

Patrones de Diseño

...

75.10 - Técnicas de Diseño

Requerimientos

- Se necesita dibujar todos los gráficos de un contenedor.
- Ver todos los archivos a partir de un directorio
- Imprimir todos los nodos de un árbol en formato xml
- El cliente necesita conocer una única interfaz, de modo de ser simple su uso:
Ej.: `Container.draw()`; `file.listAll()`; `tree.toString()`

¿Qué tienen en común estos tres problemas?

- Se tiene un objeto padre (contenedor, directorio, árbol)
- La estructura parece repetirse en sus hijos (un gráfico puede tener más gráficos, un directorio puede tener a su vez un directorio, cada nodo del árbol puede ser un nuevo árbol)
- Pero a su vez, tenemos distintos comportamientos.. (un gráfico simple, se dibuja; el gráfico que tiene otros gráficos, dibuja sus gráficos).
- **Necesitamos: Variar el comportamiento reutilizando el comportamiento de elementos más simples.**

Ejemplo

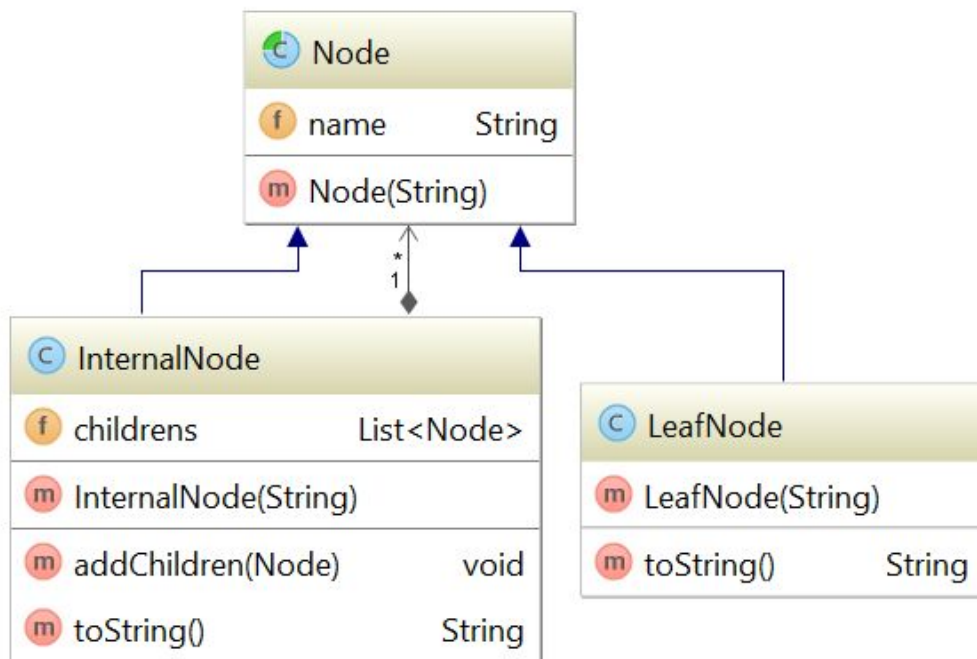
```
String orders = "<orders>";  
  
orders += "<order>";  
  
orders += "<item>";  
  
orders += "</item>";  
  
orders += "</order>";  
  
orders += "</orders>";
```

