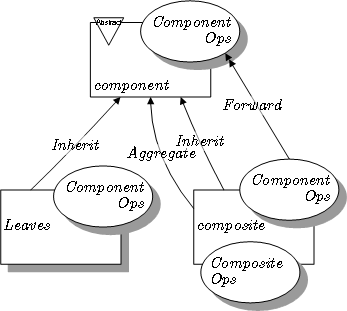
Composite Pattern used to build object hierarchy or group of related objects or part-whole hierarchy.

A system consists of subsystems or components. Components can further be divided into smaller components. Further smaller components can be divided into smaller elements. This is a part-whole hierarchy.



**package** com.sample.designpattern;

**import** java.util.ArrayList;

**import** java.util.List;

**interface** Component {

**public** **void** display();

}

**class** Leaf **implements** Component{

String name;

Leaf(String name) {

**this**.name = name;

}

@Override

**public** **void** display() {

System.***out***.println(name);

}

}

**class** Composite **implements** Component {

String name;

Composite(String name) {

**this**.name = name;

}

List<Component> components = **new** ArrayList<>();;

@Override

**public** **void** display() {

System.***out***.println(name);

print();

}

**public** **void** print() {

**for**(Component c : components) {

c.display();

}

}

**public** **void** add(Component c) {

components.add(c);

}

}

**public** **class** CompositePatternTest {

**public** **static** **void** main(String[] args) {

Composite c1 = **new** Composite("c1");

Leaf l1 = **new** Leaf("Leaf1");

Leaf l11 = **new** Leaf("Leaf11");

Leaf l111 = **new** Leaf("Leaf111");

c1.add(l1);c1.add(l11);c1.add(l111);

c1.display();

Composite c2 = **new** Composite("c2");

Leaf l2 = **new** Leaf("Leaf2");

Leaf l22 = **new** Leaf("Leaf22");

Leaf l222 = **new** Leaf("Leaf222");

c2.add(l2);c2.add(l22);c2.add(l222);

c2.display();

c1.add(c2);

c1.display();

}

}