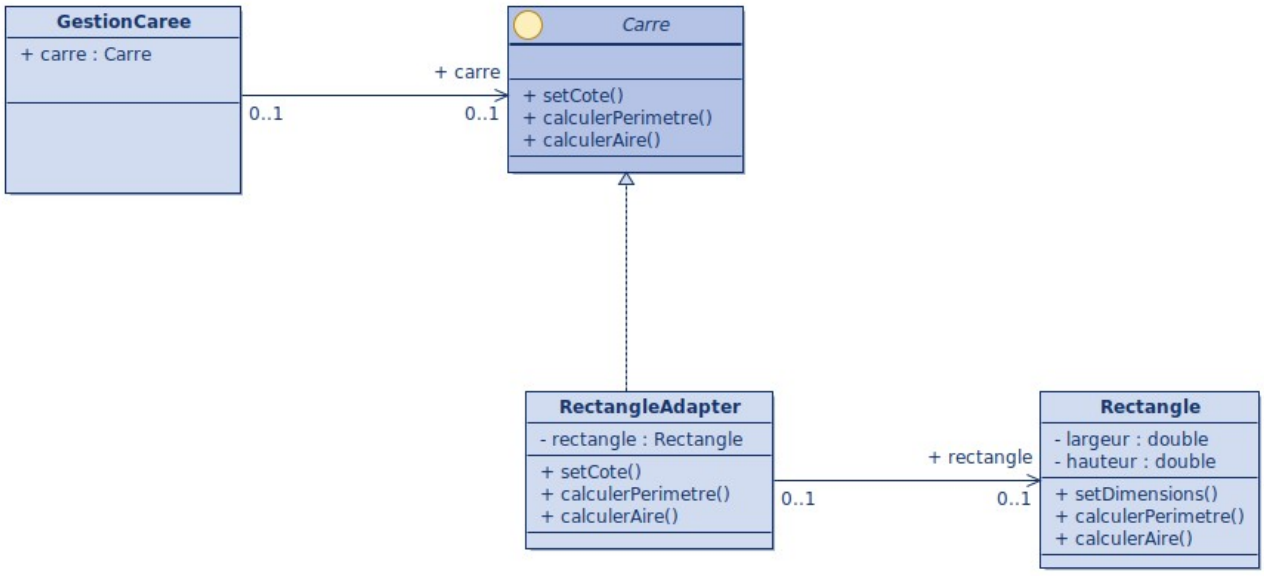
DEVOIR D'INFO 4067

Exercice 1

1. Le patron de conception adapté a ce problème c'est le patron **Adapter,** par ce que l' Adapter est utilisé pour faire correspondre une interface attendue par le client avec une classe existante incompatible afin qu'ils puissent travailler ensemble. Ici, le programme attend une interface pour gérer les carrés, mais nous avons une classe pour manipuler les rectangles. L' Adapter agira comme un intermédiaire pour rendre les rectangles utilisables comme carrés.
2. Structure générique et description des participants :
    - **Client** (GestionCarre) : Le programme qui utilise l'interface Carre.
    - **Interface** (Carré) : Interface attendue par le client.
    - **Adaptée** (Rectangle) : Classe existante pour manipuler les rectangles.
    - **Adapter** (RectangleAdapter) : Une classe intermédiaire qui implémente Interface en utilisant un Rectangle.
3. Mise en œuvre du patron Adapter



Exercice 2

1. Le patron de conception adapté a ce problème c'est le patron **Composite**, par ce que le patron composite permet de structurer un système sous la forme d'une hiérarchie d'objets. Les fichiers et les dossiers peuvent être traités de manière uniforme comme des "éléments" grâce a une interface ou une classe abstraite commune. Les opérations (décrire, ajouter, supprimer et obtenir) s'appliquent aussi bien aux éléments simples (fichiers) qu'aux conteneurs (dossiers).
2. Structure générique et description des participants :
    - **Component** : Une classe abstraite Element représentant les propriétés (nom, type) et les méthodes de base.
    - **Leaf :** Deux classes concrètes, FichierTXT et FichierPDF, qui héritent de Element.
    - **Composite** : Une classe Dossier qui contient une liste d'objets Element et implémente les méthodes de gestion des enfants (ajouter, supprimer, obtenir).
    - **Client** (GestionFichier) : Le programme qui utilise la classe abstraite Element.

3. Modélisation du problème avec le patron Composite :