

OCL partie III

Exercices

1. Nous allons tout d'abord nous familiariser avec l'éditeur UML Papyrus.
 - (a) Sauvegardez si nécessaire vos projets actuels et rechargez le projet Eclipse original pour tous partir du même point.
 - (b) Installer Papyrus, si nécessaire (Help -> Install modeling component -> Papyrus (incubation)).
 - (c) En double-cliquant sur `model` directement sous `ProjetPapyrus` on devrait pouvoir ouvrir l'éditeur graphique.
 - (d) Explorer le modèle en cliquant sur les classes et associations et en observant les propriétés dans le panneau du bas.
2. Ajouter dans la classe `Employe` les opérations suivantes à l'aide de la Palette à droite et du panneau de propriétés en bas.
 - (a) L'opération `contribueA` prenant un argument `d` de type `Document`.
 - (b) L'opération `creeDocument` ne prenant aucun argument mais retournant une valeur de type `Document`.
 - (c) L'opération `ajouterTache` qui prend un argument `t` de type `Tache`.
3. Ajouter dans la classe `Document` les opérations suivantes qui seront des *requêtes* (Query), donc qui ne changeront pas l'état du système.
 - (a) L'opération `premierAuteur` qui ne prend aucun argument et retourne un `Employe`.
 - (b) L'opération `dernierAuteur` qui ne prend aucun argument et retourne un `Employe`.
4. Ouvrez maintenant la vue hiérarchique pour vérifier que les opérations sont correctes et bien présentes.
5. Ajoutez les pré et post-conditions suivantes au fichier `.ocl`. N'oubliez pas de faire charger le modèle UML par Dresden OCL!
 - (a) Dans le `context` `Employe::contribueA(d : Document)`
 - i. Après l'appel `'d'` est dans les documents auxquels l'Employé a contribué.
 - ii. Après l'appel l'Employé est le dernier contributeur à `'d'`
 - iii. Après l'appel `'d'` a un contributeur de plus qu'avant l'appel.
 - (b) Dans le `context` `Employe::creeDocument() : Document`
 - i. Le `Document` retourné n'a qu'un seul contributeur qui est l'Employé courant.
 - ii. Après l'appel le `Document` retourné est dans les documents auxquels l'Employé a contribué.
 - iii. Le `Document` retourné a été créé par l'opération (c'est donc un nouvel objet).

(c) Dans le `context Employe::ajouterTache(t : Tache )`

- i. Avant l'appel, il faut que 't' ne soit pas déjà une Tâche du PlanDeTravail de l'Employé.
- ii. Avant l'appel, il faut que le PlanDeTravail et ses Taches soient définis (non nuls).
- iii. Après l'appel 't' est dans les Taches du PlanDeTravail de l'Employé.
- iv. Après l'appel il y a une Tache de plus dans le PlanDeTravail de l'Employé qu'avant l'appel.

6. Définissez le corps (body) des requêtes (query) suivantes.

- (a) Dans le `context Document::premierAuteur() : Employe`, la valeur de retour est le premier contributeur.
- (b) Dans le `context Document::dernierAuteur()`, la valeur de retour est le dernier contributeur.