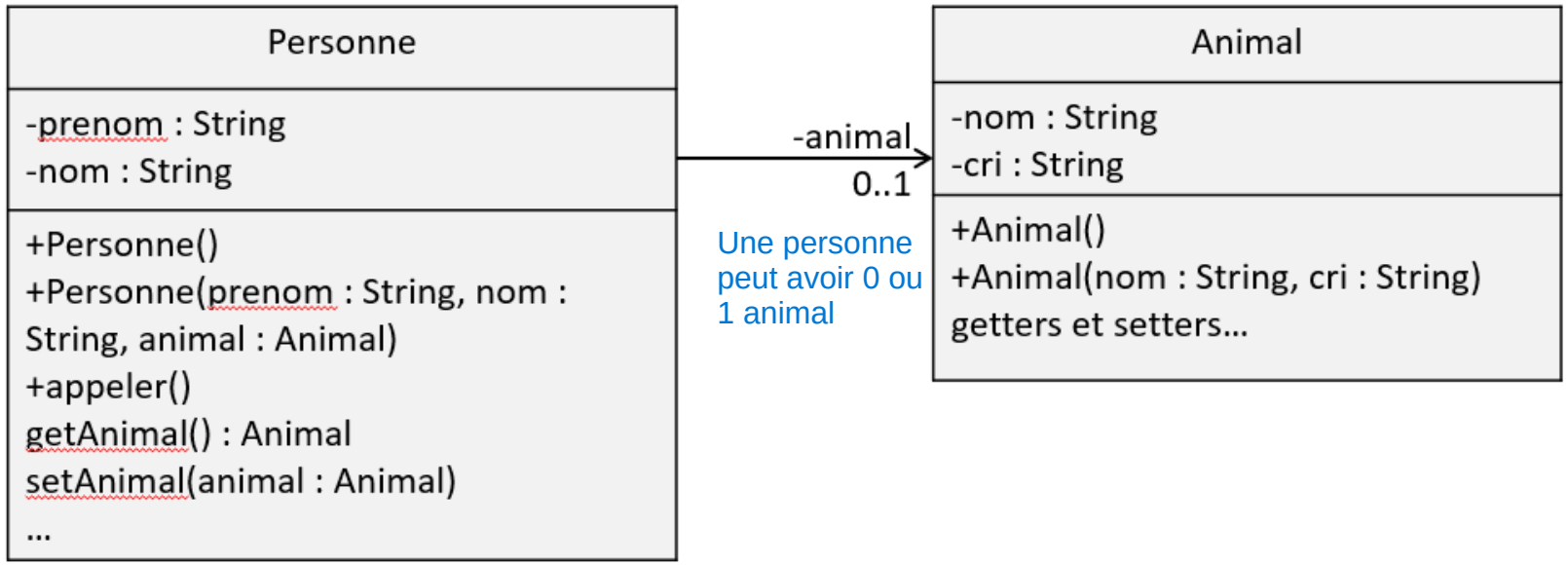
équivaut à



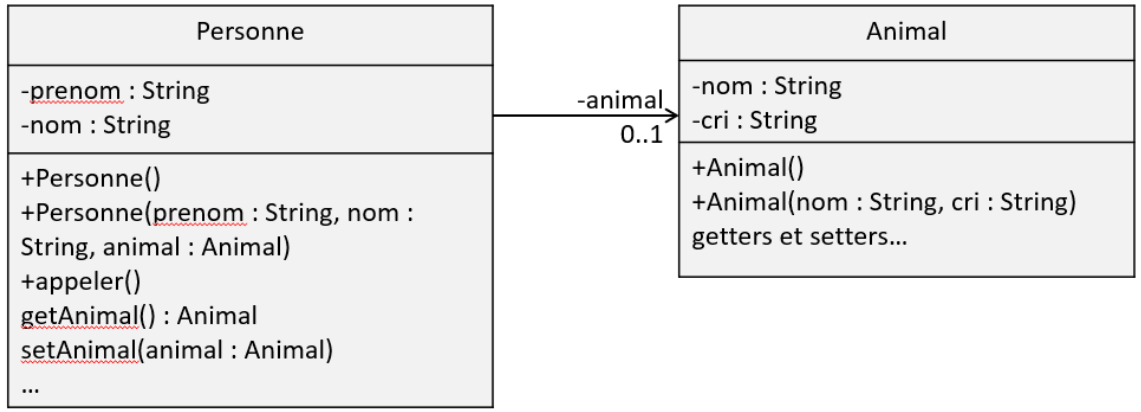
Les associations unidirectionnelles

```java
public class Personne {
    private String prenom;
    private String nom;
    private Animal animal;
```

```java
public class Animal {
    private String nom;
    private String cri;
```



