**DESIGN PATTERNS AND PRINCIPLES**

**Exercise 4: Implementing the Adapter Pattern**

**Scenario:**

You are developing a payment processing system that needs to integrate with multiple third-party payment gateways with different interfaces. Use the Adapter Pattern to achieve this.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **AdapterPatternExample**.
2. **Define Target Interface:**
   * Create an interface **PaymentProcessor** with methods like **processPayment()**.
3. **Implement Adaptee Classes:**
   * Create classes for different payment gateways with their own methods.
4. **Implement the Adapter Class:**
   * Create an adapter class for each payment gateway that implements PaymentProcessor and translates the calls to the gateway-specific methods.
5. **Test the Adapter Implementation:**
   * Create a test class to demonstrate the use of different payment gateways through the adapter.

**DESCRIPTION:**

This program uses the Adapter Pattern to integrate multiple third-party payment gateways under a common interface, PaymentProcessor.

PayPalGateway and StripeGateway are adaptee classes with their own methods.  
PayPalAdapter and StripeAdapter are adapters that convert these methods to the standard executeTransaction() defined in the PaymentProcessor interface.

The client (AdapterPatternExample) interacts with all gateways uniformly, allowing easy integration of new payment systems without changing existing code.

**PROGRAM:**

public class AdapterPatternExample {

public static void main(String[] args) {

PaymentProcessor processor1 = new PayPalAdapter(new PayPalGateway());

processor1.processPayment(1500);

PaymentProcessor processor2 = new StripeAdapter(new StripeGateway());

processor2.processPayment(2500);

}

}

interface PaymentProcessor {

void processPayment(double amount);

}

class PayPalGateway {

public void makePayment(double amount) {

System.out.println("Paid ₹" + amount + " using PayPal.");

}

}

class StripeGateway {

public void chargeCard(double amount) {

System.out.println("Charged ₹" + amount + " using Stripe.");

}

}

class PayPalAdapter implements PaymentProcessor {

private PayPalGateway paypal;

public PayPalAdapter(PayPalGateway paypal) {

this.paypal = paypal;

}

public void processPayment(double amount) {

paypal.makePayment(amount);

}

}

class StripeAdapter implements PaymentProcessor {

private StripeGateway stripe;

public StripeAdapter(StripeGateway stripe) {

this.stripe = stripe;

}

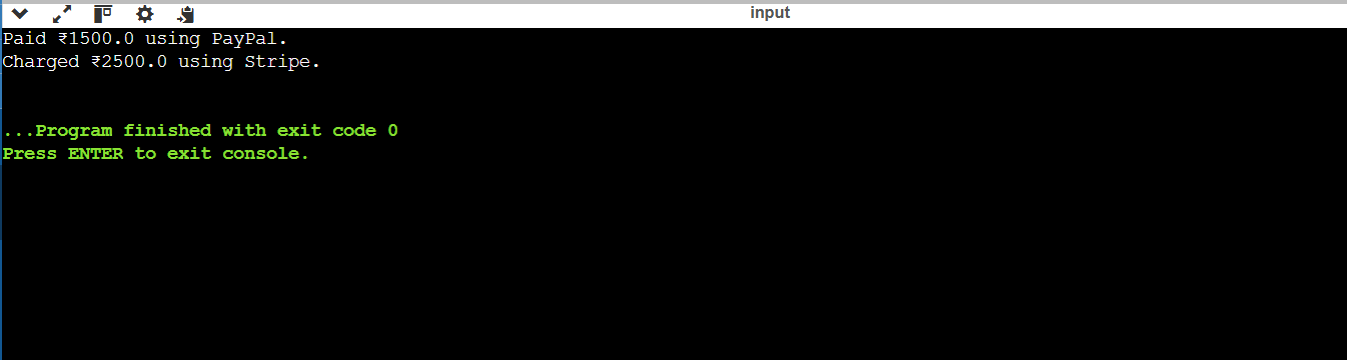
public void processPayment(double amount) {

stripe.chargeCard(amount);

}

}

**OUTPUT:**

****