

Composite

```
public interface Document {
    public void display();
    public void add(Document document);
    public void remove(Document document);
}

public class File implements Document {
    private String name;
    public File(String name) {
        this.name = name;
        System.out.println("creating new file....");
    }
    @Override
    public void display() {System.out.println("[File]: " + name);}
    @Override
    public void add(Document document) {System.out.println("error");}
    @Override
    public void remove(Document document) {
        System.out.println("error");
    }
}

public class Folder implements Document {
    private String name;
    private ArrayList<Document> folder = new ArrayList<Document>();
    public Folder(String name) {
        this.name = name;
        System.out.println("creating new folder....");
    }
    @Override
    public void display() {
        System.out.println("[FOLDER]: " + name);

        for(Document document : folder) {
            document.display();
        }
    }
    public void add(Document document) {
        folder.add(document);
        System.out.println("adding file to the folder");
    }
    @Override
    public void remove(Document document) {
        System.out.println("removing file to the folder");
        folder.remove(document);
    }
}

public class Client {
    public static void main(String[] args) {
        Document doc1 = new File("myFile");
        doc1.display();
        Document doc2 = new File("secondFile");
        doc2.display();
        Folder folder = new Folder("myFolder");
        folder.add(doc1);
        folder.add(doc2);
        folder.display();
        folder.remove(doc1);
        folder.display();
    }
}
```

