



# Design Pattern : Composite

Baptiste Lanvin  
Baptiste Posseme  
Antoine Braud  
Bafodé Koulibaly

# Plan

---

# Introduction

Définition : Les design patterns sont des solutions classiques à des problèmes récurrents de la conception de logiciels.

Quatre éléments dans la description d'un design pattern :

- L'Intention
- La Motivation
- La Structure
- L'Exemple de code

# Introduction

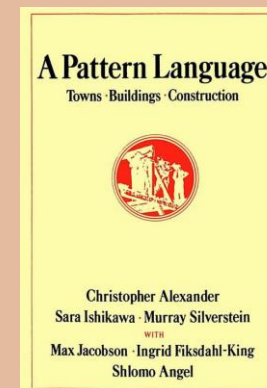
Trois groupes principaux de design pattern sont :

- Patrons de création
- Patrons structurels
- Patrons comportementaux

# Introduction

Christopher Alexander

Essai datant de 1977 nommé : A Pattern Language: Towns, Buildings, Construction

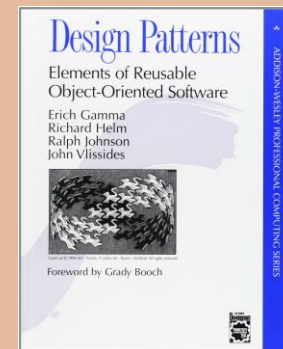


# Introduction

« Gang Of Four » composé de Erich Gamma, John Vlissides, Ralph Johnson, et Richard Helm.

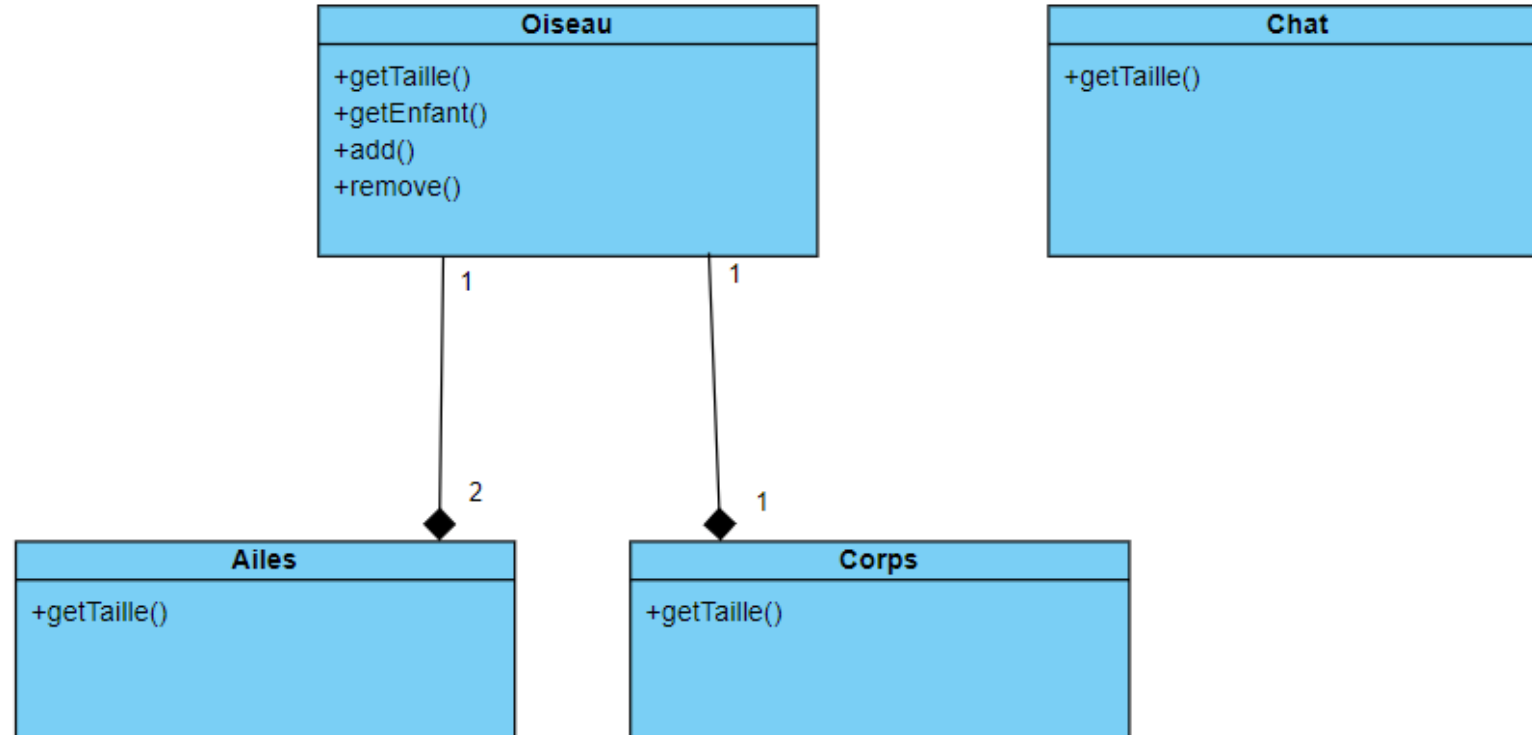
Livre datant de 1994, nommé : [Design Patterns : Catalogue de modèles de conception réutilisables](#)

