



Le génie pour l'industrie

LOG121

Conception orientée objet

Patron Composite

Enseignante: Souad Hadjres

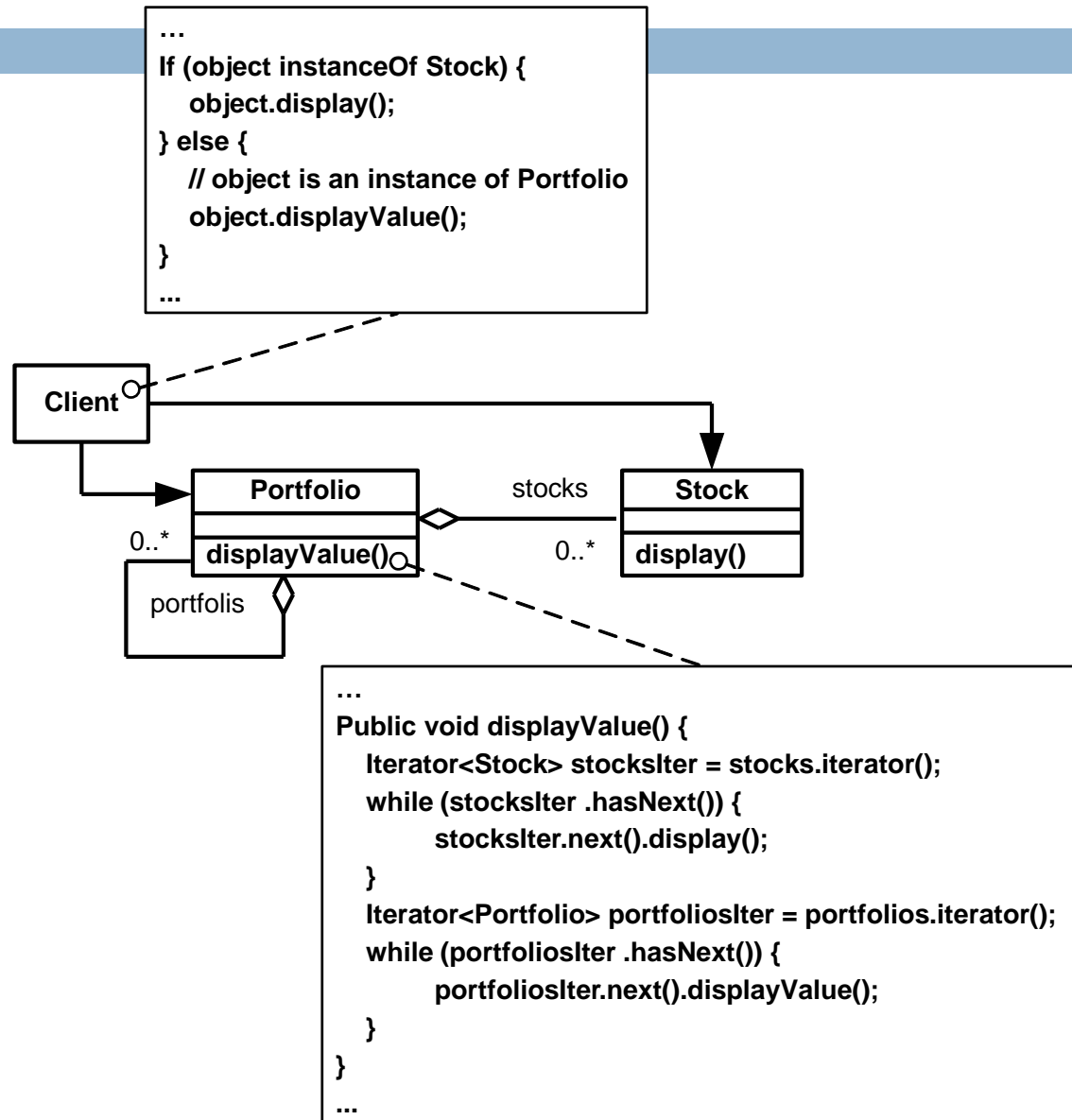
Exemple de problème de conception

2

- Dans une application financière qui supporte la gestion des portefeuilles
 - ▣ On veut construire des portefeuilles (Portfolio) qui sont composés à partir d'actions individuelles (Stock)
 - ▣ Les portefeuilles peuvent à leur tour être groupés pour former de plus larges portefeuilles ou comptes
 - ▣ On aimerait afficher l'état des actions et des portefeuilles

