

Nom : FOGUEM  
Prenom: JUNIOR KIERAN  
Matricule : 22U2095

## INF4067 : UML et Design Patterns – Patterns de Structuration

### Pattern Composite

#### 1. Quel est le patron de conception adapté pour ce problème ? Justifier

Le patron de conception adapté pour ce problème est le **patron Composite**.

##### Justification :

Le problème décrit une structure arborescente où des objets (dossiers) contiennent d'autres objets (fichiers ou sous-dossiers). On demande que l'utilisateur puisse appeler les mêmes méthodes (décrire, ajouter, etc...) sur un fichier simple ou sur un dossier complexe sans avoir à distinguer leur type au préalable.

#### 2. Donner sa structure générique et décrire les participants

La structure générique du patron Composite est la suivante :

##### Component :

C'est l'interface ou la classe abstraite commune à tous les objets de la composition (simples ou composés).

##### Leaf (Feuille) :

Implémente le comportement élémentaire

##### Composite (Composé) :

définit le comportement des composants ayant des fils, stocke les fils et implémente les opérations nécessaires à leur gestion.

##### Client :

Manipule les objets de la composition via l'interface du Component.

#### 3. Proposer une modélisation du problème à l'aide de ce patron

**Component** : dans le cas de notre exercice il s'agit du SystemFichier

\* Attributs : nom, type.

\* Méthodes : décrire(), ajouter(), supprimer(), obtenir()

**Leaf** : Nous créons une classe Fichier qui hérite de SystemFichier.

\* Elle implémentera décrire() pour afficher ses propres informations.

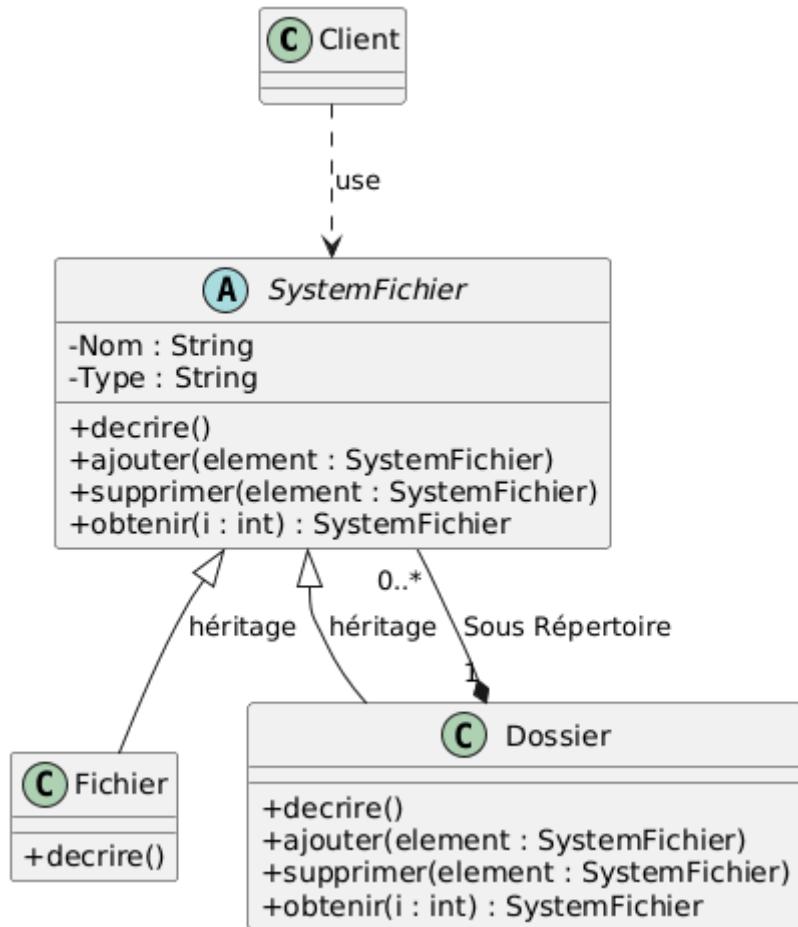
\* Elle n'aura pas à réécrire les méthodes de gestion d'enfants, héritant ainsi du comportement par défaut (lever une exception).

**Composite** : Nous créons une classe Dossier qui hérite de SystemFichier.

\* Elle contiendra une List<SystemFichier> pour stocker ses enfants.

\* Elle implémentera décrire() : elle affichera son nom, puis parcourra sa liste d'enfants en appelant décrire() sur chacun d'eux.

\* Elle réécrira les méthodes ajouter(), supprimer(), obtenir() pour manipuler sa liste d'enfants.



4.Donner le code Java de votre conception.

