

TP 9**PATTERN COMPOSITE, SUITE
DU DEVOIR DE CC**

Ce TP comporte un exercice pouvant être réalisé en 15 à 40 minutes, puis est consacré à la suite du CC.

1. Architecture de composants graphiques avec le pattern Composite

Nous souhaitons mettre en place une architecture dédiée à des éléments graphiques comme en AWT ou Swing. Le principe est qu'un composant peut être soit un élément de base tel qu'un Bouton ou un Label, soit un Panel dans lequel on peut insérer des éléments (donc éventuellement, d'autres panels).

Ecrire un diagramme de classes s'appuyant sur le pattern Composite. On pourra proposer une implémentation abstraite de l'interface Composant si l'on souhaite mutualiser certaines choses.

La méthode essentielle d'un composant est « void display() », qui affiche le contenu du composant. Bien sûr, on se contentera ici d'écrire le type et le nom du composant via System.out.println().

Optionnellement, on pourra faire en sorte que l'affichage du contenu d'un Panel soit indenté (en utilisant un paramètre int offset dans la méthode display).

À titre d'illustration, voici un exemple de main() et la sortie correspondante :

```
public static void main(String[] args) {  
  
    Panel p1 = new Panel("panneau 1");  
    p1.ajoutComposant(new Bouton("b1"));  
  
    Panel p2 = new Panel("panneau 2");  
    p2.ajoutComposant(new Bouton("b2"));  
    p2.ajoutComposant(new Label("l1"));
```

```

p1.ajoutComposant(p2);

Panel p3 = new Panel("panneau 3");
p3.ajoutComposant(new Label("I2"));
p3.ajoutComposant(new Label("I3"));
p3.ajoutComposant(new Panel("panneau 4"));

p2.ajoutComposant(p3);

p1.display(0);
System.out.println();
new Bouton("bouton seul").display(0);

}

```

Exécution :

```

[Panel] panneau 1 contenant :
[Bouton] b1
[Panel] panneau 2 contenant :
[Bouton] b2
[Label] I1
[Panel] panneau 3 contenant :
[Label] I2
[Label] I3
[Panel] panneau 4(vide)

```

```

[Bouton] bouton seul

```

2. Suite du travail sur le devoir de CC