

Adapter

```
public static void main(String[] args) {
    WightMachineAdapter wightMachineAdapter = new
    WightMachineAdapterImpl(new
    WightMachineForBabies());

    System.out.println(wightMachineAdapter.getWeightInKg());
}

public interface WightMachine{
    public double getWeightInPounds();
}

public class WightMachineForBabies implements WightMachine{

    @Override
    public double getWeightInPounds(){
        return 28;
    }
}

// ----- Adapter

public interface WightMachineAdapter{
    public double getWeightInKg();
}

public class WightMachineAdapterImpl implements WightMachineAdapter{
    WightMachine wightMachine;
    public WightMachineAdapterImpl(WightMachine
    wightMachine){this.wightMachine=wightMachine;}

    @Override
    public double getWeightInKg(){
        double wtInPound = wightMachine.getWeightInPounds();
        double wtInKg= wtInPound*.45;
        return wtInKg;
    }
}
```

Composite

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

    Directory movieDirectory = new Directory("Movie");

    FileSystem border = new File("Border");
    movieDirectory.add(border);

    Directory comedyMovieDirectory = new Directory("ComedyMovie");
    File hulchul = new File("Hulchul");
    comedyMovieDirectory.add(hulchul);
    movieDirectory.add(comedyMovieDirectory);

    movieDirectory.ls();
}

public class Directory implements FileSystem {

    String directoryName;
    List<FileSystem> fileSystemList;

    public Directory(String name){
        this.directoryName = name;
        fileSystemList = new ArrayList<>();
    }

    public void add(FileSystem fileSystemObj) {
        fileSystemList.add(fileSystemObj);
    }

    public void ls(){
        System.out.println("Directory name " + directoryName);

        for(FileSystem fileSystemObj : fileSystemList){
            fileSystemObj.ls();
        }
    }
}

public class File implements FileSystem{
    String fileName;

    public File(String name){
        this.fileName = name;
    }

    public void ls(){
        System.out.println("file name " + fileName);
    }
}

public interface FileSystem {
    public void ls();
}
```

Decorator

```
public static void main(String[] args) {
    BasePizza bp = new ExtraCheese(new FarmHouse());
    System.out.print(bp.cost());
}

public abstract class BasePizza{
    public abstract int cost();
}

public class FarmHouse extends BasePizza{
    @Override
    public int cost(){
        return 200;
    }
}

public class VegDelight extends BasePizza{
    @Override
    public int cost(){
        return 300;
    }
}

public abstract class ToppingDecorator extends BasePizza{

}

public class ExtraCheese extends ToppingDecorator{
    BasePizza bp;
    public ExtraCheese(BasePizza bp){
        this.bp = bp;
    }
    @Override
    public int cost(){
        return this.bp.cost() + 50;
    }
}

public class Mushroom extends ToppingDecorator{
    BasePizza bp;
    public ExtraCheese(BasePizza bp){
        this.bp = bp;
    }
    @Override
    public int cost(){
        return this.bp.cost() + 40;
    }
}
```