

TD 3 : Framework Spring IoC + AOP

Préparation :

Téléchargez le code source d'exemple du framework Spring fourni, avec les fichiers .jar nécessaires pour faire fonctionner le programme. Pensez à créer une librairie sous Eclipse pour mieux organiser les fichiers .jar. (Project build path → Add library → User library → new → Add jars → OK.)

Revue des notions importantes :

1) Parcourez rapidement les classes (sous-classes) de « **Person** » et celles de « **Address** » pour comprendre leur organisation (design patterns *Template Method* et *Bridge*) et le simple service fourni (sayHello). Rappel : ces classes sont les JavaBean.

2) Le mécanisme de l'IoC (*Inversion of Control*) est assuré par Spring pour réaliser *Dependency Injection*. Analysez le fichier spring.xml pour comprendre comment les beans sont définis et organisés entre eux, ainsi que l'utilisation de ces beans dans le programme TestSpringIoC. Maîtriser la différence entre les deux modes de scope : singleton et prototype.

3) Analysez la partie de configuration AOP (*Aspect Oriented Programming*), avec la classe PersonLogger, pour comprendre les notions suivantes : *Aspect*, *Pointcut*, *Join point* et *Advisor*. Comprenez surtout comment le mécanisme d'AOP permet de réaliser le journal sans modifier le programme existant.

Version 3 de Simulation Bank :

1) Reprenez la version précédente (version 2) du programme Bank. Prenez également le fichier [spring.xml](#) fourni par l'enseignant. Mettez ce fichier dans le package « simulation ». Commencez par analyser ce fichier. D'après le fichier, quelles sont les classes que l'on devra transformer en des beans gérés par Spring ? Modifiez déjà la structure de ces classes pour qu'elles respectent la convention de JavaBean et la configuration définie dans le fichier spring.xml.

2) Faites attention surtout au constructeur de la classe « Simulation ». Vous avez besoin d'y réfléchir pour voir comment réaliser la construction de la « Bank » dans une autre méthode.

2) Vous chercherez dans le programme les endroits où il faut réaliser du refactoring :

- ✓ La création des clients avec ses opérations
- ✓ La création de l'objet Simulation avec les entrées et le StatisticManager
-

3) Vous ajouterez dans le programme une petite fonctionnalité permettant d'écrire dans un fichier nommé « **ProblemReport** » les informations concernant **tous** les clients qui sont partis sans être servis. Ce fichier sera lu par le directeur de la banque et a un format comme suit (ex. pour un client) :

Dear boss, today, there is another client who has left without being served !
Here is some information about the upset client :
Arrival time : 540
Departure time : 547
.....