

## Adapter, Façade, Proxy

Le fichier compressé qui accompagne ce sujet contient :

- un exemple de mise en œuvre d'un adapter par délégation (exemple 1) qui est le modèle le plus courant et un exemple de mise en œuvre d'un par classification (exemple 2)
- un exemple de mise en œuvre d'un proxy (exemple 3)
- le code source de l'exercice n°1 (exercice1)
- le code source de l'exercice n°2 (exercice2) (vu dans le TD2)
- le code source de l'exercice n°3 (exercice3) (vu dans le TD2)

**Etape préliminaire : étudier les 3 exemples de façon à bien comprendre le principe et l'architecture de l'adapter et du proxy.**

- 1 ► Faire le diagramme de classes, et le diagramme de séquences correspondant à l'exécution du programme principal, pour les exemples 1 et 2.
- 2 ► (exercice1) : faire le diagramme de classes de l'exemple, le modifier pour y introduire un adapter de façon à encapsuler la classe CSudokuGridSaver afin de standardiser l'usage sur le modèle de l'interface existante (GridSaver).
- 3 ► (exercice2) : dans le package facade, on trouve un cas d'usage (cf TD2) de Home Cinéma. L'objectif est de créer une façade, appelée HomeCinema, qui encapsule les différentes classes et qui montre à l'utilisateur une méthode « show(titre) » qui correspond à la configuration par défaut décrite dans le programme principal. Faire le diagramme de classes intégrant la façade, le diagramme de séquences du programme avec la façade, et réécrire ce programme en utilisant la façade. Le scénario consiste à jouer le titre, puis à augmenter, après lancement du visionnage, le volume à 70 %.
- 4 ► (exercice3) : selon l'énoncé du TD2 à propos du proxy, écrire la classe ProxyWebFilter et mettre à jour le programme principal en conséquence. On utilisera le WebServer comme un strategy du proxy.