



DESIGN PATTERN

TP Les structurels

TABLE DES MATIERES

Exercice sur Adapteur	2
Exercice 1	2
Exercice sur Composite.....	2
Exercice 2	2
Exercice 3	2
Exercice sur Façade.....	2
Exercice 4	2
Exercice 5	2
Exercice sur Proxy	3
Exercice 6	3

EXERCICE SUR ADAPTEUR

EXERCICE 1

Imaginez que vous ayez une classe `Forme` avec une méthode `calculateArea()` qui calcule l'aire d'un carré. Créez un adaptateur (`Adapter`) pour cette classe afin de pouvoir utiliser la méthode `calculateArea()` avec une classe existante `Rectangle` qui a déjà une méthode similaire.

EXERCICE SUR COMPOSITE

EXERCICE 2

Le but de cet exercice est de créer une arborescence contenant l'organisation d'un service RH d'une entreprise et notamment l'organisation d'un projet :

- Un manager
- Une équipe de concepteurs
- Une équipe de développeurs

Un manager peut avoir sous ses ordres un autre manager ou une équipe

Créer la structure de ces objets sous la forme du Design Pattern Objet composite :

- Liste des classes
- Schéma UML
- Code dans un langage informatique

EXERCICE 3

Créez une hiérarchie d'objets graphiques (cercle, carré, triangle) avec une interface commune. Ensuite, utilisez le motif composite pour créer une structure arborescente de ces objets graphiques et définir des opérations qui peuvent être appliquées à toute la structure.

EXERCICE SUR FAÇADE

EXERCICE 4

Soit un sous-système permettant de lire et d'enregistrer des fichiers physiquement sur un disque dur.

Le format d'entrée est une chaîne classique.

Le format à l'intérieur du fichier est du XML

Vous devrez créer les classes permettant de simuler ces fonctionnalités

EXERCICE 5

Soit une application permettant de gérer les salariés d'une entreprise.

Lors de la saisie d'une fiche d'un salarié, vous devrez enregistrer une copie de son contrat de travail.

Il faudra donc vous connecter au sous-système

Créer les classes nécessaires :

- Via un diagramme UML
- Puis en code
-

EXERCICE SUR PROXY

EXERCICE 6

- 1) Créer une interface `FileManager` qui permet les méthodes suivantes :
 - Lecture dans un fichier
 - Ecriture dans un fichier
- 2) Implémenter cette interface via une classe `RealFilemanager`
- 3) Créer une interface `FileManagerProxy` qui ajoute une méthode de vérification d'accès
- 4) Implémenter l'interface selon le design pattern Proxy

Créer le diagramme de classes ainsi que les code correspondant