Exemple de problème de conception

3



Exemple de problème de conception

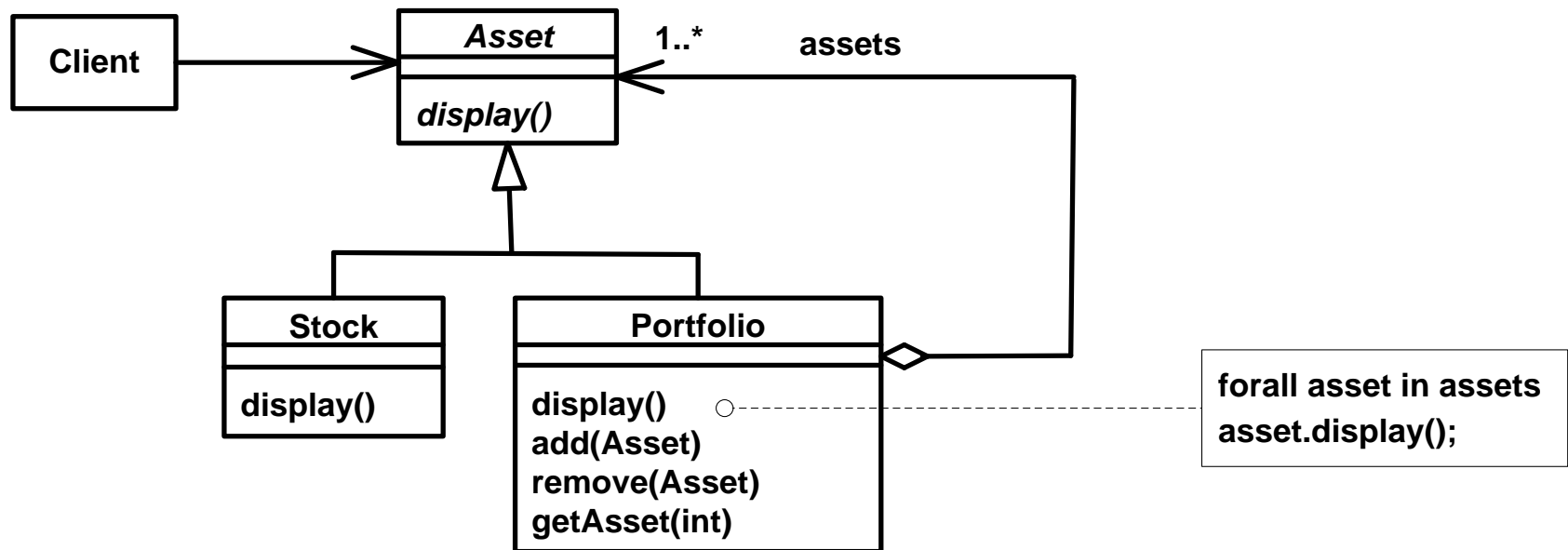
4

- Les problèmes avec cette conception
 - ▣ Le code client traite les objets de type Stock et les objets de type Portfolio différemment, malgré que l'utilisateur veuille les traiter identiquement
 - un utilisateur veut afficher l'état actuel de ses actifs qu'ils soient de type Stock ou de type Portfolio
 - ▣ Le code client est plus complexe et plus difficile à maintenir et à étendre
 - plusieurs blocks conditionnels
- Comment améliorer cette conception?

Solution au problème

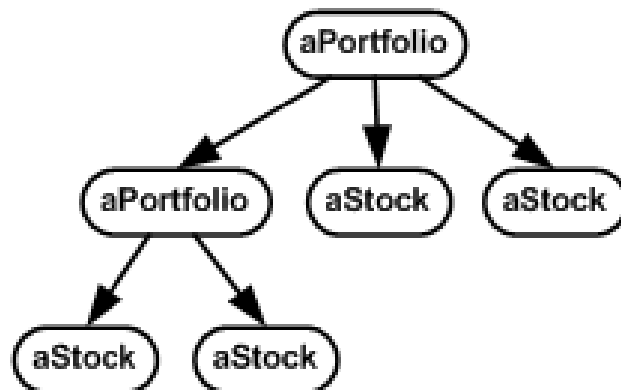
5

- Définir une classe abstraite (Asset) qui représente autant les actions simples (Stock) que les portefeuilles (Portfolio).
- Cette classe définit les opérations communes à tous les actifs qu'ils soient simples ou composés. Par exemple: l'opération `display()`.



