Projet Design Pattern

Ingénieur 2000 – IR Deuxième année
Rémi Forax, Tibault Crucy & Dominique Revuz
forax@univ-mlv.fr, tcrucy@etudiant.univ-mlv.fr, dr@univ-mlv.fr

Le projet

Le but de ce projet est de faire évoluer un code source (celui de **GanttProject**) vers une nouvelle version de Java, ici le passage de la version 1.4 vers la version 1.5.
Comme le projet **GanttProject** est faiblement documenté (et c'est un euphémisme), ce projet se découpe en deux parties, la première consistant a écrire une documentation développeur du projet, la seconde à modifier le code source pour que celui-ci soit compatible avec la version 1.5 de Java.

Source de GanttProject

Les sources de GanttProject sont disponibles à cet URL :
http://www-igm.univ-mlv.fr/~forax/ens/design-pattern/ir06-07/projet/ganttproject-src.zip

Création de la documentation développeur

Le but de cette partie consiste a écrire une documentation développeur de 20 pages environ présentant le code du logiciel **GanttProjet** à quelqu'un souhaitant modifier celui-ci.
Ce document devra contenir les informations suivantes :

1. Une introduction sur le pourquoi du logiciel
2. Une présentation générale de chaque paquetage Java du logiciel :
   Une introduction expliquant rapidement l'ordre dans lequel les paquetages seront décris.
   Un diagramme UML des dépendances entre paquetages
   (Il y a une dépendance lorsque une classe d'un paquetage à besoin d'une classe d'un autre paquetage)
   Pour chaque paquetage, une description d'au moins 5 lignes sur la responsabilité de celui-ci.
3. Puis, pour chaque paquetage,
   - une introduction expliquant l'ordre dans lequel les classes vont être décrites
   - un diagramme de classes (toujours en UML) avec les relations d'héritage et les associations entre classes. Chaque classe du diagramme devra faire figurer ses champs et ses méthodes (signature complète SVP).
   - Pour chaque classe ou groupement de classe si certaines partagent des détails d'implantations, une description de leur rôle et responsabilisée.

4. Pour finir une conclusion d'une demi-pages sur l'architecture de **GanttProject**

Généricification de GanttProject

Le but de cette seconde partie consiste à généricifier **GanttProject**, c'est-à-dire à remplacer les « *raw* » type par des types paramétrés.
Pour expliquer la généricification de **GanttProject**, vous devez produire un document d'une quinzaine de pages environ contenant les informations suivantes :

1. Une introduction expliquant l'organisation du document ainsi que l'intérêt pour un

projet de générifier son code.
2. Une explication de pourquoi le code de **GanttProject** ne compile pas avec le JDK 1.5 Ainsi que les corrections qui doit être effectuer pour que celui-ci compile et fonctionne.
3. Un récapitulatif sur deux pages des différentes techniques que vous avez utilisés pour vous assurez que la générification que vous avez effectué est correct.
4. Puis par paquetage, et par classe, une description en quelques phrases de la façon dont on doit générifier la classe.
   Attention à ne pas trop générifier, le code doit rester fonctionnelle ou à oublier l'utilisation des wilcards.

Recommandation Importante :
En aucun cas vous ne devez toucher à une autre partie du code que celle utilisant des generics (même si il y a un gros bug), de plus ne devez pas modifier l'indentation du code existant.

Outils pour l'analyse de code

Eclipse
Eclipse peut être paramétré pour lever un warning si un code utilise un raw type, ce qui vous indiquera en partie ce qu'il faut changer. De plus, eclipse possède une fonctionnalité de générification automatique (qui des fois marche).

FindBug
C'est un outils d'analyse statique du code (il existe un même un plugin eclipse) qui permet de détecter de possibles bugs sur un code. Il peut être paramétré pour détecter certain types de bugs.

Cette liste n''est pas limitative, il existe d'autres outils et d'autres IDE aidant à la générifaction. Vous êtes libres de les utiliser.
De plus, je vous conseil la lecture de cette article :
http://java.sun.com/docs/books/tutorial/extra/generics/

Détail des rendus

Le travail à effectuer doit être fait par binôme (2 personnes).

Le rendu de ce projet consiste en deux documents au format PDF (**dev.pdf** et **generification.pdf**) ainsi qu'un ZIP contenant le code de votre **GanttProject** générifié. Le tout devra être envoyé dans une archive ZIP par mail à l'ensemble des personnes indiqués en début de sujet avec dans le sujet de votre mail "[IR2] Design-pattern GanttProject binôme %s %s", les %s correspondant au nom des deux apprentis du binôme.

Date limite du rendu : 23h59m59s le 18 Février 2007