

Composite Pattern

그릇(복수)과 내용물(단수)을 동일시해서 재귀적인 구조 구축. (ex. Directory Entry)

[Architecture](#), [Modeling](#), [DesignPattern](#), [Structional](#)

Component

```
public abstract class Entry {
    public abstract String getName();
    public abstract int getSize();
    public Entry add(Entry entry) throws Exception {
        throw new Exception();
    }
    public void printList() {
        printList("");
    }
    protected abstract void printList(String prefix);

    @Override
    public String toString() {
        return getName() + " (" + getSize() + ")";
    }
}
```

Leaf

내용물. 내부에는 다른 것을 넣을 수 없음.

```
public class File extends Entry {

    private String name;
    private int size;
    public File(String name, int size) {
        this.name = name;
        this.size = size;
    }
    @Override
    public String getName() {
        return name;
    }

    @Override
    public int getSize() {
        return size;
    }
}
```

```

    }

    @Override
    protected void printList(String prefix) {
        System.out.println(prefix + "/" + this);
    }
}

```

Composite

그릇. Leaf나 Composite 역할을 넣을 수 있음.

```

public class Directory extends Entry {

    private String name;
    private ArrayList<Entry> directory = new ArrayList<>();
    public Directory(String name) {
        this.name = name;
    }
    @Override
    public String getName() {
        return name;
    }

    @Override
    public int getSize() {
        return Stream.of(directory).mapToInt(entry -> entry.size()).sum();
    }
    public Entry add(Entry entry) {
        directory.add(entry);
        return this;
    }

    @Override
    protected void printList(String prefix) {
        System.out.println(prefix + "/" + this);
        directory.forEach(entry -> entry.printList(prefix + "/" + name));
    }
}

```

Client

```
public class Main {  
    public static void main(String[] args) {  
        try {  
            Directory rootdir = new Directory("root");  
            Directory bindir = new Directory("bin");  
            Directory tmpdir = new Directory("tmp");  
            Directory usrdir = new Directory("usr");  
            rootdir.add(bindir);  
            rootdir.add(tmpdir);  
            rootdir.add(usrdir);  
            bindir.add(new File("vi", 10000));  
            bindir.add(new File("latex", 20000));  
            rootdir.printList();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

From:

<http://wiki.ledyx.xyz/> - **Ledyx WIKI**

Permanent link:

http://wiki.ledyx.xyz/doku.php?id=design_pattern:composite_pattern&rev=1509281876

Last update: **2021/02/07 03:15**

