

LE PATTERN PROXY

Source « <https://www.math.univ-paris13.fr/~chaussar/> »

Vous allez développer un petit programme qui permet de télécharger, sur son mobile, les pochettes des CDs les plus demandées du marché !

Exercice 1

Lancez le `main`, vous devriez voir apparaître une fenêtre avec un menu, permettant de sélectionner l'album à afficher. Essayez d'afficher un album, et attendez... jusqu'à l'affichage de la pochette.

Exercice 2

Vous l'avez compris l'application n'est pas très bien codée. En effet, les utilisateurs ne sont pas toujours patients. Or, du temps est nécessaire pour télécharger la couverture de l'album choisi. Si l'on clique plusieurs fois sur le bouton d'affichage, beaucoup de fenêtres apparaissent.

Pour éviter ça, il faudrait modifier un peu de code mais problème : les classes `Main`, `MainWindow` et `GetAlbumAction` sont considérées comme déjà téléchargées par le client (on ne peut plus y toucher, à moins de demander au client de faire une mise à jour du programme, chose que l'on évitera de lui proposer) et la classe `AlbumViewer` est sur un serveur extérieur, que l'on ne maîtrise pas.

Vous n'avez le droit que de rajouter des classes, ou modifier `FactoryAlbumExporter`. L'idée serait d'afficher au client une image d'attente (une image est présente dans le dossier `img`) en attendant qu'il récupère son image.

Pour ce faire, voici quelques indices :

- Vous devrez créer une classe qui sera notre proxy : elle implémentera `AlbumExporter` se faisant passer auprès du client pour un `AlbumExporter`. La fonction `displayAlbum()`, appelée par le client pour afficher la pochette, pourra ainsi être modifiée.
- Le proxy devra tout de même appeler certaines fonctions de `AlbumViewer` : il faut simplement ajouter des fonctionnalités (affichage d'une image d'attente) à cette fonction.
- Dans `AlbumViewer`, vous avez des fonctions pour récupérer l'image, et non

l'afficher.

- Vous devrez peut-être modifier aussi `FactoryAlbumExporter`.
- Dans `MyImage`, la fonction `replaceWith()` permet de modifier une image déjà affichée à l'écran.

Exercice 3

Testez votre application : lorsqu'on clique sur le bouton de sélection de pochette d'albums, la fenêtre n'affiche pas le contenu de l'image d'attente. Pour corriger cela, inspirez-vous de `AlbumViewer` pour résoudre ce problème (utilisez des threads).

Exercice 4

Parfait, le programme fonctionne bien, mais il y a encore un problème : que se passe-t-il si la personne appuie plusieurs fois sur le bouton pour afficher les albums ? Et oui, il va toujours s'afficher des dizaines de couvertures d'album. Modifiez votre proxy afin qu'il bloque la création en cascade de couvertures d'albums.