**Exercise 4: Implementing the Adapter Pattern**

public class AdapterPatternExample {

interface PaymentProcessor {

void processPayment(double amount);

}

static class PayPalGateway {

public void sendPayment(double amount) {

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

}

}

static class StripeGateway {

public void makePayment(double amount) {

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

}

}

static class RazorpayGateway {

public void transferAmount(double amount) {

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

}

}

static class PayPalAdapter implements PaymentProcessor {

private PayPalGateway payPal;

public PayPalAdapter(PayPalGateway payPal) {

this.payPal = payPal;

}

public void processPayment(double amount) {

payPal.sendPayment(amount);

}

}

static class StripeAdapter implements PaymentProcessor {

private StripeGateway stripe;

public StripeAdapter(StripeGateway stripe) {

this.stripe = stripe;

}

public void processPayment(double amount) {

stripe.makePayment(amount);

}

}

static class RazorpayAdapter implements PaymentProcessor {

private RazorpayGateway razorpay;

public RazorpayAdapter(RazorpayGateway razorpay) {

this.razorpay = razorpay;

}

public void processPayment(double amount) {

razorpay.transferAmount(amount);

}

}

public static void main(String[] args) {

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

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

PaymentProcessor razorpay = new RazorpayAdapter(new RazorpayGateway());

paypal.processPayment(2500);

stripe.processPayment(3800);

razorpay.processPayment(1500);

}

}

OUTPUT:

A screenshot of a computer program

AI-generated content may be incorrect.