□ Contexte

- ▣ Des objets primitifs peuvent être combinés pour former des objets composites lesquels peuvent être regroupés pour former d'autres objets
- ▣ Les clients doivent utiliser de façon uniforme un objet composite comme un objet primitif



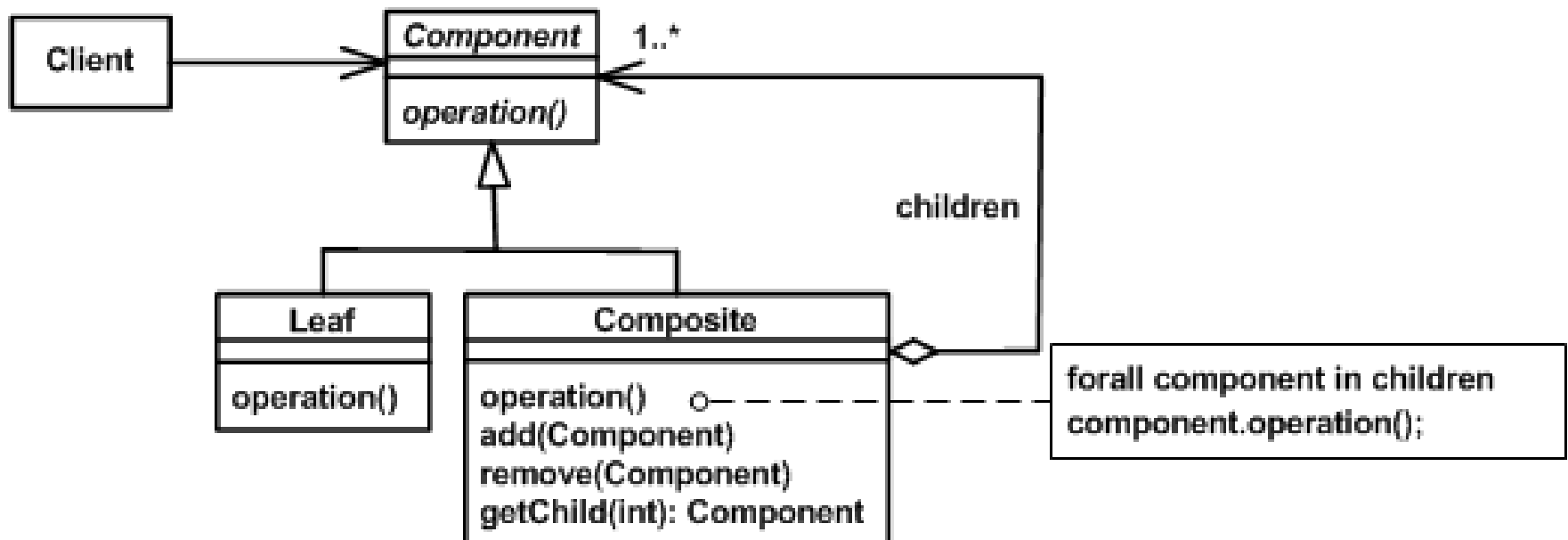
□ Solution

- ▣ Définir une interface commune aux objets primitifs et aux objets composites; appelons cette interface « Component »
- ▣ Un objet composite est construit à partir des objets de type « Component »
- ▣ Lorsqu'une méthode d'un objet composite est appelée, le composite appelle la méthode de ses composants qu'ils soient des objets primitifs ou composites, et il combine les résultats obtenus

