PC2 – LA GESTION DES NOTES

MODULE TC131D : BASES DE LA PROGRAMMATION OBJET
UV TC131 : OUTILS DE L'INGÉNIEUR

SOMMAIRE

Prérequis 1

Approfondissements 1

Préambule 1

I. Sujet 2

II. identification des classes 2

III. exercices pas à pas 2

IV. les autres classes (facultatif) 3

PRÉREQUIS

Connaissances de base de la programmation objet et de Java étudiées dans le cours « Découverte de Java »

APPROFONDISSEMENTS

À la suite de ce TD, vous pouvez améliorer vos compétences en travaillant les programmes sur machine.

PRÉAMBULE

L'objectif de cette petite classe est de prendre en main les constructions simples de la programmation orientée objet en Java. En particulier, les exercices proposés abordent :

- L'identification des classes candidates à partir d'un énoncé ;
- La notion de classe comme un type ;
- L'appel et la définition de méthodes ;
- Le principe d'encapsulation ;
- La création d'objets.

CC BY NC SA

I. SUJET

Le travail à réaliser pendant cette petite classe concerne le développement d'une version simple d'un logiciel de gestion de notes d'élèves dans différents cours. Chaque cours a un intitulé et un coefficient réel. Les étudiants ont un nom, un prénom et un matricule qui leur est attribué automatiquement par le logiciel. Il correspond à un numéro séquentiel initialisé à 1 et qui s'incrémente à chaque ajout d'un nouvel étudiant. Les notes saisies doivent être comprises entre 0 et 20.

Les fonctionnalités offertes par le logiciel sont :

1. Saisir la liste des étudiants ;
2. Saisir, à partir du terminal, les notes (des flottants) de tous les étudiants d'un cours;
3. Calculer la moyenne des notes d'un cours et la rendre en résultat ;
4. Afficher, sur le terminal, les notes de tous les étudiants d'un cours ;
5. Afficher la liste des élèves.

Pour simplifier le problème, nous supposons, sauf consigne contraire, que tous les étudiants suivent un seul cours.

II. IDENTIFICATION DES CLASSES

À partir du sujet identifiez les classes candidates et leurs attributs. Spécifiez ensuite au moins les méthodes nécessaires pour assurer les 5 fonctionnalités du logiciel. Déterminez, enfin, dans quelle classe doit être codée chacune de ces méthodes.

III. EXERCICES PAS À PAS

Sachant que pour réaliser un logiciel il est possible de trouver plusieurs solutions correctes, dans cette suite d'exercices nous allons vous guider pas à pas pour aboutir à une solution facile à implémenter et respectant les principes de la programmation orientée objet.

A. LA GESTION DES NOTES

Nous supposons que la gestion de notes est réalisée par une seule classe `MarksManager` qui possède les méthodes suivantes :

- `public void addMarks()` : saisit, à partir du terminal , les notes (des flottants) de tous les élèves d'un cours ;
- `public float mean()` : calcule la moyenne des notes d'un cours et la rend en résultat
- `public void showMarks()` : affiche, sur le terminal, les notes de tous les élèves d'un cours ;
- `public MarksManager(Course course, int numberStudents)` : crée une instance de la classe. Le cours et son nombre d'élèves sont passés en paramètres.

Donnez le code de la classe `MarksManagerUse` qui utilise les méthodes de la classe `MarksManager` pour gérer les notes de 5 élèves en physique et 7 élèves en mathématiques.

```
public class MarksManagerUse {
      public static void main(String[] args) {
```

BY NC SA

```
            // Create manager for marks
            // ...

            // Add new marks
            // ...

            // Show marks
            // ...

            // Compute the mean
            // ...

      }
}
```

Donnez le code de la classe `MarksManager`. Dans cette classe les notes seront stockées dans un tableau de flottants.

B. LE PRINCIPE DE L'ENCAPSULATION

Il est décidé de placer les deux classes précédentes dans un même paquetage pc2.

Modifiez vos classes pour satisfaire cette contrainte.

Quel serait le résultat de la compilation du code suivant de la classe `MarksManagerUse` :

```
MarksManager pManager = new MarksManager("physique", 5);

// Add new marks

pManager.addMarks();

// Modifying marks by direct access to marks on manager

pManager.marks[3] = 14;

pManager.marks[1] = 25;

pManager.marks[19] = 17;
```

Proposez une solution au problème soulevé.

Ajoutez une méthode à la classe `MarksManager` qui permette la modification de la note d'un élève. La description de la méthode est la suivante :

`public void modifyMark(int numero, float mark)`, où numero est le numéro de l'élève et mark indique la nouvelle note.

IV. LES AUTRES CLASSES (FACULTATIF)

Cette partie facultative en PC permet d'ancrer les acquis sur la création de classes et l'application du principe d'encapsulation. Elle sera néanmoins traitée en TP2.

CC BY NC SA

A. LES COURS

Donnez le code la classe Cours en veillant à définir le bon niveau d'encapsulation des attributs et des méthodes.

B. LES ÉTUDIANTS

Donnez le code la classe étudiant en veillant à définir le bon niveau d'encapsulation des attributs et des méthodes.

Donnez le code de la classe `StudentManager` permettant de gérer dans un tableau la liste des étudiants saisie par l'utilisateur.

CC BY NC SA