interface Car {

    public void assemble();

}

class BasicCar implements Car {

    @Override

    public void assemble() {

        System.out.print("Basic Car.");

    }

}

class CarDecorator implements Car {

    protected Car car;

    public CarDecorator(Car c) {

        this.car = c;

    }

    @Override

    public void assemble() {

        this.car.assemble();

    }

}

class SportsCar extends CarDecorator {

    public SportsCar(Car c) {

        super(c);

    }

    @Override

    public void assemble() {

        super.assemble();

        System.out.println(" Adding features of Sports Car.");

    }

}

class LuxuryCar extends CarDecorator {

    public LuxuryCar(Car c) {

        super(c);

    }

    @Override

    public void assemble() {

        super.assemble();

        System.out.println(" Adding features of Luxury Car.");

    }

}

class Demo {

    public static void main(String[] args) {

        BasicCar c1=new BasicCar();

      SportsCar sport=new SportsCar(c1);

      sport.assemble();

      BasicCar c2=new BasicCar();

      LuxuryCar luxury=new LuxuryCar(c2);

      luxury.assemble();

    }

}

import java.sql.Connection;

class MySQLHelper {

    public static Connection getMySqlDBConnection() {

        // get MySql DB connection using connection parameters

        return null;

    }

}

class OracleHelper {

    public static Connection getOracleDBConnection() {

        // get Oracle DB connection using connection parameters

        return null;

    }

}

class HelperFacade {

    static Connection getConnection(String type) {

        Connection con = null;

        if (type.equals("oracle")) {

            con = OracleHelper.getOracleDBConnection();

        } else if (type.equals("mysql")) {

            con = MySQLHelper.getMySqlDBConnection();

        }

    }

public class Demo1 {

    public static void main(String[] args) {

        Connection con;

con=HelperFacade.getConnection("oracle");

    }

}

class Request {

    private String value;

    private String description;

    public Request(String description, String value) {

        this.description = description;

        this.value = value;

    }

    public String getValue() {

        return value;

    }

    public String getDescription() {

        return description;

    }

}

abstract class Handler {

    protected Handler m\_successor;

    public void setSuccessor(Handler successor) {

        m\_successor = successor;

    }

    public abstract void handleRequest(Request request);

}

class ETC extends Handler {

    public void handleRequest(Request request) {

        if (request.getValue().equals("ETC")) { // if request is eligible handle it

            System.out.println(" handled by ETC:");

            System.out.println("\tRequest : " + request.getDescription()

                    + request.getValue());

        } else {

            m\_successor.handleRequest(request);

        }

    }

}

class CSE extends Handler {

    public void handleRequest(Request request) {

        if (request.getValue().equals("CSE")) { // if request is eligible handle it

            System.out.println(" handled by CSE:");

            System.out.println("\tRequest : " + request.getDescription()

                    + request.getValue());

        } else {

            m\_successor.handleRequest(request);

        }

    }

}

public class Demo2 {

    public static void main(String[] args) {

        Request r1=new Request("For CSE DEPT","CSE");

    Handler h1=new ETC();

    Handler h2=new CSE();

    h1.setSuccessor(h2);

    h1.handleRequest(r1);

    }

}