Les associations unidirectionnelles




```java
public class Personne {
    private String prenom;
    private String nom;
    private Animal animal;

    public Personne() { }
    public Personne(String prenom, String nom, Animal animal) {
        this.prenom = prenom; this.nom = nom; this.animal = animal;
    }

    public Animal getAnimal() {
        return animal;
    }
    public void setAnimal(Animal animal) {
        this.animal = animal;
    }

    public void appeler() {
        System.out.println("Viens " + animal.getNom() + " !");
    }
```

```java
public class Animal {
    private String nom;
    private String cri;
```

accolades manquantes?

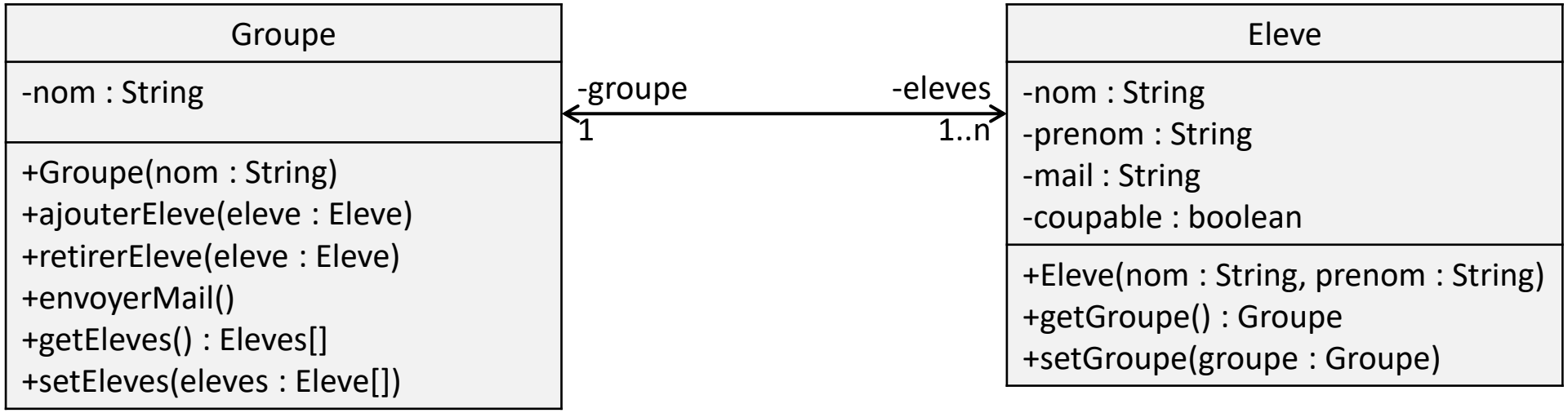
Les associations bidirectionnelles

- Une association bidirectionnelle est navigable dans les deux sens.
  - Exemple :
    - Un élève sait dans quel groupe il est
    - Un groupe connaît la liste de ses élèves

Les associations bidirectionnelles

- Une association bidirectionnelle est navigable dans les deux sens.
  - Exemple :
    - Un élève sait dans quel groupe il est
    - Un groupe connaît la liste de ses élèves

