

Exercice1 Adapter

On a développé un programme qui sait utiliser une interface qui offre des services pour gérer les carrés.

Seulement, nous ne disposons qu'une classe qui manipule les rectangles : définir les paramètres du rectangle, calculer son périmètre et son aire.

On souhaite utiliser cette classe sans changer l'interface que connaît notre programme.

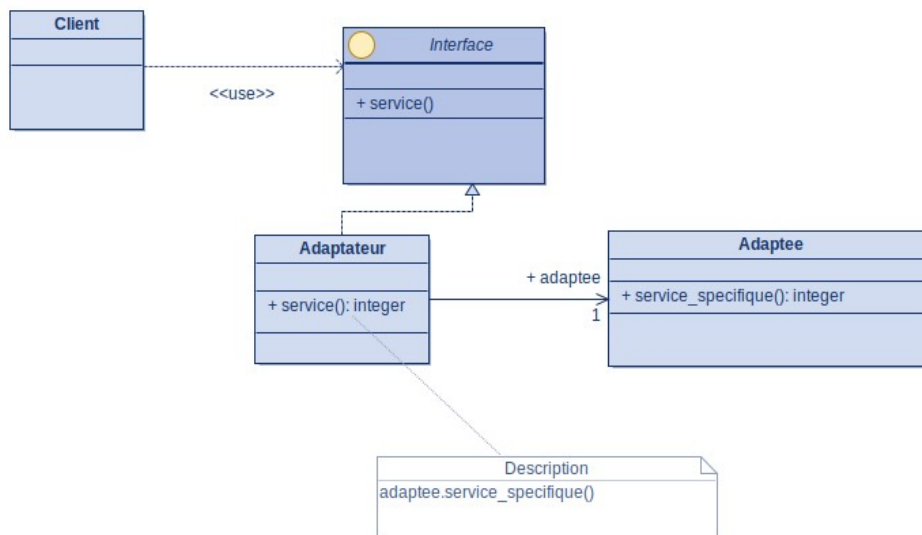
1. Quel est le patron de conception adapté à ce problème
2. Donner sa structure générique et décrire les participants
3. Proposez une solution qui met en œuvre ce patron pour résoudre problème
4. Donner le code de votre solution

solution

1. le patron adapter pour ce problème est le patron adapter

2.

2.1 structure générique



2.2 liste des participants

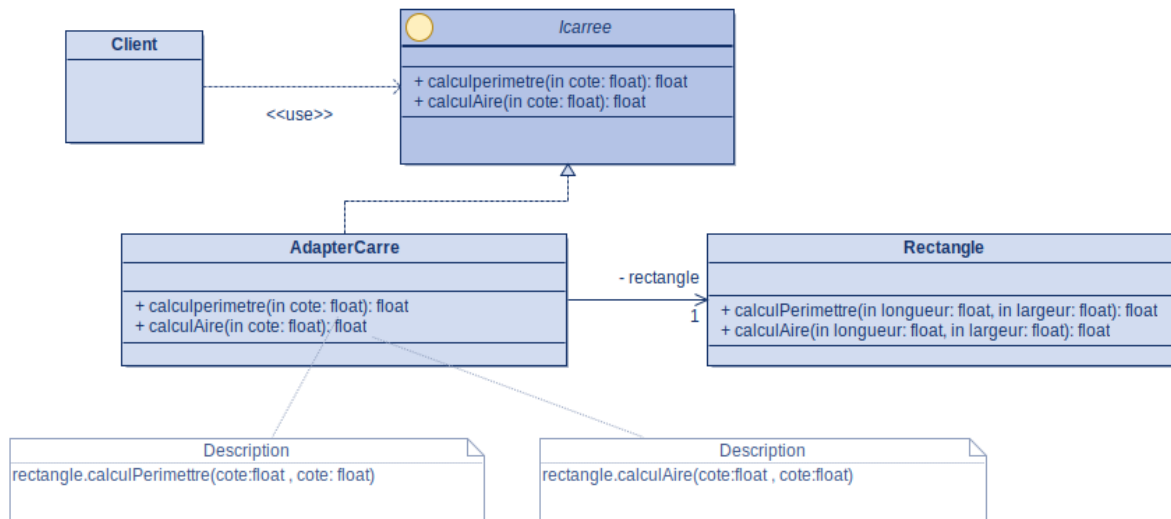
le client: c'est le programme principale qui a connaissance de l'interface et qui sais l'utiliser.

