Assurance qualité logiciel

Projet final: Gestion des notes des étudiants

Le but de ce projet est de créer des classes permettant de représenter des étudiants (classe Student), cours(Classe Course) et des notes (Classe Grade).
Vous allez sans doute découvrir un certain nombre de nouveaux outils durant ce projet. Le temps d'apprentissage de ces outils peut être au début frustrant mais ils vont vous faire gagner du temps par la suite.

Tâche 1 : classe étudient:

Chaque étudiant est identifié par:

-Numéro d'étudiant
- Nom
-Prénom

Tâche 2 : classe cours:

Chaque étudiant est identifié par:

-Numéro du cours
-Code
-Titre

Tâche 3 : classe notes:

Chaque étudiant est identifié par:

-Identifient
-Numéro d'étudiant
-Numéro du cours
-Note

Travail demandé

Il vous est demandé de mettre en place une application permettant de créer, et afficher les étudiants, les cours et les notes, ainsi que permettre de saisir les notes des étudiants. L'application doit permettre d'enregistrer les données dans un fichier texte par étudiant et

d'afficher le relevé de notes en temps réel lorsque l'utilisateur entrera le numéro de l'étudiant.

Quoi utiliser ?

Il est recommandé :

- d'utiliser C# pour faciliter le développement;
- Application Console

Par contre, vous pouvez utiliser n'importe quelle langue de programmation.

Comment valider son projet

Les étudiants sont appelés à travailler avec Git et Github. Ils doivent s'appuyer sur les notions vues en classes mais aussi des nombreuses ressources disponibles en ligne afin de créer leur propre branche vers laquelle pousser leur travail personnel. Cela signifie que chaque étudiant aura sa propre branche, contenant son travail.

Le travail à rendre doit comporter tous les fichiers nécessaires à la prise en main du projet, essentiellement le code source

Standards de programmation

Identificateurs

Les identificateurs doivent être aussi descriptifs que possible et conforment aux conventions suivantes :

- Les noms de classes et d'interfaces doivent débuter par une lettre majuscule ;
- Les noms de méthodes et de variables débutent par une minuscule, sauf si la méthode est un constructeur ;

- Pour les identificateurs composés de plusieurs mots n'utilisez pas le caractère souligné (underscore), faites débuter le second mot et ceux qui suivent par une majuscule;
- Les noms de constantes, c'est-à-dire les variables déclarées final, sont en majuscules et utilisent le caractère souligné comme séparateur de mots;
- Le nom d'une méthode devrait débuter par un verbe, par exemple, calculeNote et non note;
- Une méthode dont la seule fonction est d'accéder aux variables d'instances (attributs) devrait comporter le nom de la variable, par exemple, getAttribute, setAttribute et isAttributeEqualTo(value) ;
- (optionnel) Lorsqu'un paramètre formel réfère à une variable d'instance, le nom du paramètre et de la variable devraient être identique, ce qui signifie que vous devez utiliser la notation this.variable pour les différencier, par exemple,

```
public class Entier {

        private int valeur;

    public Entier( int valeur ) {
        this.valeur = valeur;
        }
}
```

Variables

- Ne déclarez que les variables qui sont utilisées ;
- Évitez les allocations de mémoire inutiles ;
- N'utilisez pas la même variable pour des usages différents ;
- Ne confondez pas les variables d'instance et les variables locales; déclarez chacune à l'endroit approprié ;
- Chaque variable devrait avoir un bref commentaire descriptif. C'est ce que l'on appelle le dictionnaire des données, il devrait être placé au début de chaque méthode, avant ou à côté des déclarations.

Structure des programmes

- La structure de vos programmes est très importante. Introduisez vos propres interfaces et décomposer vos algorithmes en sous algorithmes afin d'en améliorer la clarté.
- Chaque méthode, incluant la méthode principale, devrait être précédée d'un commentaire décrivant son but ainsi que l'algorithme utilisé (sauf si ce dernier est très simple). Soyez succinct ;

Entrées/Sorties

- De même, lors de l'entrée de données au clavier, votre programme doit imprimer un incitatif (*prompt*) décrivant à l'usager ce qu'il ou elle doit être entré au clavier.
- Votre programme doit tester la validité des données saisies et gérer les exceptions.
- Votre programme doit générer un fichier texte pour chaque étudiant, ce fichier doit contenir toutes les informations sur l'étudiant.
- Votre programme doit imprimer le contenu du fichier lorsque le numéro de l’étudiant est saisi.

Corrections des programmes

La correction des programmes est faite sur la base de la validité du code, de la qualité (qui inclut la clarté) et la gestion des erreurs. Chacune des parties de votre solution, par exemple chaque méthode, est évaluée individuellement par rapport aux conditions et exigences de l'énoncé. Dans la majorité des cas, le projet comporte un programme principal (méthode main) visant à trouver un grand nombre d'erreurs logiques.