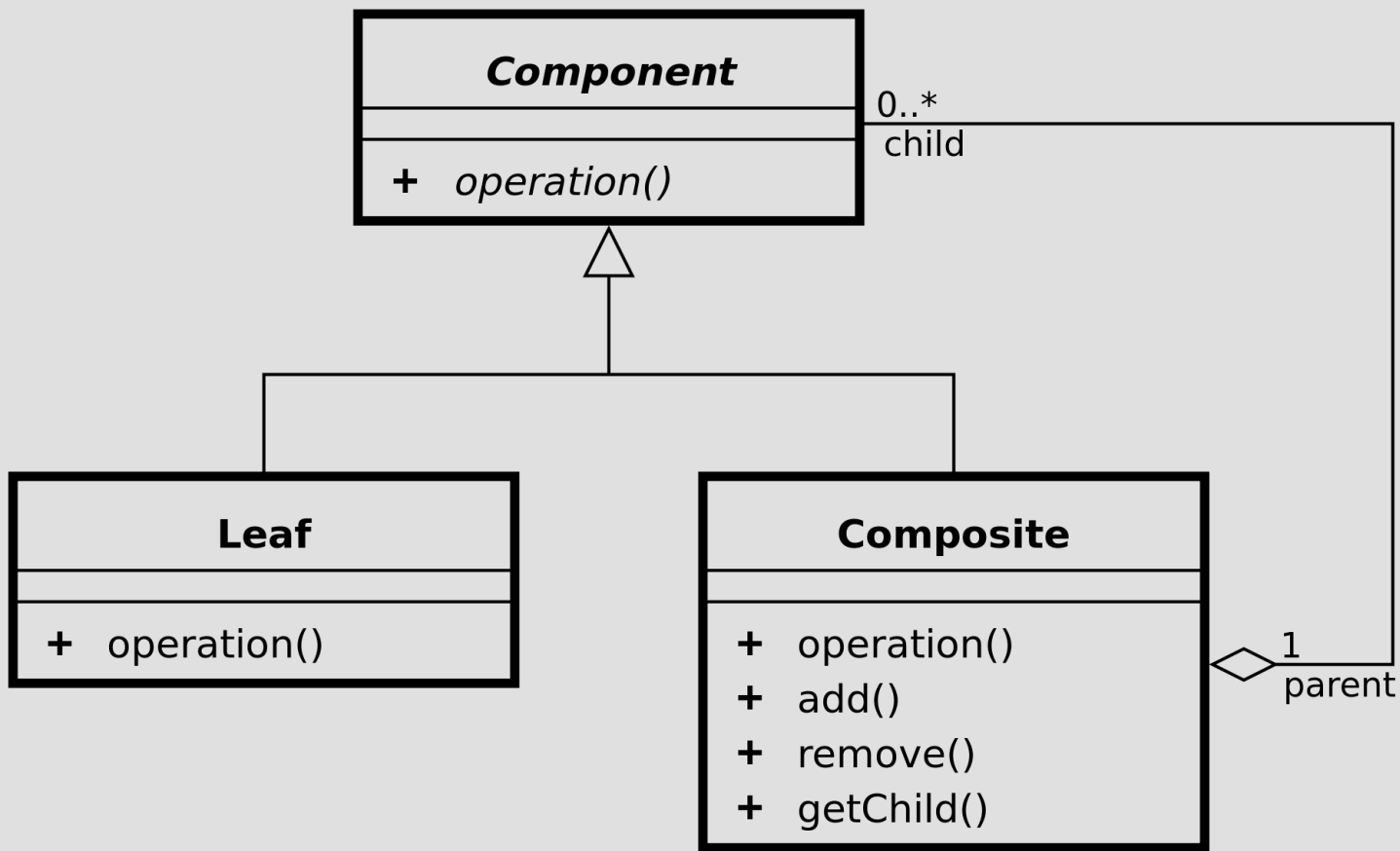
```
InternalNode orders = new InternalNode ("orders");  
  
InternalNode order = new InternalNode ("order");  
  
orders.addChildren(order);  
  
Node item = new LeafNode("item");  
  
order.addChildren(item);  
  
System.out.print(orders.toString());
```

```
public class InternalNode extends Node {
    List<Node> childrens;
    public InternalNode(String name) {
        super(name);
        childrens = new ArrayList<>();
    }
    public void addChildren(Node nodo) {
        this.childrens.add(nodo);
    }
    public String toString() {
        String value = "<" + this.name + ">";
        for(Node children : childrens) {
            value += children.toString();
        }
        value += "</" + this.name + ">";
        return value;
    }
}
```

```
public class LeafNode extends Node {
    public LeafNode(String name) {
        super(name);
    }
    public String toString() {
        return "<" + this.name + "> </" + this.name + ">";
    }
}
```

Composite





?