Le patron Composite

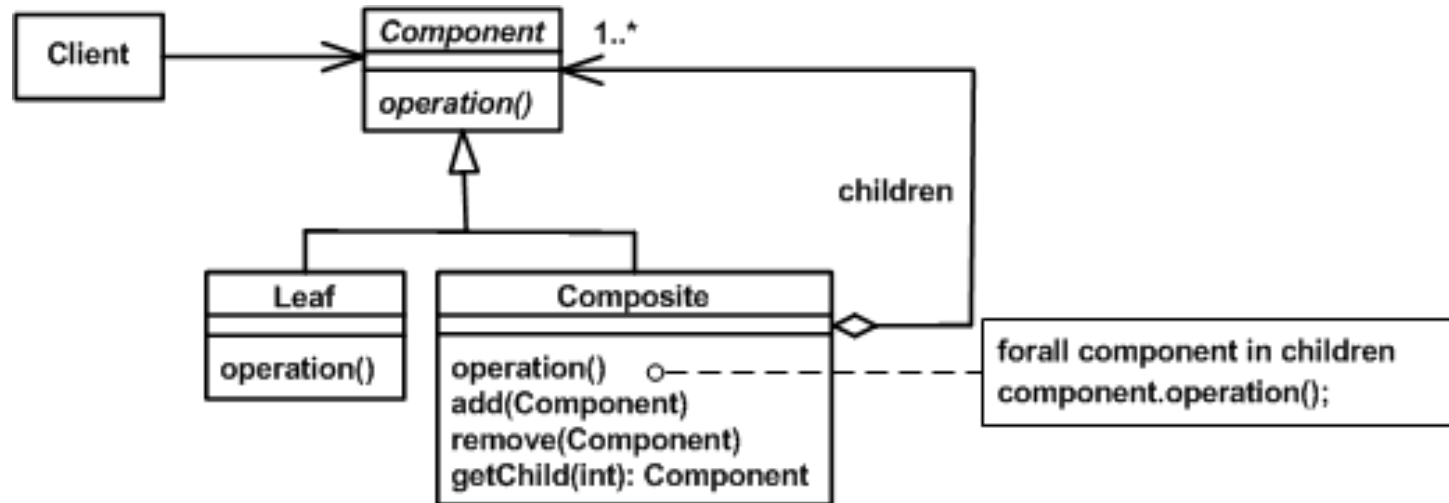
8

□ La structure générique du patron dans GoF



Le patron Composite

9

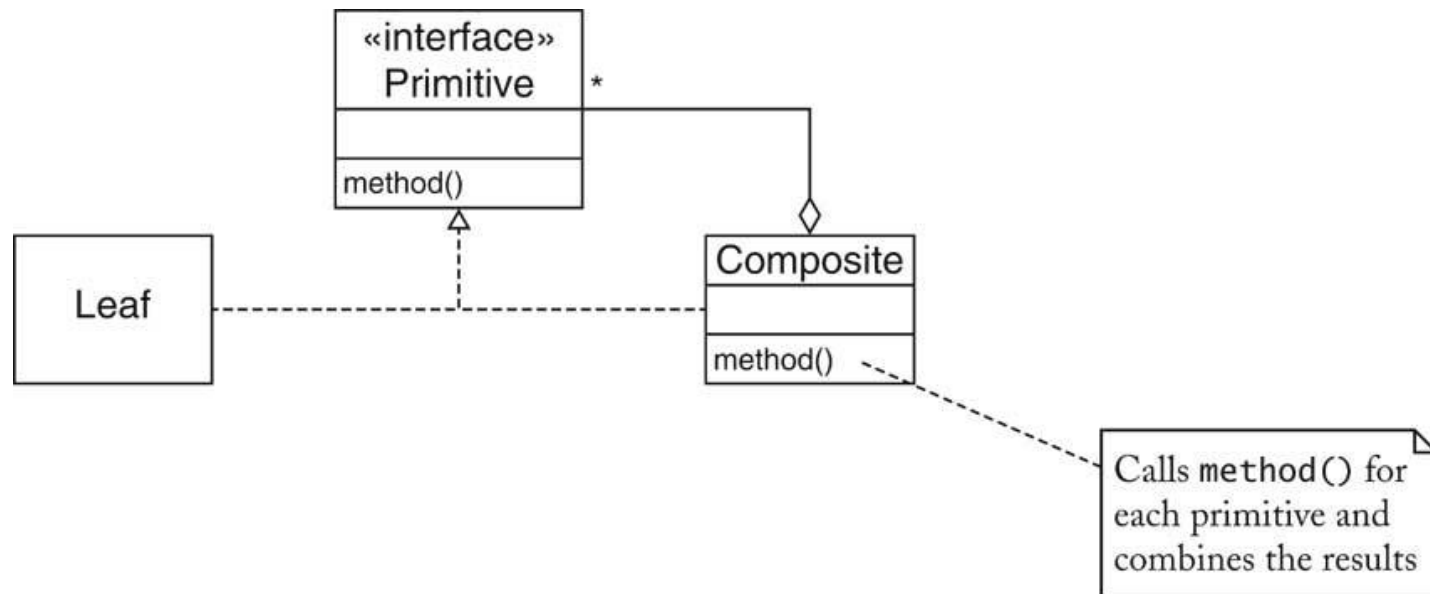


Nom dans le patron de conception	Nom dans l'exemple de la gestion des portefeuilles
Component	Asset
Composite	Portfolio
Leaf	Stock