Visual Studio Code 20 déc. 21:31

Main.java

```
File Edit Selection View Go Run Terminal Help ← → Search — Main.java - Visual Studio Code
```

home > super-kiki > Bureau > TP\_TD\_INF\_4067 > FOGUEM\_Junior\_Kieran\_22U2095 > Code\_Patron\_Composite > Main.java > ...

```
1 import java.util.ArrayList;
2 import java.util.List;
3
4 // 1. Le Composant Abstrait
5 abstract class SystemFichier {
6     protected String nom;
7     protected String type;
8
9     public SystemFichier(String nom, String type) {
10         this.nom = nom;
11         this.type = type;
12     }
13
14     public abstract void decrire();
15     public abstract void ajouter(SystemFichier e);
16     public abstract void supprimer(SystemFichier e);
17     public abstract SystemFichier obtenir(int i);
18 }
19
20 // 2. La Feuille (Fichier)
21 class Fichier extends SystemFichier {
22     public Fichier(String nom, String type) {
23         super(nom, type); // type sera "txt" ou "pdf"
24     }
25
26     @Override
27     public void decrire() {
28         System.out.println("Fichier: " + nom + " [Type: " + type + "]");
29     }
30
31     // Un fichier ne peut pas contenir d'éléments
32     @Override
33     public void ajouter(SystemFichier e) {
34         /* Optionnel: lever une exception ou afficher une erreur */
35     }
36
37     @Override
38     public void supprimer(SystemFichier e) { }
```

Visual Studio Code      20 déc. 21:32

Main.java 1 ×

```
home > super-kiki > Bureau > TP_TD_INF_4067 > FOGUEM_Junior_Kieran_22U2095 > Code_Patron_Composite > Main.java > ...
```

```
21  class Fichier extends SystemFichier {  
22  
23      @Override  
24      public SystemFichier obtenir(int i) {  
25          return null;  
26      }  
27  }  
28  
29 // 3. Le Composite (Dossier)  
30 class Dossier extends SystemFichier {  
31     private List<SystemFichier> enfants = new ArrayList<>();  
32  
33     public Dossier(String nom) {  
34         super(nom, type: "dossier");  
35     }  
36  
37     @Override  
38     public void decrire() {  
39         System.out.println("\n--- Dossier: " + nom + " ---");  
40         for (SystemFichier e : enfants) {  
41             e.decrire(); // Appel récursif (Cœur du patron Composite)  
42         }  
43     }  
44  
45     @Override  
46     public void ajouter(SystemFichier e) {  
47         enfants.add(e);  
48     }  
49  
50     @Override  
51     public void supprimer(SystemFichier e) {  
52         enfants.remove(e);  
53     }  
54  
55     @Override  
56     public SystemFichier obtenir(int i) {  
57         if (i >= 0 && i < enfants.size()) {  
58             return enfants.get(i);  
59         }  
60     }  
61 }
```

Visual Studio Code      20 déc. 21:33

Main.java 1 ×

```
home > super-kiki > Bureau > TP_TD_INF_4067 > FOGUEM_Junior_Kieran_22U2095 > Code_Patron_Composite > Main.java > ...
```

```
80  
81 // Exemple d'utilisation  
82 public class Main {  
83     Run|Debug  
84     public static void main(String[] args) {  
85         // Cr ation des fichiers  
86         SystemFichier f1 = new Fichier(nom: "notes", type: "txt");  
87         SystemFichier f2 = new Fichier(nom: "rapport", type: "pdf");  
88  
89         // Cr ation des dossiers  
90         Dossier root = new Dossier(nom: "C:");  
91         Dossier docs = new Dossier(nom: "MesDocuments");  
92  
93         // Construction de l'arbre  
94         docs.ajouter(f1);  
95         docs.ajouter(f2);  
96         root.ajouter(docs);  
97         root.ajouter(new Fichier(nom: "config", type: "txt"));  
98  
99         // Affichage uniforme via l'interface SystemFichier  
100        root.decrire();  
101    }
```

## RESULTAT DE CETTE IMPLEMENTATION :

```
super-kiki@superkiki-XPS-13-9350:~/Bureau/TP_TD_INF_4067/FOGUEM_Junior_Kieran_22
U2095$ cd Code_Patron_Composite/
super-kiki@superkiki-XPS-13-9350:~/Bureau/TP_TD_INF_4067/FOGUEM_Junior_Kieran_22
U2095/Code_Patron_Composite$ javac Main.java
super-kiki@superkiki-XPS-13-9350:~/Bureau/TP_TD_INF_4067/FOGUEM_Junior_Kieran_22
U2095/Code_Patron_Composite$ java Main

--- Dossier: C: ---

--- Dossier: MesDocuments ---
Fichier: notes [Type: txt]
Fichier: rapport [Type: pdf]
Fichier: config [Type: txt]
super-kiki@superkiki-XPS-13-9350:~/Bureau/TP_TD_INF_4067/FOGUEM_Junior_Kieran_22
U2095/Code_Patron_Composite$
```