



PROGRAMMATION ORIENTÉE OBJET ET INTERFACES

ESILV S6 ~ 2017-2018

「 TD1 - Conception Objet 」

Objectif(s) du TD : Concevoir un programme de gestion des cours dans une université fictive

Le contexte :

- Dans une université sont dispensés des cours.
- Ces cours sont suivis par les étudiants qui s'y inscrivent eux-mêmes.
- Chaque cours peut nécessiter en pré-requis qu'un autre cours ait été suivi et réussi, (note>10)
- Les cours sont dispensés par des professeurs.
- L'affectation d'un professeur à un cours est effectuée par la secrétaire pédagogique (la matérialisation physique de l'université).
- Pour obtenir son diplôme, un étudiant doit suivre le parcours pédagogique (l'ensemble des cours) associé à ce diplôme ; et avoir la moyenne à chacun de ces cours.

Pour des raisons pédagogiques le contexte fixé est volontairement simplifié.

On souhaite développer cette application en programmation objet.

Les questions qui se posent :

1. Par où commencer ?
2. Quels sont les objets (ou classes) ?
3. Quelles informations doivent-ils détenir ?
4. Quelles méthodes internes ?
5. Comment les programmer ?

Des éléments de réponse :

1. détermination des objets et/ou des classes :
 - réflexion et analyse du contexte et de l'utilisation prévue du programme
2. pour l'analyse :
 - les cartes Class-Responsability-Collaborators (CRC)
Elles permettront de déterminer les informations et les actions dont chaque classe est responsable.
3. la programmation :

- Programmation de chaque classe (champs et méthodes en C#) en C# avec Visual Studio
- L'interface utilisateur, qui dans une application réelle serait constitué d'objets tirés des WPF (Windows Presentation Foundation), sera dans ce TD remplacé par un IHM composé d'un menu en mode console permettant d'établir le dialogue entre l'humain et les objets (L'utilisation des WPF sera étudiée ultérieurement dans ce cours)

Remarque :

Lors de la phase de programmation, le Main sera utilisé pour instancier quelques premiers objets et permettre une interactivité avec l'utilisateur sans avoir à créer un par un chacun des objets participant à cette application.

Les cas d'utilisation :

Pour terminer de spécifier l'application, et déterminer les rôles de chacune des classes participant à l'application, il est nécessaire de définir les besoins auxquels elle (l'application) doit répondre et la manière dont elle sera utilisée.

Ces sont les cas d'utilisation :

- un étudiant s'inscrit à un cours (si son inscription est possible)
- un étudiant consulte sa note à un cours
- un étudiant obtient la liste de ses notes à un parcours et valide (ou non) le diplôme correspondant
- l'administration affecte un professeur à un cours
- un professeur consulte la liste des horaires de ses cours
- un étudiant valide (ou non) un diplôme (note supérieure à la moyenne pour chacun des cours du diplôme)
- l'administration enregistre la note d'un étudiant à un cours
- le professeur consulte la moyenne du cours (pour l'ensemble des étudiants inscrits à ce cours)

La spécification de l'application (et la liste des cas d'utilisation correspondante) est bien entendue (ici aussi) simplifiée par rapport à une situation réelle pour des raisons pédagogiques.

Le travail à effectuer :

Le travail demandé s'effectuera en groupe de 6 étudiants

1. Identifier les classes
2. Spécifier les propriétés et les méthodes de ces classes en utilisant les cartes CRC
3. Coder chacune des classes en C#
4. Utiliser le Main fourni pour disposer d'une d'interface utilisateur (le menu) et de l'instanciation des premiers objets
Vérifiez et créez dans vos classes des constructeurs conformes aux lignes d'instanciation initiale d'objets du Main fourni.

Première partie :

Pour cette partie : travail individuel

- Identifier les classes :
Elles correspondent à des ”choses” du monde réel existant en multiples exemplaires dont chacune est identifiable individuellement.
- Constituez la liste des classes que vous avez identifiées à partir de la description du contexte et de la liste des cas d'utilisation.

