

## - TD 4 - Scolarité (suite)

On souhaite faire évoluer le logiciel pour la gestion des formations d'une université.

### 1. Charge financière

L'application sera également utilisée par différents services avec des objectifs potentiellement différents. Par exemple, la direction de la formation doit connaître la charge théorique (c'est-à-dire en supposant qu'il y a qu'un groupe CM/TD/TP pour le niveau concerné et ses sous-niveaux), le service financier souhaite connaître la charge réelle (c'est-à-dire en utilisant le nombre effectif de groupes CM/TD/TP). La charge réelle d'un module est calculée en général selon la formule:

$$\text{nb.h.CM} * \text{fact.CM} * \text{nb.g.CM} + \text{nb.h.TD} * \text{fact.TD} * \text{nb.g.T} + \text{nb.h.TP} * \text{fact.TP} * \text{nb.g.TP}$$

On considère que le nombre de groupes CM/TD/TP pour chaque UE est fourni par une méthode `int[] getNbGroupes(String module)` de la classe `Administration`; la même classe fournit des constantes correspondant aux facteurs CM/TD/TP. D'autres modes de calculs peuvent être facilement ajoutés par la suite; par exemple, une qui considère que tous les types d'heures ont la même pondération (i.e. considérer que tous les facteurs ont valeurs 1).

On doit avoir la possibilité de changer dynamiquement la manière de calculer la charge. Dans cette nouvelle version de l'application on doit pouvoir obtenir la charge pour un diplôme (ou collégium, discipline, etc.) soit explicitement au travers d'une méthode à prévoir, ou au travers de la présentation qui doit maintenant faire figurer cette information:

```
Collegium FST - Contact: SF, nombre UEs: 7
  Informatique - Contact: SF, nombre UEs: 7
    Master Info - Contact: HC, nombre UEs: 5
      Semester S7 (Contact: FL)
        Modules: [ACL, DP]
        Charge: 234
      Semester S8 (Contact: FL)
        Modules: [IA, IHM]
        Charge: 152
    ...
```

### 2. Information supplémentaires pour les modules

On souhaite ajouter aux modules une information supplémentaire - le code qui indique la thématique (27 pour info) et le niveau (M1, L2, etc.). La classe utilisée précédemment (qui est déclarée `final`) ne peut plus être modifiée (excepté pour ajouter des `implements` au niveau de sa déclaration) mais il faudra néanmoins la réutiliser car ses méthodes sont difficiles à (re)implémenter. Par exemple, quand l'information supplémentaire est ajoutée on obtient un affichage:

```
Semester S7 (Responsable: FL)
  Modules: [ACL (27-M1), DP (27-M1)]
  Nombre heures: 88
```