

POO - Labo 6

- Groupe: L06GrJ
- Etudiants: Calum Quinn, Dylan Ramos
- Date de dernière modification: 05.11.2023

Choix d'implémentation

Classe `Etudiant`

Nous avons choisi d'ajouter un attribut privé `groupe` et une méthode publique `definirGroupe()` à la classe `Etudiant`.

En effet, pour pouvoir afficher l'étudiant lors d'un appel `toString()`, il faut savoir à quel groupe il appartient afin d'obtenir un affichage similaire à celui de l'énoncé.

Classe `Professeur`

Nous avons choisi d'ajouter un tableau de leçons en tant qu'attribut privé et une méthode publique `definirLecons()` à

la classe `Professeur`. En effet, pour pouvoir afficher l'horaire d'un professeur, il faut savoir à quelles leçons il participe.

Classe `Groupe`

Nous avons choisi d'ajouter un tableau d'étudiants en tant qu'attribut privé à la classe `Groupe` qui sera rempli à l'aide du constructeur lors de la création d'un groupe. Le constructeur vérifie qu'il y ait au moins 1 étudiant (pour respecter la cardinalité de la relation) et itère sur chaque étudiant pour leur assigner le groupe.

Enfin, comme pour la classe `Professeur`, nous avons ajouté un tableau de leçons en tant qu'attribut privé afin de stocker les leçons auxquelles le groupe participe. Cela s'avère utile pour l'affichage de l'horaire.

Classe `Lecon`

Etant donné que la méthode `horaire()` de la classe `Lecon` est statique, nous avons implémenté la méthode `horaire()`

des classes `Professeur` et `Groupe` de manière à ce qu'ils fassent appel à la méthode `horaire()` de la classe `Lecon`.

En effet, l'affichage de l'horaire a toujours le même format que ce soit pour un professeur ou un groupe, il est donc

plus adapté de factoriser le code à un seul endroit. De plus, nous stockons les leçons dans les classes

`Professeur`

et `Groupe`, ce qui nous permet de les passer en paramètre à la méthode `horaire()` de la classe `Lecon`.

Pour la création de l'horaire, nous avons choisi d'utiliser un tableau de `String` à deux dimensions afin de pouvoir créer les cellules colonne par colonne. En effet, créer les cellules de cette manière permet de simplifier le code lors

de la création d'une cellule de plusieurs périodes. Enfin, une fois le tableau rempli, nous concaténons les colonnes

afin d'obtenir l'affichage souhaité.