23 design patterns



# Modélisation

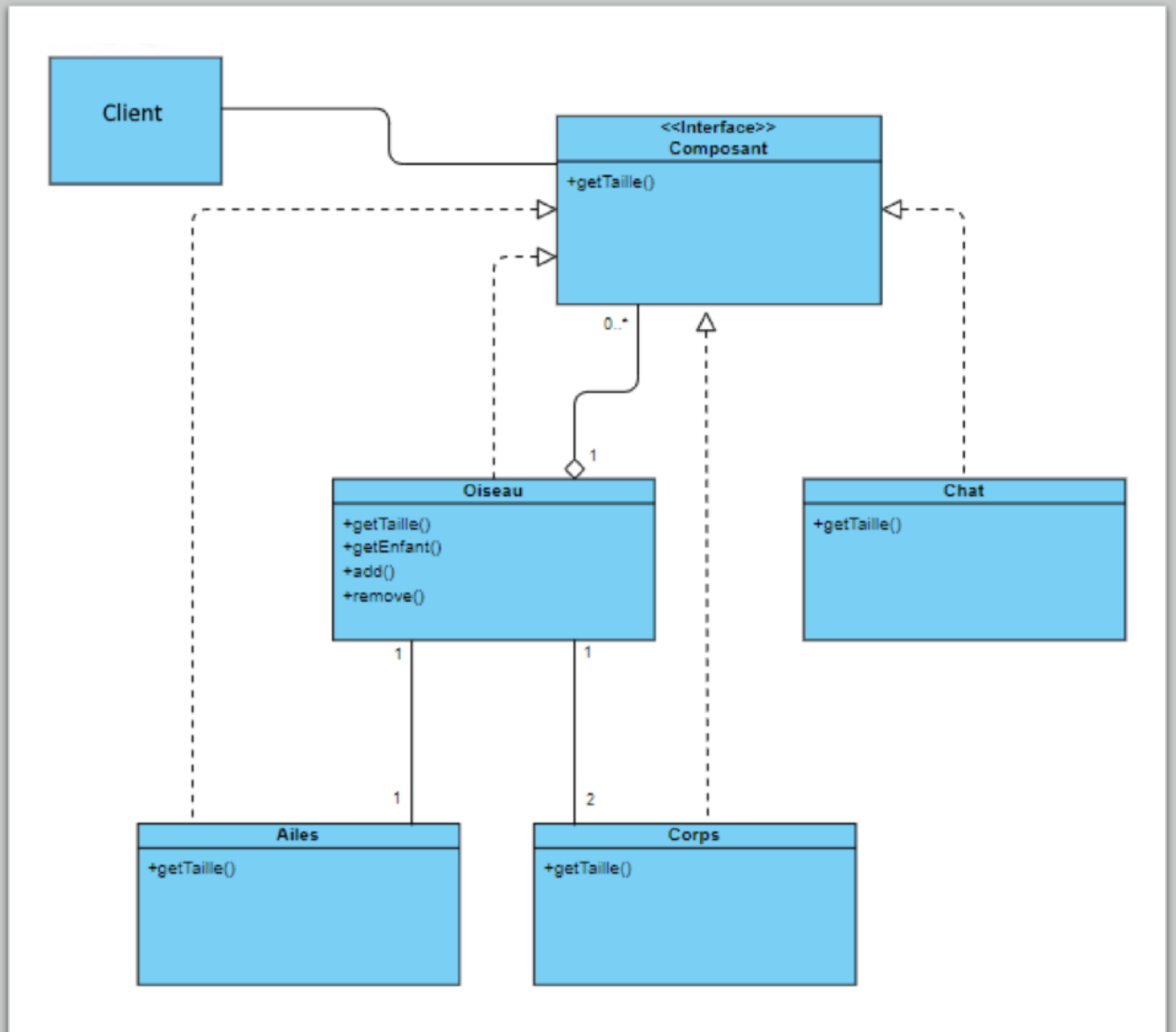
---

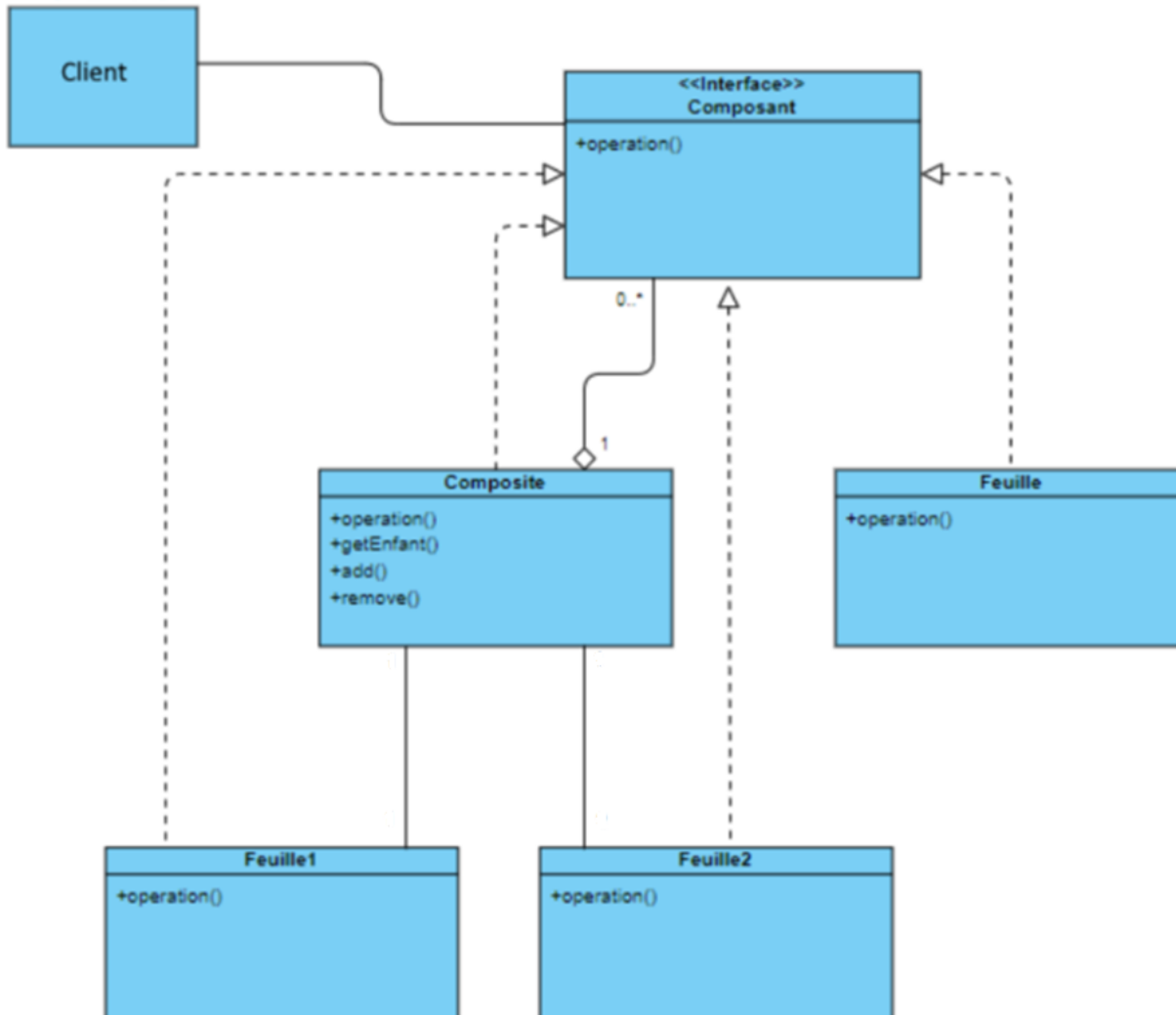


# Sans Composite



# Avec Composite





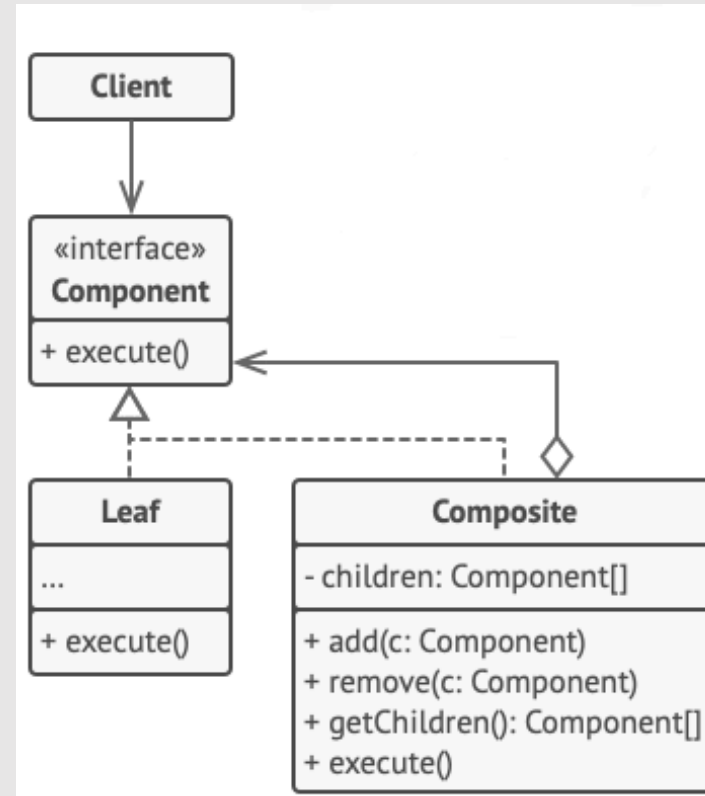
