

TP Java 10

Consignes

On rappelle ici les consignes des exercices :

- Les sections peuvent être effectuées indépendamment mais il est conseillé de les effectuer dans l'ordre
- La qualité du code (Structure, nommage, indentation, commentaires, ...) est prise en compte dans la notation
- Le code fourni doit être rendu avec la javadoc correspondante.
- Le rendu devra se faire sous la forme d'un lien git, Dropbox ou Drive à envoyer par mail.
- L'intégration d'outils de Linting (PMD) sera également prise en compte.
- Chaque section vaut pour 10 points

Contexte

On souhaite réaliser un programme de gestion des notes des élèves d'un collège. Le collège est organisé comme suit :

- 6 classes par niveau (6^e , 5^e , 4^e , 3^e) numérotées de A à F : ($6^e A$, $6^e B$, ...)
- 20 élèves par classe
- 9 matières en tout : mathématiques, français, anglais, histoire-géographie, physique, sciences naturelles, arts, musique, sport, langue vivante
- 3 matières optionnelles : latin, grec, anglais avancé
- 3 épreuves (3 notes) par matière

Partie 1 - Collecte des données

Concevez un programme java qui permette de générer 3 épreuves différentes (donc 3 notes) par matière et par élève de toutes les classes. Proposez une solution pour stocker ces notes dans un fichier xml ou json global. Vous intégrerez les contraintes suivantes :

- Les cours de langue vivante et de physique ne commencent qu'en 5^e
- Les cours facultatifs permettent d'augmenter la moyenne générale (à raison de 0.1 point supplémentaire par point au dessus de 10).
- Les cours de sport et de musique ne comportent que 2 notes
- Chaque élève possède, aléatoire, entre 0 et 2 matières facultatives
- Un élève ne peut étudier parallèlement latin et grec

Partie 2 - Traitement et regroupement

Modifiez votre programme pour y intégrer les fonctionnalités suivantes :

- Consulter l'ensemble des notes d'une classe
- Consulter l'ensemble des notes d'un élève
- Consulter l'ensemble des notes de toutes les classes d'un niveau
- Afficher les maximum, minimum, moyenne et médiane de chaque classe pour chaque note de chaque matière

Partie 3 - Affichage

Au moyen de graphes (histogrammes ou courbes), modifiez votre programme pour instaurer les fonctionnalités suivantes :

- Comparer les moyennes dans une matière entre toutes les classes d'un niveau (on donnera à l'utilisateur la possibilité de choisir la moyenne). On représentera les moyennes par classe sur un histogramme.
- Comparer la répartition des notes entre toutes les classes de chaque niveau (on dessinera la gaussienne pour chaque classe) pour une épreuve donnée
- Comparer la répartition des notes entre toutes les classes de chaque niveau pour la moyenne d'une matière
- Comparer la répartition des moyennes générales entre toutes les classes de chaque niveau

Partie 4 - Accès en ligne

On souhaite à présent permettre à l'utilisateur d'accéder à notre moteur de gestion de notes en réseau. On va donc séparer la partie affichage graphique de la partie moteur et leur permettre de communiquer via des sockets.

Rajoutez les fonctionnalités suivantes à votre outil : Lorsque l'utilisateur demande l'affichage des notes (graphes ou détail), celles-ci doivent être transmises via un json dans un socket. On vérifiera le fonctionnement en démarrant deux clients pour le même serveur.

Partie 5 - Import

A partir d'un fichier de notes au format txt formaté comme suit :

```
6B
HISTOIREGEO
NomA PrenomA 10.0
NomB PrenomB 11.0
NomC PrenomC 9.5
```

Faites en sorte de pouvoir rajouter des notes dans votre fichier json d'entrée en important le fichier transmis