operation()	display()

Le patron Composite

10

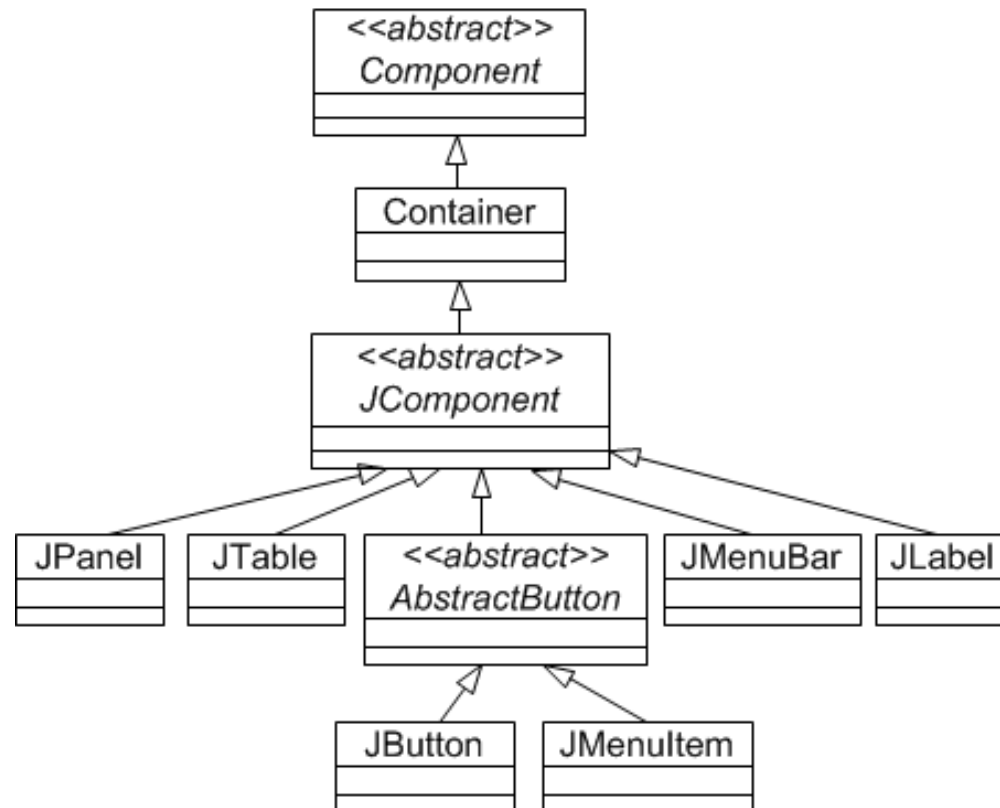
□ La structure du patron dans Horstamn



Le patron Composite

11

- Un autre exemple d'application du patron
 - ▣ Extrait de la librairie Java pour la conception des interfaces utilisateur (<http://java.sun.com/docs/books/tutorial/uiswing/>)

