

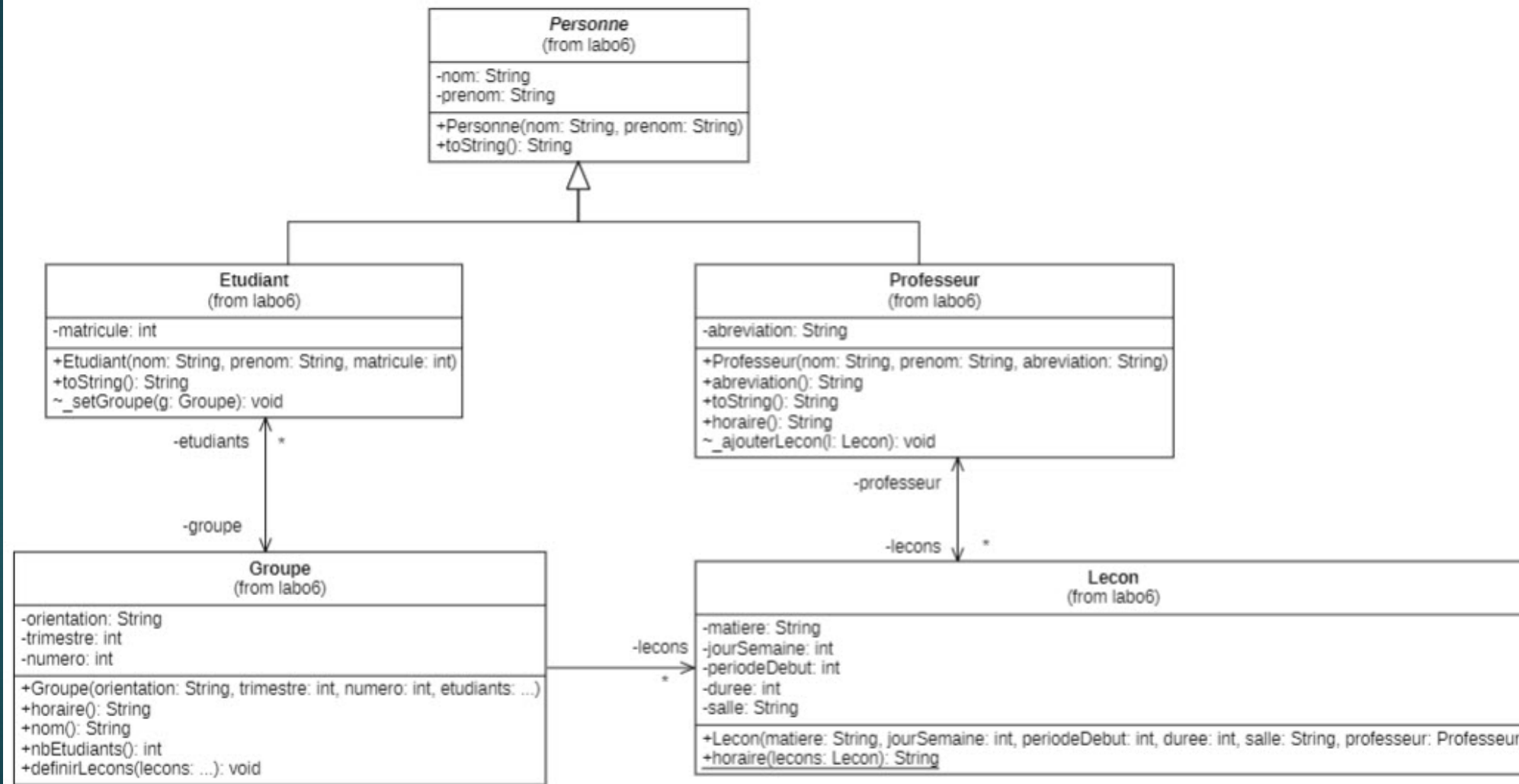
Feedback labo 6

NOVEMBRE 2022

R. RENTSCH

Diagramme de classes UML

2



Non respect de la donnée

3

► Vu

- nom prénom au lieu de prénom nom dans affichage Etudiant / Professeur
- groupe IS6-2 au lieu de SI6-1
- affichage des horaires sans en-tête
(par ex "Horaire du groupe IL6-1 (4 etudiants)")
- Dans classe Etudiant :
 - **public** void setGroupe(Groupe groupe)
 - **public** Groupe getGroupe()

Erreurs / maladresses diverses

4

- ▶ Classe Personne pas déclarée abstraite
(à faire quand bien même pas en italique sur le schéma de la donnée)
- ▶ throws RuntimeException ou IllegalArgumentException dans la signature de certaines méthodes
- ▶ Dans classe Etudiant
 - ▶ protected void setGroupe() au lieu de void setGroupe
 - ▶ Corollaire : Toute classe dérivée de Etudiant et externe au paquetage peut modifier le champ groupe
 - ▶ (pire) champ groupe déclaré avec visibilité paquetage et non private
 - ▶ Corollaire : Plus d'encapsulation

Erreurs / maladresses diverses

5

```
// classe Etudiant
```

```
// Pas bon
```

```
public String toString() {  
    return "Etud. " + super.toString() + " (#" + matricule + ") - " + groupe.nom();  
}
```

```
// Bon
```

```
public String toString() {  
    if (groupe == null)  
        return String.format("%s (%d)", super.toString(), matricule);  
    else  
        return String.format("%s (%d) - %s", super.toString(), matricule, groupe.nom());  
}
```

3 problèmes

- NullPointerException levée si groupe vaut null
- "Etud. "
- Non usage de String.format

Erreurs / maladresses diverses

6

```
// classe Groupe
```

```
// pas bon|
```

```
public Groupe(String orientation, int trimestre, int numero, ArrayList<Etudiant> etudiants) {...}
```

```
// bon
```

```
public Groupe(String orientation, int trimestre, int numero, Etudiant... etudiants) {...}
```

L'ellipse permet à l'utilisateur de passer en paramètre effectif :

- un objet Etudiant
 - un tableau d'Etudiant
 - une collection d'Etudiant
-
- ... donc beaucoup plus flexible !

Erreurs / maladresses diverses

7

// classe Groupe

// pas bon

```
private Etudiant[] etudiants;  
private Lecon[] lecons;
```

// bon

```
private List<Etudiant> etudiants;  
private List<Lecon> lecons;
```

// classe Groupe

// pas bon

```
public void definirLecons(Lecon... lecons) {  
    this.lecons = lecons;  
}
```

// bon

```
public void definirLecons(Lecon... lecons) {  
    Collections.addAll(this.lecons, lecons);  
}
```