Programmation Objet et UML

TP - le modèle de conception "Observateur/Observable"

(inspiré d'un sujet de TP de G. Casiez)

Le Modèle de conception "Observateur/Observable" a été introduit en 1994 dans le célèbre livre du "Gang of Four" (Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides) : *Design Patterns - Catalogue de modèles de conceptions réutilisables* (titre français). Ce livre a été le premier à formaliser la notion de "Modèle de Conception" ("Design Pattern" en anglais).

Ce modèle de conception est largement répandu et est aussi utilisé dans certains méta-modèles de conception tel que le complexe "MVC" (Modèle-vue-contrôleur). La Fig. 1 donne son diagramme de classes tel qu'on le trouve dans Java 1.6. Il s'agit d'une classe Observable qui modélise un objet (appelons-le "objet central") qui est observé par plusieurs objets observateurs, implémentant l'interface Observer. Un observateur est "abonné" auprès de l'objet central via la méthode addObserver (il peut être "désabonné" via la méthode removeObserver). Lorsque l'objet central change, on peut appeler sa méthode notifyObservers et tous les observateurs sont alors notifiés : leur méthode update est appelée. On peut faire passer un message lors de cet appel via l'argument de type Object dans notifyObservers, ou bien l'observateur peut aller lui-même chercher les informations dont il a besoin. La méthode setChanged permet de marquer que l'objet central a changé, fait qui peut être consulté via la méthode hasChanged.

Ce fonctionnement simule une diffusion d'informations d'un émetteur vers de nombreux récepteurs ("broadcast") et a l'avantage d'être souple (on peut "s'abonner" et se "désabonner" en une instruction) et permet la gestion d'observateurs très nombreux.

Typiquement, ce modèle peut être utilisé dans une interface graphique, où divers composants graphiques affichent simultanément des informations liées à des données qui évoluent dans le temps.

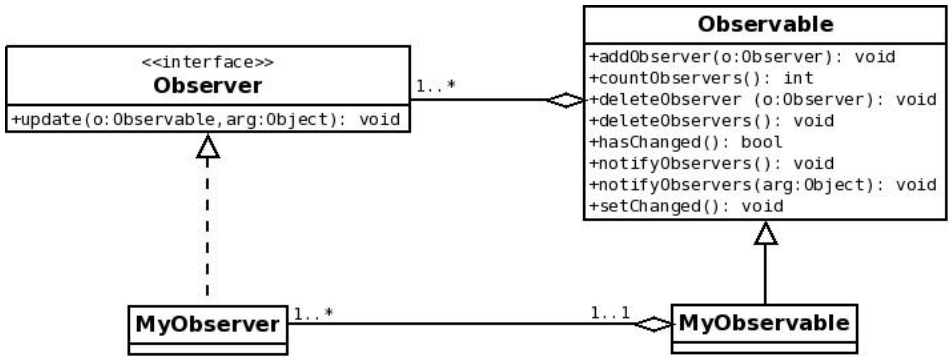

FIG. 1 – Diagramme de Classes du modèle Observateur/Observable (les noms des méthodes sont ceux de Java 1.6)

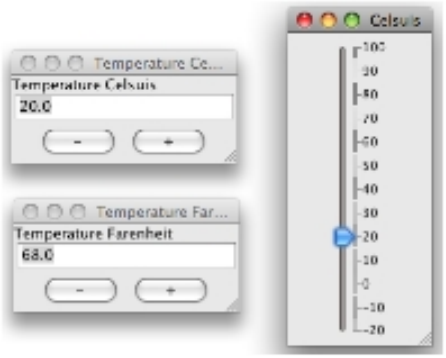

FIG. 2 – Illustration d'une implémentation possible de l'interface graphique demandée

Travail d'implémentation

L'objectif du TP est de créer une interface graphique avec la librairie Java Swing permettant le contrôle de la température en degrés Celsius ou Farenheit. L'interface graphique se compose de trois "vues" représentant la même température sous des formes différentes (voir Fig. 2). La modification d'une des vues doit mettre automatiquement à jour les autres vues. Les trois vues doivent implémenter l'interface Observer, et la valeur du thermomètre est contenue dans une classe Observable.

Les champs de texte peuvent être des JTextField contenus dans un JComponent et la barre de défilement peut être un JSlider.

Vous pouvez (devez !) visiter l'URL `http://download.oracle.com/javase/6/docs/api/` (ou tapez "java api 6" dans votre moteur de recherche favori). Il s'agit de l'API de java 1.6, très pratique pour les programmeurs Java. Vous y trouverez les spécifications des classes dont vous avez besoin.