

DM n° 1 — Gestionnaire des bulletins d'un établissement

Antoine de ROQUEMAUREL (Groupe 3.1)

1 Avant-propos

Ce rapport contient les choix d'implémentation et de conception pour un projet dans le cadre de la L3 Informatique parcours Ingénierie des Systèmes Informatiques de l'université Toulouse III – Paul Sabatier. Il s'agit d'un logiciel permettant la gestion des bulletins d'un établissement scolaire.

Celui-ci doit être développé en Java en utilisant la bibliothèque graphique **Swing**, ceci à l'aide de l'éditeur WYSIWYG Matisse présent dans *Netbeans*. Ci-joint à ce rapport est donc présent les sources du logiciel sous forme de projet Netbeans.

2 Conception

L'application a été conçue en différents packages afin de faciliter le développement et la relecture du code. Cette répartition a été faite en deux packages principaux : les données dites « métiers », et l'interface graphique.

Ainsi, au besoin il est tout à fait possible de ne changer que l'interface de notre programme.

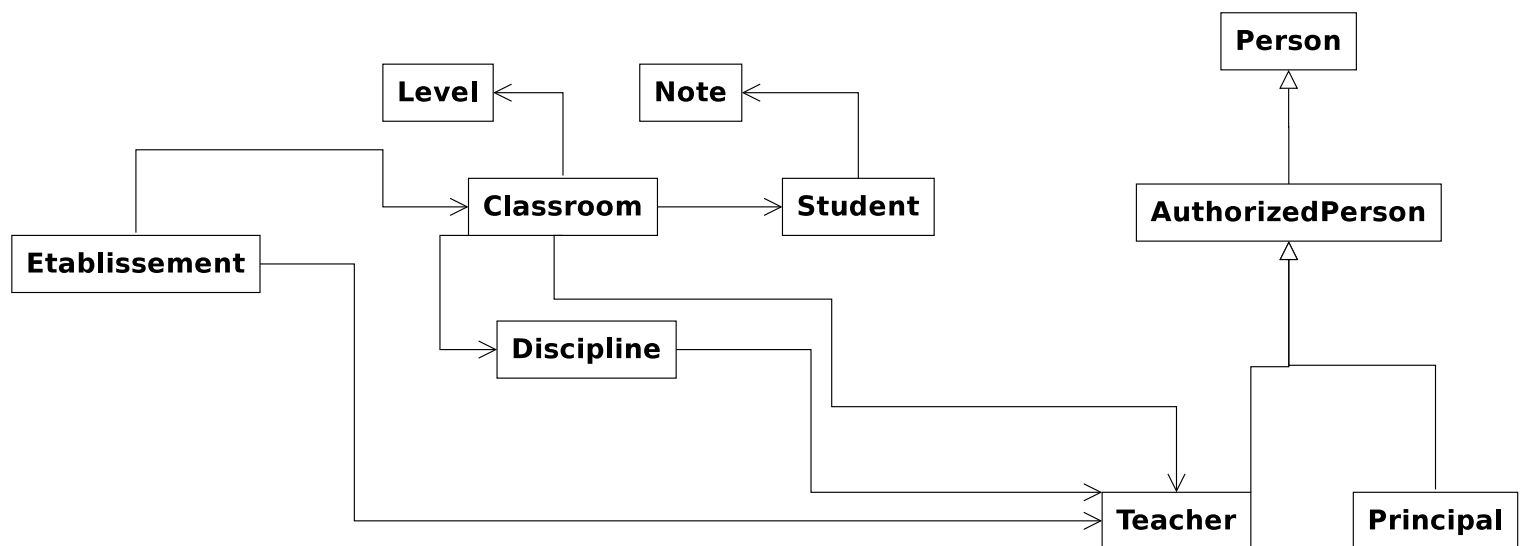


FIGURE 1 – Diagramme de classe des données métiers

2.1 Répartition en package

Notre application utilise un assez grand nombre de package afin de découper au mieux le problème, nous allons voir ici comment les fichiers sont organisés.

2.1.1 Données métiers

Les données de l'application, indépendamment de l'interface.

etablissement Les données concernant l'établissement.

etablissement.classroom Tout ce qui concerne une classe, les notes, le niveau, ...

etablissement.person Toutes les personnes

2.1.2 Interface Graphique Swing

Lancement de l'interface C'est le package racine, permettant de lancer la fenêtre principale et les principaux panneaux.

Gestion des données Ce package contient des sous-package permettant de gérer les données, c'est-à-dire, les vues, les modèles et les éditeurs.

gui.data.editors Classes editors

gui.data.models Classes modèles

gui.data.renderers Classes de vues

Images Celui-ci ne contient que les images permettant d'afficher des icônes dans les boutons.

Panneau du directeur Contient toutes les fenêtres et panneaux permettant l'utilisation du logiciel connecté en tant que principal.

gui.principal Panneau principal

gui.principal.classrooms Gestion des classes

gui.principal.teachers Gestion des professeurs

Panneau des professeur De la même manière que le package du principal, celui-ci gère les professeurs.

2.2 Connexion avec l'IHM

Afin de simplifier au maximum le problème, je suis partis du principe que l'interface Swing ne peut lancer qu'un seul jeu de données simultanément, ainsi la création de plusieurs instance de **MainFrame** pourrait empêcher le bon fonctionnement du programme.



Il aurait pu être intéressant de transformer la classe **MainFrame** en singleton, cela n'a pas été fait car cela ne fait pas l'objet du cours et n'est pas indispensable

La classe **MainFrame**, contient plusieurs attributs statiques, avec ces données, on peut obtenir toutes les informations métiers et les modifier très facilement.

L'attribut `_etablissement` contient toutes les données de l'application, comme le montre le diagramme de classe figure 1.

Le second attribut intéressant est `_currentPerson`, c'est la personne qui s'est connecté, ça permet d'afficher des informations différentes en fonction de l'utilisateur.

2.3 Utilisation des collection Java

Afin d'avoir un système le plus efficace est robuste possible, j'ai utilisé plusieurs types de collections, particulièrement pour aider à un trie efficace grâce au `TreeSet`, mais également grâce à une `Map` liant une matière à une note : celle-ci sont présente chez l'élève.

3 Choix d'interface

Le développement de l'interface s'est effectuée en respectant ce qui était demandé dans le sujet, ainsi peu de liberté ont été prise afin de satisfaire au maximum la demande.

Des contrôles ont été fait au niveau de l'insertion des notes : une note doit être entre 0 et 20, ce contrôle s'effectue à l'aide d'un *spinner*.

4 Données de tests

Afin de pouvoir tester le logiciel dans de bonnes conditions, des données de tests sont insérés au lancement du logiciel. Pour cela, il faut exécuter la méthode `main` présent dans le package `managementStudent`. Celle-ci va insérer automatiquement un proviseur, des professeurs, des classes avec leurs élèves associés ainsi que des notes : Toutes les données de lectures peuvent être essayés, il est ensuite possible d'ajouter et d'éditer des données depuis le logiciel.



Il n'était demandé aucune sérialisation ou stockage des données dans une base de données, ainsi une fois le logiciel fermée, toutes les données sont définitivement perdues. Le lancement du logiciel réinsèrera le jeu d'essais.

Il aurait été possible de stocker les données à l'aide d'une base *Sqlite* afin d'avoir la puissance et la souplesse des bases de données, ou une simple sérialisation de nos objets aurait pu faire l'affaire : ces deux solutions ont l'avantage d'être rapide à mettre en place.