# Généralisation

# Composite

---

- Design Pattern Structuraux => Organise les classes d'un programme
- Il est réservé aux applications aux structures arborescentes
- Uniformité avec les objets et les compositions d'objets

# Diagramme de Classes



**Classe Client :**

- Manipule les objets de la composition

**Interface Component :**

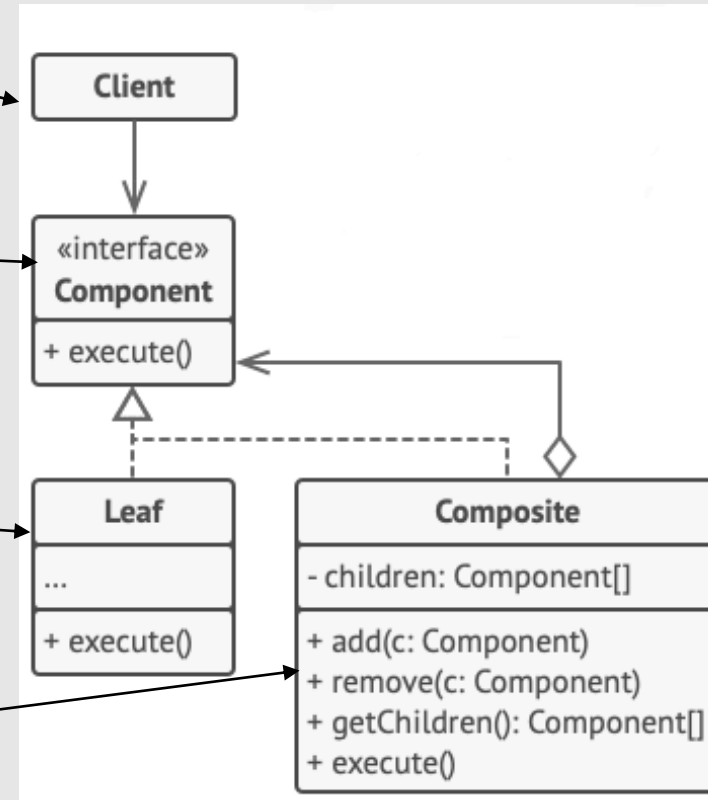
- Interface pour tous les objets
- Décrire les méthodes

