import java.util.ArrayList;

import java.util.List;

// Component class

abstract class FileComponent {

public void add(FileComponent fileComponent) {

throw new UnsupportedOperationException();

}

public void remove(FileComponent fileComponent) {

throw new UnsupportedOperationException();

}

public FileComponent getChild(int i) {

throw new UnsupportedOperationException();

}

public String getName() {

throw new UnsupportedOperationException();

}

public void display() {

throw new UnsupportedOperationException();

}

}

// Leaf class

class File extends FileComponent {

private String name;

public File(String name) {

this.name = name;

}

public String getName() {

return name;

}

public void display() {

System.out.println("File: " + getName());

}

}

// Composite class

class Directory extends FileComponent {

private String name;

private List<FileComponent> components = new ArrayList<>();

public Directory(String name) {

this.name = name;

}

public void add(FileComponent fileComponent) {

components.add(fileComponent);

}

public void remove(FileComponent fileComponent) {

components.remove(fileComponent);

}

public FileComponent getChild(int i) {

return components.get(i);

}

public String getName() {

return name;

}

public void display() {

System.out.println("Directory: " + getName());

for (FileComponent component : components) {

component.display();

}

}

}

// Client code to test Composite Pattern

public class FileSystemTest {

public static void main(String[] args) {

FileComponent file1 = new File("File1.txt");

FileComponent file2 = new File("File2.txt");

FileComponent file3 = new File("File3.txt");

Directory dir1 = new Directory("Dir1");

Directory dir2 = new Directory("Dir2");

dir1.add(file1);

dir1.add(file2);

dir2.add(file3);

dir2.add(dir1);

dir2.display();

}

}