**Exercise 5: Implementing the Decorator Pattern**

interface Notifier {

    void send(String message);

}

class EmailNotifier implements Notifier {

    @Override

    public void send(String message) {

        System.out.println("Email sent: " + message);

    }

}

abstract class NotifierDecorator implements Notifier {

    protected Notifier wrappedNotifier;

    public NotifierDecorator(Notifier notifier) {

        this.wrappedNotifier = notifier;

    }

    @Override

    public void send(String message) {

        wrappedNotifier.send(message);

    }

}

class SMSNotifierDecorator extends NotifierDecorator {

    public SMSNotifierDecorator(Notifier notifier) {

        super(notifier);

    }

    @Override

    public void send(String message) {

        super.send(message);

        sendSMS(message);

    }

    private void sendSMS(String message) {

        System.out.println("SMS sent: " + message);

    }

}

class SlackNotifierDecorator extends NotifierDecorator {

    public SlackNotifierDecorator(Notifier notifier) {

        super(notifier);

    }

    @Override

    public void send(String message) {

        super.send(message);

        sendSlack(message);

    }

    private void sendSlack(String message) {

        System.out.println("Slack message sent: " + message);

    }

}

public class Main {

    public static void main(String[] args) {

        System.out.println("Single Channel:");

        Notifier email = new EmailNotifier();

        email.send("Server is down.");

        System.out.println("\nEmail + SMS:");

        Notifier emailSms = new SMSNotifierDecorator(new EmailNotifier());

        emailSms.send("High CPU usage alert!");

        System.out.println("\nEmail + SMS + Slack:");

        Notifier multiChannel = new SlackNotifierDecorator(

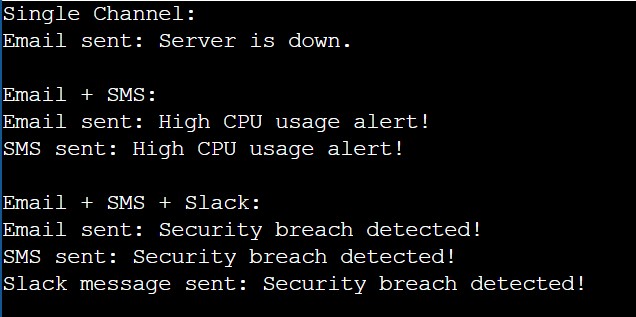
                                    new SMSNotifierDecorator(

                                        new EmailNotifier()));

        multiChannel.send("Security breach detected!");

    }

}

Output: