

# Contrôle Final TIC 303 – UFE - 2014

---

## Durée

3 heures.

## Documents autorisés

Uniquement les supports distribués pendant le cours :

<http://richard.grin.free.fr/ufe/cours/tic303/supports/index.html>

**IMPORTANT** : vous devrez **bien séparer les questions** sur votre copie (n'écrivez pas le code de classes qui répondent à toutes les questions, sans que l'on puisse distinguer quelles parties des classes répondent à quelles questions) et donner vos réponses **dans l'ordre des questions**. Si vous ne savez pas répondre à une question vous devez laisser un espace vide non écrit pour cette question sur votre copie.

La note prendra en compte la présentation et la clarté du code. Si vous pensez que votre code n'est pas évident à comprendre, écrivez des commentaires.

Si vous le voulez, vous pouvez écrire le code avec du crayon à papier si le code est bien lisible (pas de crayon trop clair).

## Thème

Calcul des moyennes des étudiants.

## Présentation du sujet

Chaque étudiant peut avoir 2 types de notes qui sont prises en compte pour calculer une moyenne :

- une « note simple » ;
- une « moyenne » qui est calculée à partir d'autres notes.

Chacune des notes a un coefficient qui est utilisé pour le calcul de la moyenne.

### Exemple :

Un étudiant a une « note simple » qui a la valeur 15 en anglais. Cet étudiant a aussi une « moyenne » de programmation objet qui est composée de 2 notes simples : une note en C++ (valeur 12) qui a le coefficient 3 et une note en Java (valeur 10) qui a le coefficient 4.

La moyenne générale de l'étudiant est une « moyenne » calculée à partir de la moyenne de programmation objet (coefficient 4) et de la note d'anglais (coefficient 1).

Sa moyenne générale est calculée comme suit :

- calcul de la note de programmation objet (moyenne de C++ et de Java avec les coefficients 3 et 4 :  $(3 \times 12 + 4 \times 10) / 7$  ;
- calcul de la moyenne générale (moyenne de la note de programmation objet, qui est elle-même déjà une moyenne, et de l'anglais avec les coefficients 4 et 1) :  $(4 \times \text{MPO} + 1 \times 15) / 5$ .

MPO est la moyenne calculée pour la programmation objet. Sa moyenne sera donc environ 11,69 (résultat arrondi de  $(4 \times 76/7 + 15) / 5$ ).

Ce contrôle va vous amener à implémenter ce calcul de moyenne en vous faisant résoudre plusieurs questions. Si vous n'avez pas su résoudre une question, vous pouvez tout de même passer à la question suivante en supposant les questions précédentes résolues. N'oubliez pas de laisser un espace vide pour la question non résolue.

Toutes les classes que vous allez écrire seront dans le **paquetage eg.ufe.moyenne**, sauf s'il est dit le contraire dans la question.

## Partie 1 : description du calcul d'une moyenne

### Question 1 :

Vous allez commencer par écrire une classe toute simple, **NoteCoefficient**, qui sera utilisée par la classe que vous écrirez dans la question 2.

Cette classe représente un couple formé d'un identificateur du cours de la note (type String) et d'un coefficient (de type int). Une instance de cette classe permettra, par exemple, de représenter la note pour le cours « C++ », avec le coefficient 2. « C++ » est l'identificateur du cours de la note.

Cette classe a 2 variables d'instances **idCoursNote** et **coefficient**, un constructeur qui prend 2 valeurs en paramètre pour les affecter à ces variables d'instance et 2 getters pour ces 2 variables, une méthode toString(), et c'est tout.

### Question 2 :

La classe **DefinitionMoyenne** que vous allez écrire dans cette question, va servir à indiquer comment calculer une moyenne. Par exemple (le code que vous allez écrire doit se conformer à ces quelques lignes) les lignes suivantes indiquent que le calcul de la moyenne de « Programmation objet » se fait en prenant la note de C++ avec le coefficient 2 et la note de Java avec le coefficient 3 :

```
DefinitionMoyenne definitionPO =
    new DefinitionMoyenne("Programmation objet");
definitionPO.addNoteCoeff("C++", 2);
definitionPO.addNoteCoeff("Java", 3);
```

Cette classe va utiliser la classe NoteCoefficient de la question précédente. Plus précisément, elle contiendra

- Une variable d'instance de type String qui correspond à l'identificateur de la note/moyenne que vous définissez (« Programmation objet » dans les lignes de code données en exemple ci-dessus).
- Une variable d'instance de type List<NoteCoefficient> qui décrit comment il faut calculer la moyenne.
- Une méthode qui retourne un itérateur (interface java.util.Iterator étudié en cours) qui itère sur la List<NoteCoefficient> (dans une question suivante elle sera utilisée par une autre classe pour calculer la moyenne).

