

Génie logiciel avancé

Travail pratique #1

Modélisation et programmation orientée-objet avec Java

Professeur : HO Tuong Vinh
Date énoncée: 3/8/2017
Date de remise : 19/8/2017, avant 17h30

Un petit gestionnaire de tâches

Ce travail a pour but de vous rappeler les concepts de la modélisation avec UML et programmation **orientée-objet** avec Java. De plus, il vous fait familiariser à un environnement de développement intégré (IDE) « libre » (Open Source) ECLIPSE. Il vous est demandé aussi d'utiliser GitHub (github.com) pour la gestion de source code de votre projet.

Vous avez à réaliser un programme en **Java** et à rédiger un rapport qui décrit votre travail.

1. PROGRAMMATION

1.1 Spécification de l'application

Vous devez réaliser un petit gestionnaire de tâches pour une équipe de travail. Ce gestionnaire fournit à l'utilisateur les fonctionnalités suivantes:

1. Créer, modifier, supprimer, ajouter une tâche
2. Créer, modifier, supprimer, ajouter un membre
3. Assigner une tâche à un membre
4. Chercher et afficher tous les tâches assignées à un membre (par son ID)
5. Chercher et afficher tous les tâches en fonction de leur status (avec le nom du assigné)

Une tâche est composée de ces informations suivantes :

- ID
- Nom
- Une description
- Status : nouveau, en-progrès, terminé.

Un membre est composé de ces informations suivantes :

- ID
- Nom.

1.2 Contraintes de programmation

- Il faut programmer de façon orientée objet
- Il faut assurer que la communication entre l'utilisateur et le système est conviviale.
- Les consignes de codage doivent être respectées.

2. RAPPORT

Le rapport doit aborder les points suivants :

- Une introduction de votre travail
- Les exigences fonctionnelles (utiliser le diagramme Use Case de l'UML) et non-fonctionnelles de l'application
- La conception (utiliser le diagramme de classe et de séquence)
- L'implémentation (utiliser Junit pour le test unitaire)
- Le test d'acceptation
- Conclusion (auto-évaluation de votre travail)
- Code source (en annexe).

3. EVALUATION

Le programme sera évalué sur 65% et le rapport sur 35%. Un programme qui plante à l'exécution, même dans une situation extrême respectant la spécification haute, se verra attribuer zéro sur 65%.