La variable eleve est généralement un tableau



Les associations bidirectionnelles

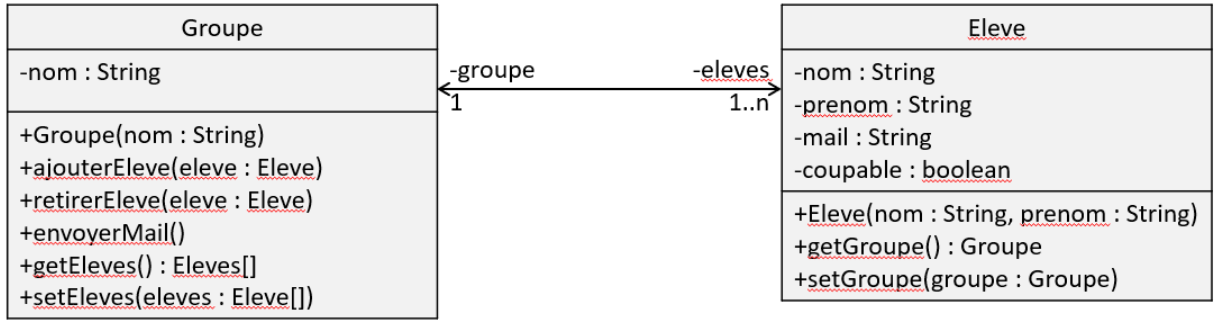
```java
public class Groupe {
    private String nom;
    private Eleve[] eleves;

    public Groupe(String nom, int nbEleves) {
        this.nom = nom;
        eleves = new Eleve[nbEleves];
    }

    public Eleve[] getEleves() {
        return eleves;
    }

    public void setEleves(Eleve[] eleves) {
        this.eleves = eleves;
    }

    public void envoyerMail() {
        for (Eleve current : eleves) {
            System.out.println(
                    current.getPrenom()
                    + " a bien recu le mail.");
        }
    }
```




```java
public class Eleve {
    private String nom;
    private String prenom;
    private Groupe groupe;
    private boolean coupable;

    public Groupe getGroupe() {
        return groupe;
    }
    public void setGroupe(Groupe groupe) {
        this.groupe = groupe;
    }
}
```

Les associations bidirectionnelles

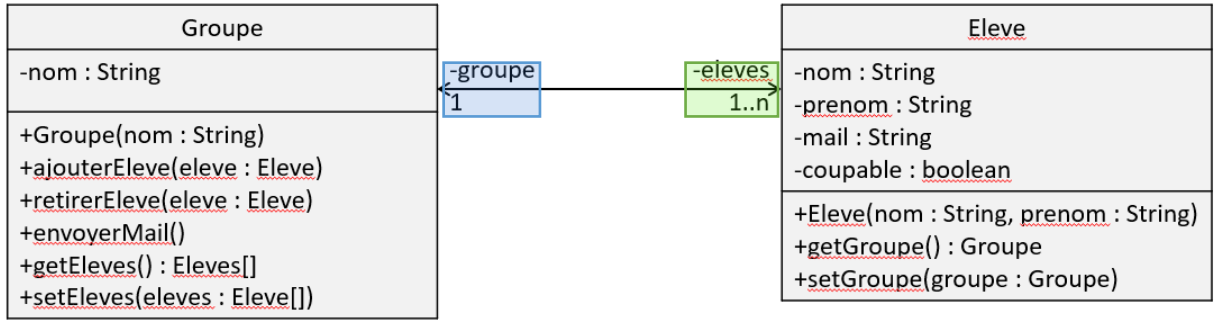
```java
public class Groupe {
    private String nom;
    private Eleve[] eleves;

    public Groupe(String nom, int nbEleves) {
        this.nom = nom;
        eleves = new Eleve[nbEleves];
    }

    public Eleve[] getEleves() {
        return eleves;
    }

    public void setEleves(Eleve[] eleves) {
        this.eleves = eleves;
    }

    public void envoyerMail() {
        for (Eleve current : eleves) {
            System.out.println(
                    current.getPrenom()
                    + " a bien recu le mail.");
        }
    }
```




```java
public class Eleve {
    private String nom;
    private String prenom;
    private Groupe groupe;
    private boolean coupable;

    public Groupe getGroupe() {
        return groupe;
    }
    public void setGroupe(Groupe groupe) {
        this.groupe = groupe;
    }
```

Atelier 4.1
Ajouter ou virer un élève ?

Faire:
-ajouterEleve
-retirerEleve

Démo
Risque d’incohérence

Atelier 4.2
Retour chez le médecin

Java
Associations