Une des valeurs qui sert pour le calcul d'une moyenne peut elle-même être une moyenne qui est calculée. Par exemple, la moyenne « Moyenne générale » peut être décrite ainsi (« Programmation objet » étant la moyenne décrite dans les lignes de code données en exemple ci-dessus) :

```
DefinitionMoyenne definitionMoyenneGenerale =  
    new DefinitionMoyenne("Moyenne générale");  
definitionMoyenneGenerale.addNoteCoeff("Programmation objet", 4);  
definitionMoyenneGenerale.addNoteCoeff("Anglais", 1);
```

### Question 3 :

Comme dans le TP sur les collections, vous allez ajouter dans la classe **DefinitionMoyenne** une variable de classe qui va conserver dans une **map** (de type `java.util.Map`) toutes les définitions des moyennes. Cette map va jouer le rôle d'une base de données qui contient les définitions des moyennes à calculer.

La clé des éléments de la map sera l'identificateur de la note (de type `String`) et la valeur sera de type `DefinitionMoyenne`. Comme dans le TP, les valeurs seront ajoutées à la map dans le constructeur de `DefinitionMoyenne`.

Comme dans le TP, vous ajouterez aussi une méthode **getter** pour demander à la classe `DefinitionMoyenne` de donner, par exemple, la définition de la note « Programmation objet ». Par exemple :

```
DefinitionMoyenne definition =  
    DefinitionMoyenne.getDefinitionMoyenne("Programmation objet");
```

Pour faciliter le travail du correcteur, écrivez le code demandé dans cette question 3 à part, pas à l'intérieur du code que vous avez déjà écrit pour la classe `DefinitionMoyenne`.

## Partie 2 : des étudiants ; les notes et les moyennes des étudiants

### Question 4 :

Écrivez la classe **Note** avec un constructeur qui prend en paramètre l'id du cours de la note (de type `String`) et sa valeur. L'id du cours de la note désigne la même chose que dans la question 1 ; il peut être égal à « C++ » ou « Java » par exemple. La valeur de la note est, par exemple, 12 ou 18,5.

La classe contient aussi un accesseur (`getter`) pour l'intitulé et la valeur de la note, et aucune autre méthode.

### Question 5 :

Vous allez écrire une classe **Etudiant**. Elle représente un étudiant et elle a une variable d'instance **nom** pour le nom de l'étudiant et une variable d'instance **notes** de type `java.util.Map` pour ses notes. Les clés de la map sont les identificateurs des cours des notes (par exemple « C++ ») et les valeurs sont les valeurs des notes (par exemple 10.5).

La classe contient un `getter` pour le nom de l'étudiant.

La classe contient une méthode **ajouterNote** pour ajouter une note à l'étudiant. Elle prend 2 paramètres : un identificateur du cours d'une note et un `double` (pour la valeur de la note ajoutée).

On pourra avoir, par exemple ce type de code :

```
Etudiant etudiant = new Etudiant("Bob");  
etudiant.ajouterNote("C++", 12);
```

### Question 6 :

Ajoutez à la classe **Etudiant** la méthode **getMoyenne**(String idCoursNote) qui calcule la moyenne de l'étudiant pour la note dont l'identificateur est passé en paramètre. On aura par exemple :

```
double moyennePO = etudiant.getMoyenne("Programmation objet");
```

Si l'identificateur du cours d'une note n'est pas dans la Map retournée par la méthode `DefinitionMoyenne.getDefinitionMoyenne` (de la question 3), elle est considérée comme une note simple qui ne nécessite pas de calcul, comme, par exemple, la note de Java dans les exemples donnés ci-dessus.

Aide : cette méthode `getMoyenne` sera récursive puisqu'une moyenne peut être calculée à partir d'autres moyennes, comme c'est le cas pour la moyenne générale définie en exemple dans la question 2.

Vous supposerez que l'étudiant a bien toutes les notes qu'il faut pour calculer la moyenne.

### Question 7 :

Vous allez reprendre la question précédente pour tenir compte du cas où un étudiant n'a pas une des notes nécessaires au calcul d'une moyenne.

Dans le cas où le calcul d'une moyenne utilise une note que n'a pas l'étudiant, votre méthode `getMoyenne` devra lancer une **NoteException** (**exception contrôlée par le compilateur**). Cette exception devra contenir les informations sur l'étudiant et la note manquante pour pouvoir les fournir à la méthode qui va l'attraper.

Ecrivez la classe `NoteException` et réécrivez la méthode `getMoyenne` pour tenir compte de cette exception.

La classe `NoteException` contiendra seulement un constructeur qui prend un paramètre l'id du cours d'une note et un étudiant, un getter pour l'étudiant et un getter pour l'id du cours de la note.

### Question 8 :

Ecrivez une classe **TestMoyenne** qui contient une méthode **main** pour tester les classes que vous avez écrites en affichant la moyenne générale d'un étudiant qui a les notes données dans l'exemple du début de cet énoncé (12 en C++ ; 10 en Java ; 15 en anglais). La moyenne générale sera calculée avec les coefficients qui sont donnés dans l'exemple du début de l'énoncé.

S'il manque une note à l'étudiant, un message d'erreur qui explique clairement le problème devra être affiché par la méthode. Par exemple, « Etudiant Bob n'a pas de note pour Java ».