Compte rendu

Project de CPOA

Vincent Bourdoncle, Yannick Mayeur

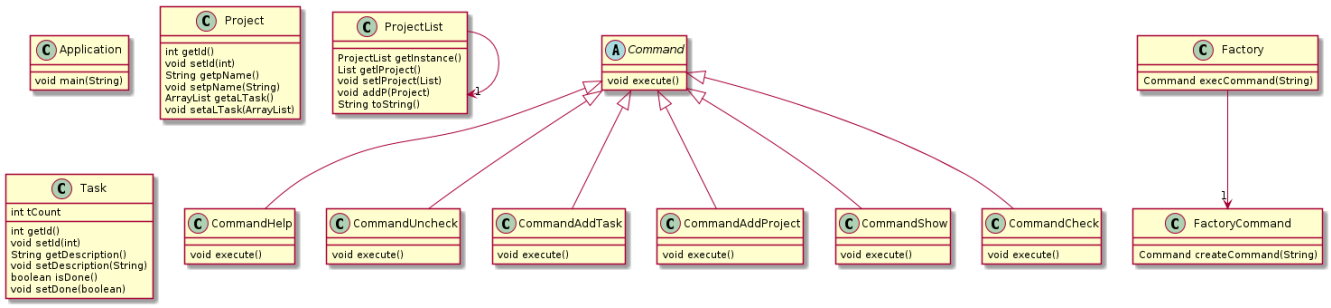
Sommaire

1. Introduction

2. Conception orienté objet

Le java étant un language orienté objet nous avons pour le refactoring naturellement utilisé une approche objet. Cette approche permet entre autre une maintenance plus facile du code, mais surtout la grande facilitation d'ajout de nouvelle fonctionnalité.

Voici le diagramme de classe générer par plantUML après refactoring :



Nous avons fais le choix de creer plusieurs nouvel classe.
Une classe Application d'on le seul but est de créer la liste de projet et de recevoir les entrées claviers.
Plusieurs classes modèle. Une pour les tâches, les projets et une pour la liste de projet. Nous avons fait l'hypothèse qu'il n'existe qu'une seule liste de project, et avons donc implémenter le design pattern *Singleton*. Parmi les classes modèles nous avons également les commandes. Pour gérer celle-ci nous avons implémenter le design patten *Factory*. Ainsi quand l'application lit l'entrée clavier la factory crée un nouvelle Commande qui aura son éxécution.

3. Ajout de fonctionnalité

3.1. Fonctionnalité de supression

Du fait que nous avons une *Factory* pour les commandes en rajouter une nouvelle est très simple, nous avons donc pu en rajoutant seulement une classe héritant de commande créer cette nouvelle fonctionnalité.

4. Conclusion

Le refactoring de cette application aura était une éxperience très enrichissante. D'une part nous avons pu voir l'intérêt de respecter les bonnes pratiques objets, avec un ajout de fonctionnalités bien plus simple. Et d'autres parts nous avons pu mettre en oeuvre certaine de nos nouvelles connaissances acquiéri au cours des cours de CPOA.