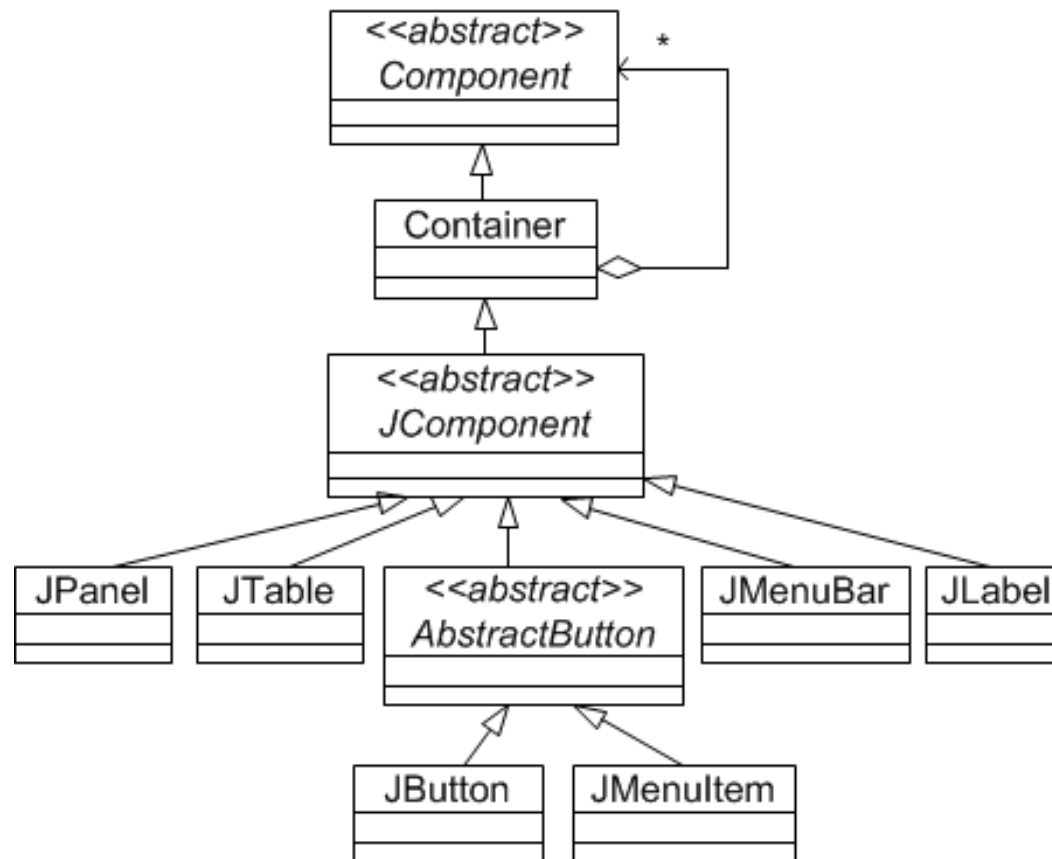


- Containers et Components de Java
 - ▣ Containers contiennent des Components de l'interface utilisateur
 - JPanel contient des JButtons
 - ▣ Containers peuvent être inclus dans un autre container
 - On peut inclure un JPanel dans un autre JPanel
 - ▣ Container est aussi un « Component »

