**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.

//interface paymentprocessor

public interface PaymentProcessor {

void processPayment(String amount);

}

public class PaypalGateway {

public void sendPayment(String amount) {

System.out.println("Processing Paypal payment of $" + amount);

}

}

public class RazorpayGateway {

public void makePayment(double rupees) {

System.out.println("Processing Razorpay payment of ₹" + rupeest);

}

}

public class PaypalAdapter implements PaymentProcessor {

private PaypalGateway payPalGateway;

public PaypalAdapter(PaypalGateway payPalGateway) {

this.paypalGateway = payPalGateway;

}

public void processPayment(String amount) {

payPalGateway.sendPayment(amount);

}

}

public class RazorpayAdapter implements PaymentProcessor {

private RazorpayGateway razorpayGateway;

public RazorpayAdapter(RazorpayGateway razorpayGateway) {

this.razorpayGateway = razorpayGateway;

}

@Override

public void processPayment(String amount) {

// string to double

double amountDouble = Double.parseDouble(amount);

razorpayGateway.makePayment(amountDouble);

}

}

public class Test {

public static void main(String[] args) {

// Paypal

PaypalGateway payPal = new PaypalGateway();

PaymentProcessor paypalAdapter = new PayPalAdapter(payPal);

paypalAdapter.processPayment("100.00");

// Razorpay

RazorpayGateway razorpay = new RazorpayGateway();

PaymentProcessor razorpayAdapter = new RazorpayAdapter(razorpay);

razorpayAdapter.processPayment("750.50");

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.