

TP1 : Prise en main de l'atelier de génie logiciel

Le but de ce TP est double :

- découverte des fonctionnalités de l'atelier de génie logiciel,
- modélisation des premiers exemples vus en TD.

Découverte des outils et choix d'un éditeur UML

Vous trouverez sur G:/ (windows) ou /share (linux) une présentation de l'IDE (Integrated Development Environment) Eclipse, du gestionnaire de version SVN, de l'outil de documentation Javadoc et de l'éditeur UML MagicDraw.

Un éditeur UML vous permet de réaliser les diagrammes à rendre dans les TP et de générer le code correspondant. MagicDraw est déjà installé sur les machines de l'ISTIC et est relativement simple d'utilisation. Il existe d'autres éditeurs, dont certains qui peuvent intégrer à Eclipse. Le choix de celui que vous utiliserez vous est laissé libre. Vous pouvez aussi utiliser un éditeur UML Web, tel que <https://www.genmymodel.com/>.

Exercice 1 : Associations

À partir du TD sur les associations (TD 3), réalisez les diagrammes de classes afin de mettre en œuvre les différentes associations décrites ci-dessous :

- 1 - Un étudiant peut suivre un cours et un cours a au maximum un étudiant.
- 2 - Un étudiant peut suivre un cours et un cours peut accueillir tous les étudiants inscrits.
- 3 - Un étudiant peut suivre autant de cours qu'il le souhaite et un cours peut accueillir tous les étudiants inscrits.

Créez un projet Java `fr.istic.m1.aco.tp1` dans Eclipse. À partir des diagrammes de classe réalisés, générez et corrigez si nécessaire le code correspondant dans votre projet. Ce code doit mettre en œuvre les différentes possibilités d'association. Il assurera notamment l'intégrité référentielle de ces associations en prenant en compte la possibilité de modification par une classe extérieure.

Exercice 2 : Débug

Créez un `main` qui manipule des cours et des étudiants correspondant à l'association demandée en 1.3. Placez un point d'arrêt dans le `main` avant l'appel à une méthode et lancez l'exécution en mode débbug. Consultez la valeur des différents objets dans la vue Variables. Exécutez le reste du code en mode pas à pas.

Exercice 3 : Javadoc

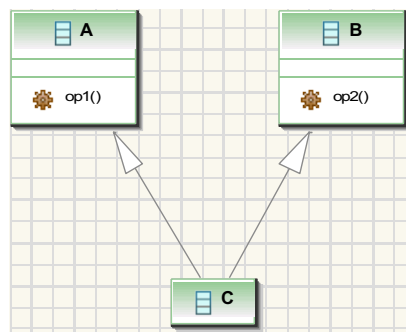
Commentez le code de vos classes en ajoutant le nom des auteurs et en commentant les attributs correspondants aux associations. Générez la Javadoc à partir de ces commentaires.

Exercice 4 : SVN

Il y a un dépôt SVN dédié aux TP d'ACO sur la forge de l'ISTIC à l'adresse : <http://forge.istic.univ-rennes1.fr/projects/tp1aco2017/>. Pour y accéder vous devez d'abord vous rendre sur la forge de l'ISTIC (<http://forge.istic.univ-rennes1.fr/>), vous connecter avec votre identifiant sésame et votre mot de passe et demander à rejoindre le projet M1 TP 1 ACO 2017-2018 (accessible par la recherche). Vous pouvez ensuite ajouter l'adresse du dépôt dans la vue SVN Repositories de votre Eclipse. Créez dans le *trunk* de ce dépôt un répertoire à vos noms et un sous-répertoire tp1. Commitez votre projet dans le répertoire tp1.

À méditer

Pour prendre conscience de la difficulté du *mapping* entre UML et Java, vous pouvez réfléchir à vos heures perdues à la mise en oeuvre de l'héritage multiple (autorisé en UML) en Java.



Bon TP