Le patron Composite

13

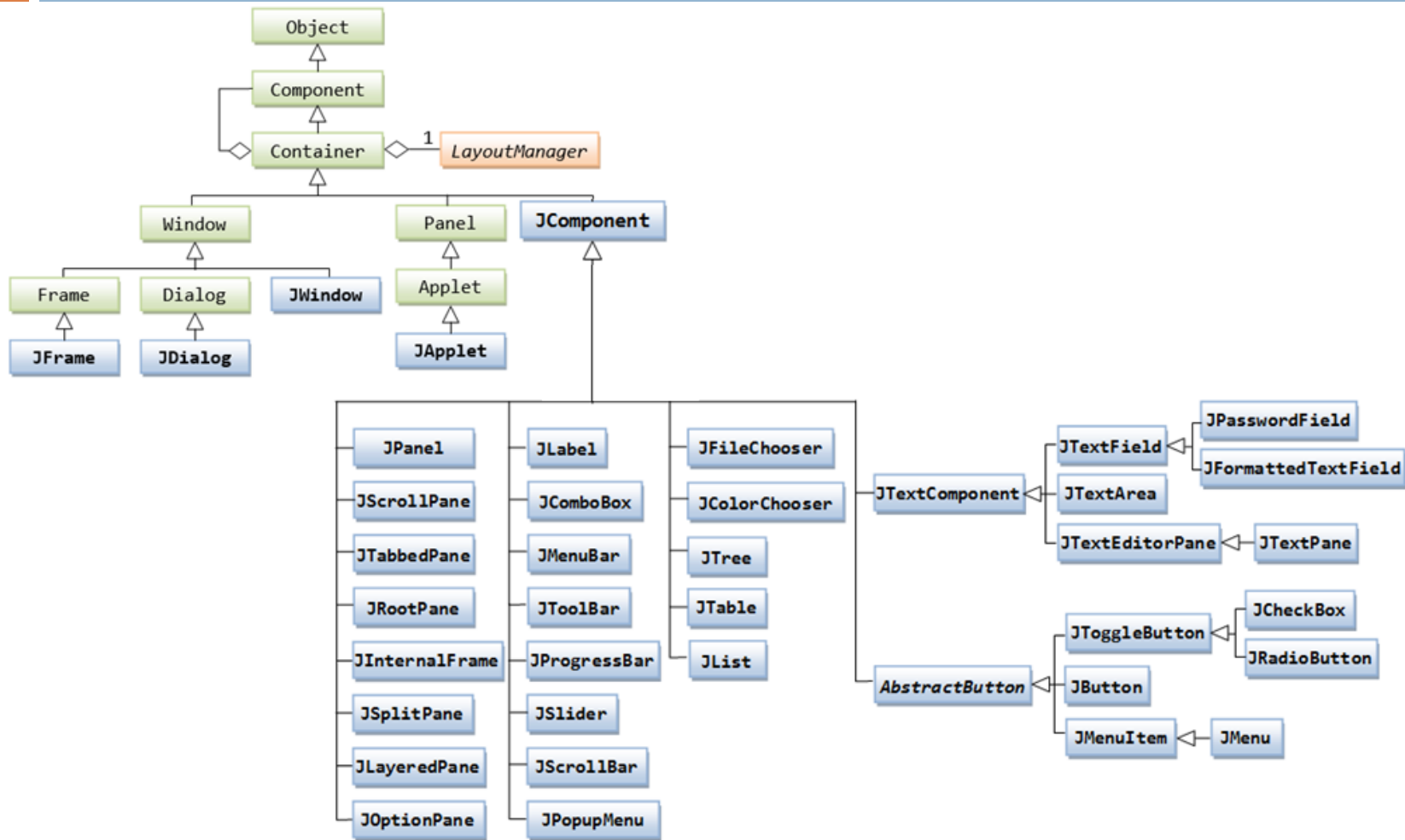
□ Containers et Components de Java



Attention: JFrame hérite de Container, mais on ne peut pas mettre un JFrame dans un autre

Le patron Composite

14



https://www3.ntu.edu.sg/home/ehchua/programming/java/J4a_GUI_2.html

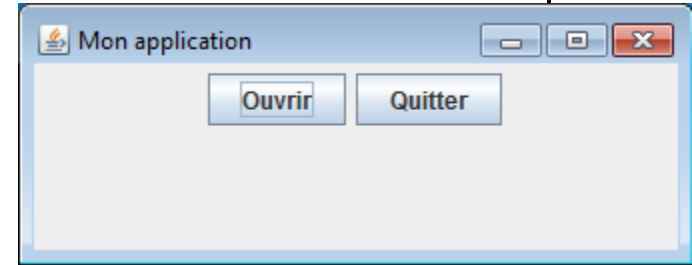
Le patron Composite

15

□ Containers et Components de Java

Nom dans le patron de conception	Nom dans l'exemple des composants graphiques Java
Component	Component
Composite	Container
Leaf	un composant sans enfant (p. ex. JButton)
operation()	une méthode de Component (exemple: getPreferredSize)

```
public class UneApplication extends JFrame implements ActionListener {
    private JButton openButton = new JButton("Ouvrir");
    private JButton exitButton = new JButton("Quitter");
    public UneApplication() {
        super("Mon application");
        setSize(400, 400);
        JPanel myPanel = new JPanel();
        myPanel.add(openButton);
        myPanel.add(exitButton);
        add(myPanel);
        openButton.addActionListener(this);
        exitButton.addActionListener(this);
    }
    public void actionPerformed(ActionEvent event) {
        if (event.getSource() == openButton)
            JOptionPane.showMessageDialog((Component)event.getSource(), "Ouvrir");
        else if (event.getSource() == exitButton)
            System.exit(0);
    }
    public static void main(String[] argv) {
        UneApplication application = new UneApplication();
        application.setVisible(true);
    }
}
```



Le patron Composite

17

- Connaissez vous d'autres exemples où le patron s'applique?