Deuxième partie :

Pour cette partie : travail collectif en groupe de 6 étudiants

A chaque étudiant est attribué le rôle de l'une des classes et donc la carte CRC correspondant à cette classe. Chaque étudiant concerné se construit alors une carte CRC sur le modèle fourni ci dessous (dimensions 1/4 de feuille A4)

<i>ClassName</i>	<i>Collaborators</i>
<i>Responsabilities</i>

Vous procéderez ensuite à un jeu de rôle animé par l'étudiant qui n'a pas de carte CRC à compléter. Pour chaque cas d'utilisation l'animateur organisera les échanges (”messages” au sens de la POO) entre les divers acteurs concernés

Chaque acteur notera sur sa carte, ce que sa classe est sensée savoir (la ou les données sous sa responsabilité) ou savoir faire (les méthodes), ainsi que les liens de collaboration qu'elle établit avec d'autres acteurs.

Cette activité peut conduire à prendre en compte des cas d'utilisation non prévus dans la description initiale.

Exemple :

L'étudiant 'françois' veut connaitre sa note en 'info fondamentale' .

Pour cela l'objet étudiant 'françois' doit envoyer un message demandant sa note à l'objet cours 'info fondamentale' lui demandant la note de l'étudiant 'françois'.

L'objet cours (et donc la classe correspondante) a donc sous sa responsabilité la liste des notes de chacun des étudiants pour l'instance interrogée de la classe.

Il s'agit d'une propriété (la liste des note des étudiants) commune à chaque instance de la classe (chaque cours). Le responsable de la carte CRC note donc sur sa carte côté responsabilité de sa classe : liste des notes des étudiants.

De plus les messages 'demander la note' et 'renvoyer la note' doivent correspondre à une méthode dans l'une des classes etudiant ou cours. Choix que vous aurez à faire par un débat entre vous.

En plaçant côte à côte deux cartes qui collaborent l'une avec l'autre, il sera aisé de comprendre la collaboration entre les deux cartes et de vérifier le bon fonctionnement de cette collaboration.

De plus, et c'est le rôle de l'animateur, vous devez vous poser toutes les questions connexes. Cet échange entre l'objet étudiant 'françois' et l'objet cours 'info fondamentale' est-il possible directement ? Ou bien

une phase préliminaire est-elle nécessaire ? Comment active t'on l'objet étudiant 'françois' ? Par l'interface utilisateur ... Oui mais pour savoir qu'il y a un objet étudiant 'françois' il faut connaître la liste des objets étudiant, afficher cette liste pour choisir 'françois', l'activer et lui faire envoyer un message... Et qui connaît la liste de étudiants ?... De même quels sont les cours suivis par 'françois' ? qui la connaît ? qui peut en fournir la liste qui permettra d'identifier le cours désiré ?

Dans ce jeu de rôle chacun déterminera quelles informations sont sous sa responsabilité (ses champs ou attributs) et à quels messages (ses méthodes) il peut être amené à répondre.

L'animateur veillera à ce que tous les cas d'utilisation aient été explorés et que les classes ainsi constituées permettent de répondre à l'utilisation décrite par les cas d'utilisation (y compris avec les besoins intermédiaires identifiés cf l'exemple ci-dessus).

A la fin de cet exercice vous disposerez donc d'une spécification des champs et méthodes de chacune des classes de votre application.

COURS	
Nom	Etudiant
Numéro	Professeur
Liste d'attente	...
Étudiants inscrits	
Enseignant	
Ajouter des étudiants	
Supprimer des étudiants	
...	

La phase suivante sera le codage et les tests unitaires

Si votre travail a été fait de manière complète, vous disposerez des éléments constituant votre programme, prêts à être assemblés (en les complétant avec le Main fournit pour l'interface utilisateur)

La dernière phase sera l'intégration des classes et du Main ainsi que les test d'intégration.

Troisième partie :

Pour cette partie : vous travaillerez individuellement dans votre groupe de 6 étudiants.

Vous préciserez la spécification complète des méthodes de votre carte (paramètres formel et type de retour).

Vous communiquerez aux classes avec lesquelles vous collaborez les signatures de ces méthodes afin qu'elles puissent y faire appel. Puis chacun code la classe dont il détient la carte CRC.

L'animateur veille à la cohérence des classes, des appels de méthodes et des constructeurs. Et il procède à l'intégration des classes (après leur codage) avec le main fourni (contenant le menu et les instantiations initiales.

A la fin cette partie, vous disposez d'un application de gestion des inscriptions aux cours d'une université...