L'interface: que l'on souhaite adapter pour utiliser un élément

Adaptateur: l'élément qui converti l'interface pour l'adapter a utiliser l'élément que le client souhaite utiliser;

Adaptée: l'élément que le programme souhaite utiliser

3



4 voir code

Exercice2 composite

On considère un système de gestion de fichiers. Ce système gère des dossiers(répertoires) et chaque répertoire peut contenir des sous répertoires (pouvant eux aussi contenir des sous répertoires) et des fichiers. Le système ne connaît que 2 types de fichiers : les fichiers TXT et les fichiers PDF.

Chaque élément du système de fichier est caractérisé par :

- Deux attributs : nom et type (txt, pdf, ou dossier)
- Des méthodes :
- `decire` : affiche le nom et le type de l'élément. Un dossier affiche en plus, la

description de tous les éléments qu'il contient.

- ajouter un élément à un autre
- supprimer un élément d'un autre
- obtenir le ieme élément d'un autre

1. Quel est le patron de conception adapté pour ce problème ? Justifier
2. Donner sa structure générique et décrire les participants
3. Proposer une modélisation du problème à l'aide de ce patron
4. Donner le code Java de votre conception.

Solution

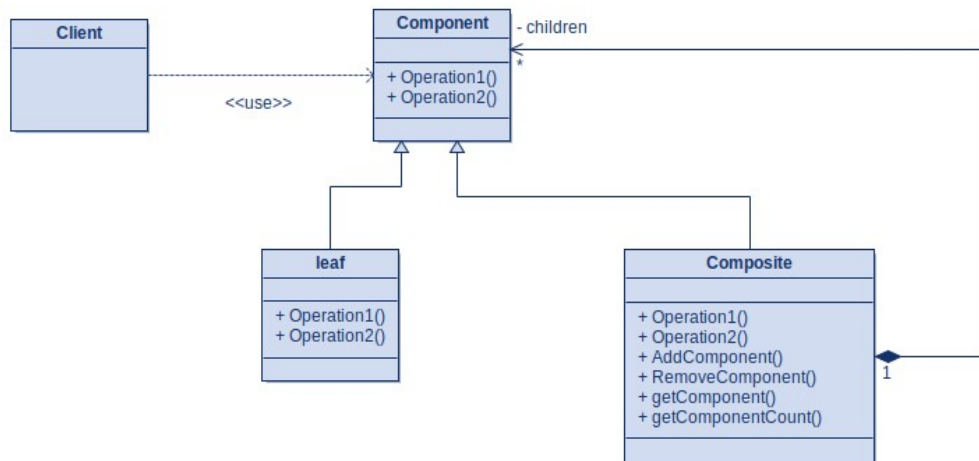
1. le patron adapter pour ce problème est le patron Composite car il nécessite une structure arborescente et où les dossier(conteneur ou objet complexe) peuvent contenir des fichiers ou d'autre dossier et où les opérations (comme afficher, ajouter, ou supprimer) doivent être appliquées de manière uniforme aux fichiers et aux dossiers. Le patron Composite est idéal pour gérer ce type de problème, car :

Il permet de traiter les objets individuels (fichiers) et les conteneurs (dossiers) de manière uniforme.

Il simplifie les opérations récursives comme afficher ou supprimer un élément.

2.

2.1 structure générique



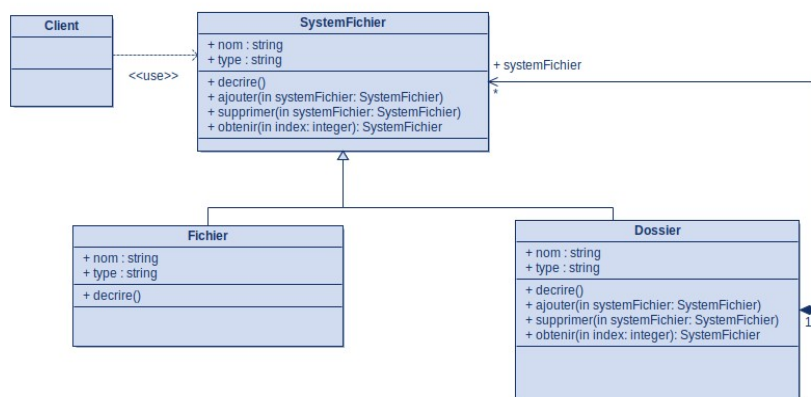
2.2 liste des participants

le client: c'est le programme principale qui a connaissance de l'interface et qui sais l'utiliser pour interagir uniformément avec les objets leaf et composite.

Component: déclare les l'interface commune a tout les objets de la structure

leaf: implémente le comportement élémentaire , objet finaux qui n'ont pas besoin de sous élément.

Composite: décrit le comportement des composant ayant plusieurs fils, contient une collection d'autre composant



4 voir code