Design patterns and principles

SuperSet ID:6412063

Exercise 4: Implementing the Adapter Pattern

Code:

public interface PaymentProcessor {

void processPayment(double amount);

}

public class PaypalGateway {

public void makePaypalPayment(double amount) {

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

}

}

public class StripeGateway {

public void makeStripeCharge(double amount) {

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

}

}

public class PaypalAdapter implements PaymentProcessor {

private PaypalGateway paypal;

public PaypalAdapter(PaypalGateway paypal) {

this.paypal = paypal;

}

public void processPayment(double amount) {

paypal.makePaypalPayment(amount);

}

}

public class StripeAdapter implements PaymentProcessor {

private StripeGateway stripe;

public StripeAdapter(StripeGateway stripe) {

this.stripe = stripe;

}

public void processPayment(double amount) {

stripe.makeStripeCharge(amount);

}

}

public class PaymentTest {

public static void main(String[] args) {

// Using PayPal through adapter

PaymentProcessor paypalProcessor = new PaypalAdapter(new PaypalGateway());

paypalProcessor.processPayment(100.00);

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

stripeProcessor.processPayment(250.50);

}

}

Output:

A close-up of a text

Description automatically generated