Design patterns et architectures logicielles

Étude de cas : le design pattern *Observer*

Filière : SUD-Cloud & IoT

Travail réalisé par :

- Chrif El Asri Hanane
- El Gamous Khalid

Encadrant:

Prof. Abdeslam EN-NOUAARY

2020/2021

Introduction

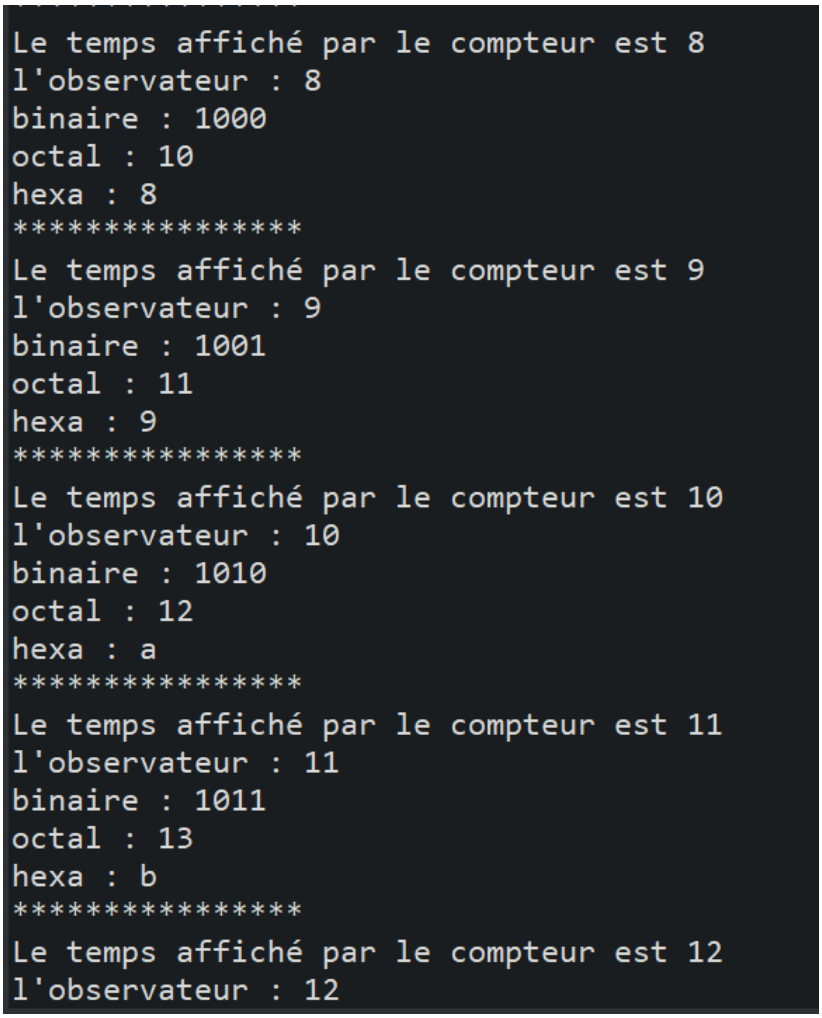
L'**Observateur** est un patron de conception comportemental qui permet de mettre en place un mécanisme de souscription pour envoyer des notifications à plusieurs objets, au sujet d'événements concernant les objets qu'ils observent. Lors de cette étude de cas on va comprendre comment il fonctionne.

I. Etude de cas

Dans cette étude de cas on veut réaliser un compteur cyclique modulo 60 (genre d'une minuterie) à utiliser par exemple dans un contrôleur de feu de circulation. Chaque fois que le compteur change de valeur celle-ci est affichée sur des dispositifs en binaire, en octal, et en hexadécimal.

Pour cela, on va utiliser le design pattern **Observal** pour que les classes *binaire*, *octal* et *hexadécimal* soient notifiées simultanément par la classe **Observateur**. L'observé dans cette étude de cas est le compteur qui se présente comme une valeur entière. Chacune des trois classes peut s'inscrire et se désinscrire auprès de l'observé pour afficher la valeur du compteur dans la base correspondante.

Pour implémenter cette solution, on va utiliser une API JAVA ; la classe ***Observable*** et l'interface ***Observer***.



Les trois classes peuvent s'abonner à l'aide de la fonction **addObserver** prédéfinie par la classe **Observable** et peuvent se désabonner à l'aide de la fonction **deleteObserver**

Conclusion

On constate qu'il est préférable d'utiliser le patron de conception Observateur quand des modifications de l'état d'un objet peuvent en impacter d'autres, et que l'ensemble des objets n'est pas connu à l'avance ou qu'il change dynamiquement. Et quand certains objets de votre application doivent en suivre d'autres, mais seulement pendant un certain temps ou dans des cas spécifiques.

NB : Vous pouvez trouver le code source dans ce lien :

www.github.com/hanane-ca/Design-Patterns.git