

Réusinage de modèles UML

Une bonne pratique de développement est d'alterner les ajouts de fonctionnalités avec des étapes de réusinage (refactoring). Les réusinages se sont tout d'abord appliqués au code : un réusinage consiste à modifier le code pour le rendre plus lisible, plus facile à comprendre et à modifier, sans changer ses fonctionnalités. Au sein d'une approche IDM, on effectue également des réusinages, mais au niveau des modèles. Nous allons écrire ici quelques méthodes de réusinage de modèle UML.

1. Écrire une méthode de réusinage qui permet de déplacer une classe d'un package à un autre. Pour implémenter facilement cette méthode, on supposera pouvoir récupérer une classe et un package à partir de leur nom (ce qui dans le cas général est faux, il faut disposer du nom qualifié). La méthode de refactoring prendra donc juste en paramètre le nom d'une classe, et le nom du package cible. Si le package n'existe pas, le réusinage échoue.
2. Écrire une méthode de réusinage qui remplace un attribut public par un attribut privé et une paire d'accesseurs (en lecture et en écriture). Cette méthode prend en paramètre le nom de l'attribut et le nom de la classe qui le contient. On supposera toujours que l'on peut récupérer une classe à partir de son nom.
3. Écrire une méthode de réusinage qui fait "remonter" une méthode dans une super-classe. Cette méthode prend en paramètre le nom de la classe d'origine, le nom de la super classe, et le nom de la méthode à faire remonter. Si le nom de la superclasse ne correspond pas à une super-classe de la classe origine, si le nom de la méthode donné ne correspond pas à une méthode de la classe d'origine, ou si une méthode concrète du même nom que la méthode à faire remonter existe déjà dans la super-classe spécifiée, alors le réusinage échoue.
4. Écrire une méthode de réusinage qui remplace un attribut de type simple par un attribut objet. Si une classe A possède un attribut x de type T, on créera une nouvelle classe X qui possèdera un attribut x de type T, et la classe A possèdera désormais un attribut x de type X. Cette méthode prendra en paramètre le nom de la classe où se trouve l'attribut à réusiner, et le nom de l'attribut à réusiner.

NB Pour pouvoir définir des attributs de type simple dans un modèle UML, il est nécessaire de définir d'abord un nouveau DataType qui porte le nom du type simple que vous voulez utiliser (String, Integer, ...).

Pour ceux qui ont fini en avance :

- utiliser KUnit
- utiliser des exceptions pour les erreurs