**Classe Leaf :**

- Composant qui n'a pas de sous-élément

**Classe Composite :**

- Elle peut avoir des sous-éléments enfants et délègue les tâches



# Principe SOLID

---



- Principe de responsabilité unique
- Principe d'ouverture/fermeture
- Principe de substitution de Liskov
- Principe de séparation des interfaces
- Principe d'inversion des dépendances

## Avantages

Code simple et léger pour le client

Flexibilité et extensibilité

Travail sur des structures complexes plus facile

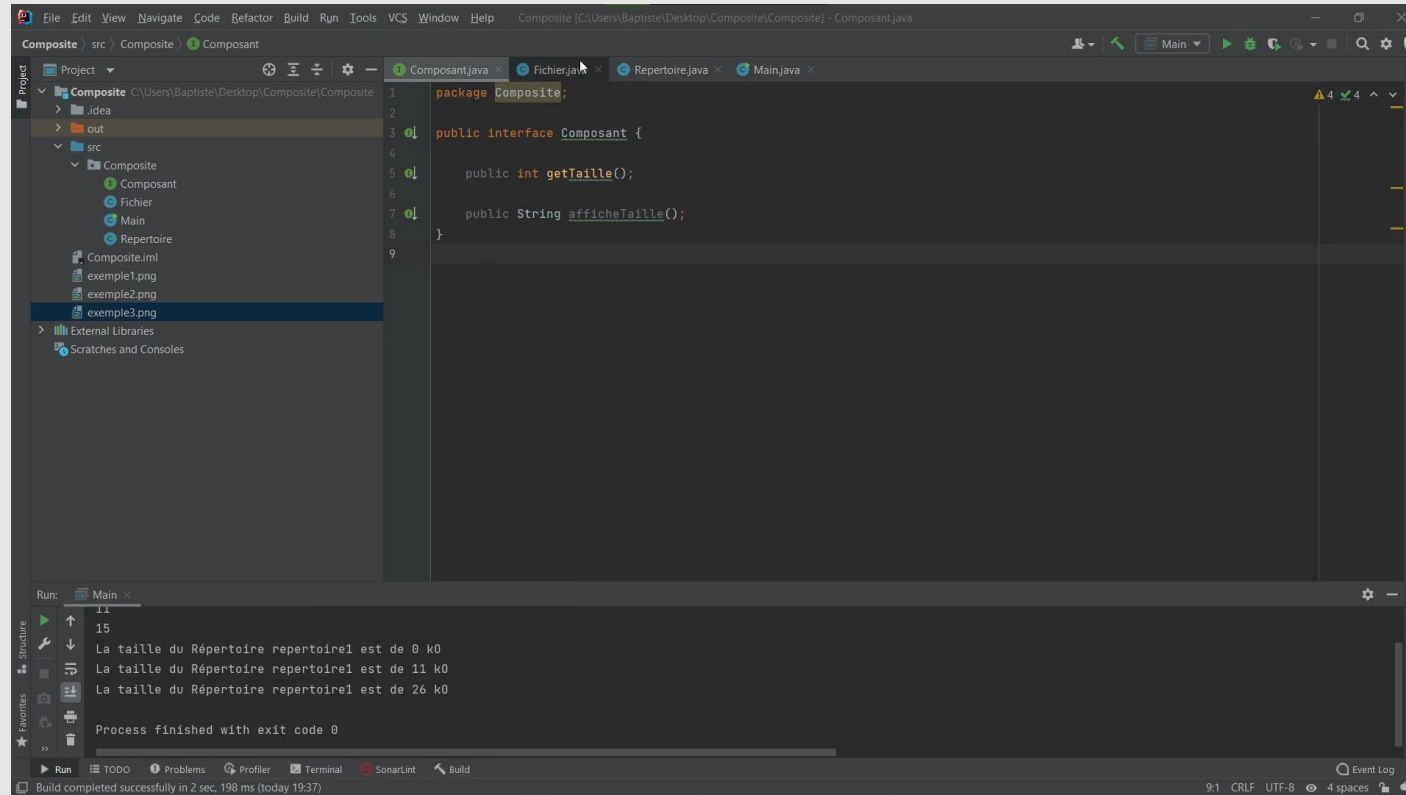
## Inconvénients

Mise en œuvre car l'uniformisation du code pose problème



# Live coding

---



<https://youtu.be/5kSf9JYIEs0>

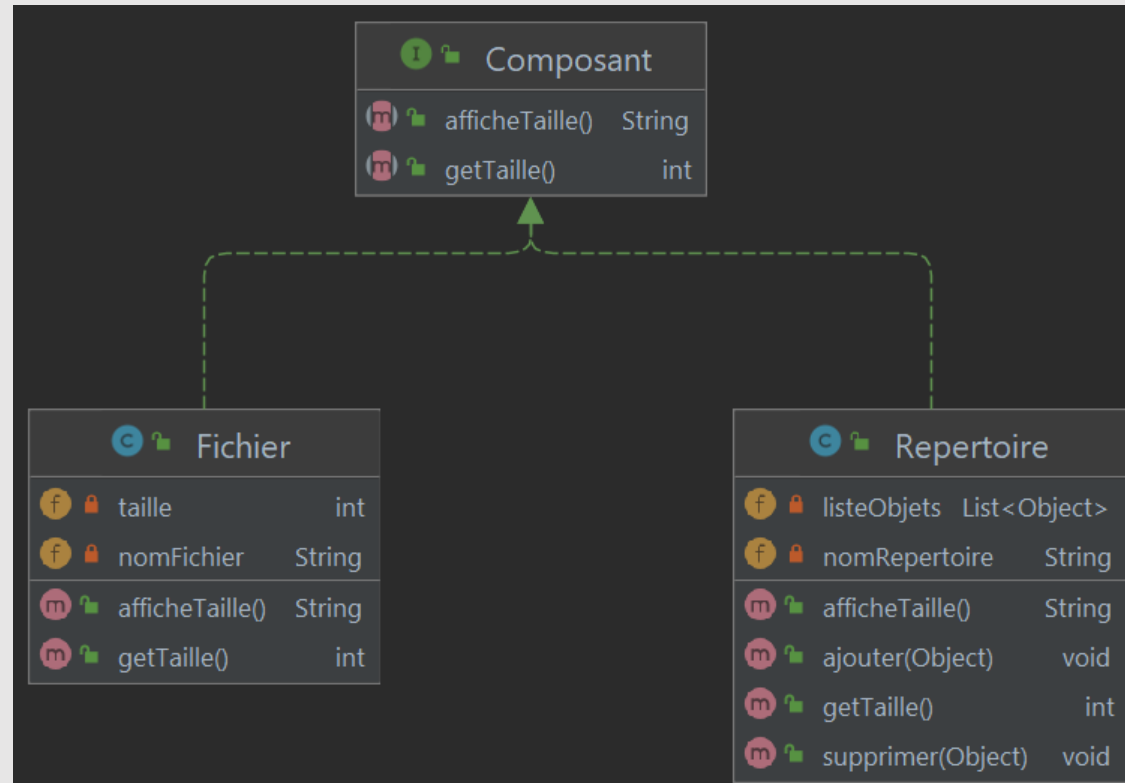


Diagramme de Classes

# Classes Java utilisées pour le pattern Composite



Java.awt.Component

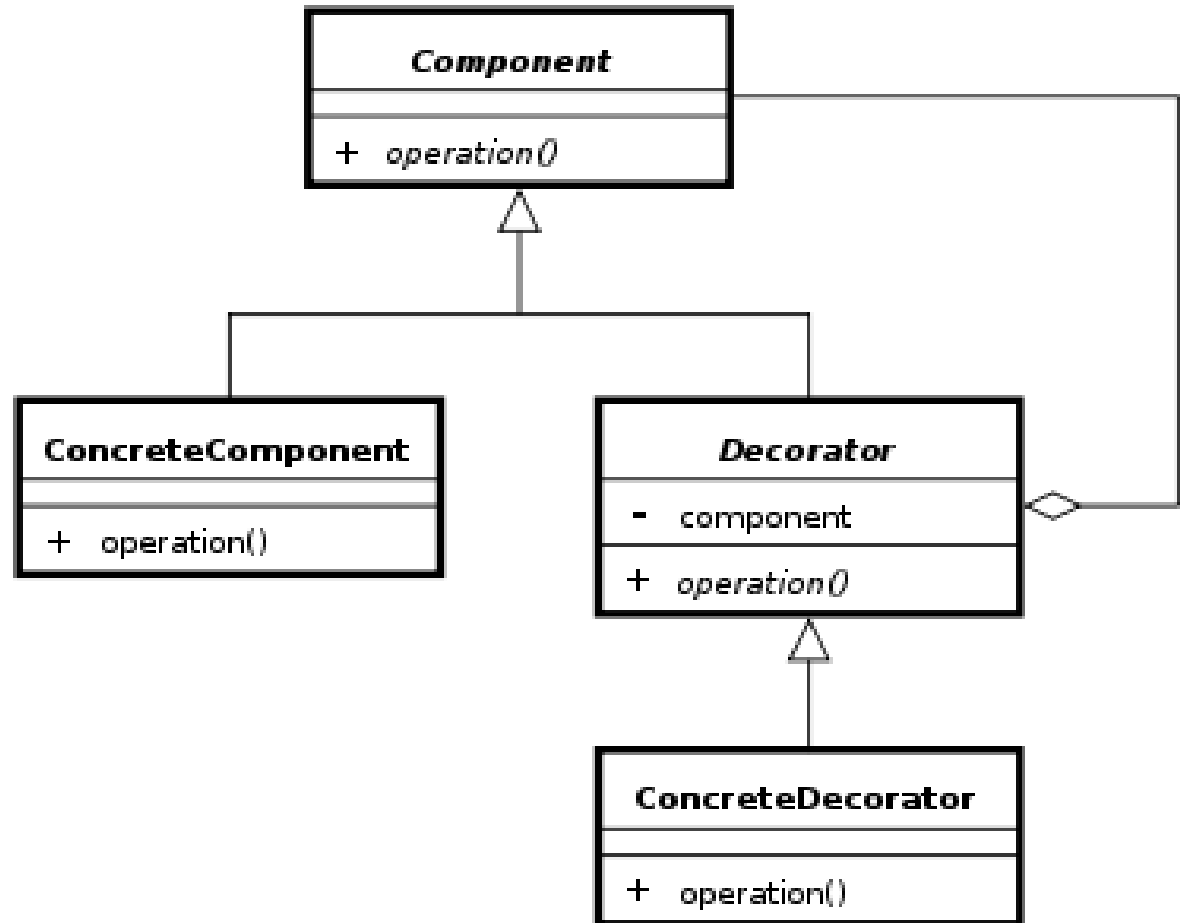
Java.awt.Button

Java.awt.Container

# **Le Design Pattern Decorator**

---

- Design Pattern structurel
- Utilise une classe abstraite
- Diagrammes similaires
- Même principe de relation



# Quizz

---

## Sources :

<https://www.codingame.com/playgrounds/10154/design-pattern-composite/presentation>

<https://www.novencia.com/design-pattern-du-gof-gang-of-four/>

<https://refactoring.guru/fr/design-patterns/composite>

<https://www.ionos.fr/digitalguide/sites-internet/developpement-web/composite-pattern/>

[https://en.wikipedia.org/wiki/Composite\\_pattern](https://en.wikipedia.org/wiki/Composite_pattern)

<https://docs.oracle.com/javase/8/docs/technotes/tools/windows/javadoc.html>

[http://miageprojet2.unice.fr/@api/deki/files/1457/=05\\_Composite.pdf](http://miageprojet2.unice.fr/@api/deki/files/1457/=05_Composite.pdf)

<https://www.youtube.com/watch?v=25nonh-W5po>

[https://perso.esiee.fr/~bureaud/Unites/ln4r21/0809/Docs/patterns\\_coinvide\\_cours4.pdf](https://perso.esiee.fr/~bureaud/Unites/ln4r21/0809/Docs/patterns_coinvide_cours4.pdf)

<https://github.com/amilajack/reading/blob/master/Design/GOF%20Design%20Patterns.pdf>

<https://medium.com/elp-2018/composite-design-pattern-7d079a2bc86d>