**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.

**Main.java :-**

public class Main {

    public static void main(String[] args) {

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

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

        System.out.println("Processing different payment gateways:\n");

        stripe.processPayment(499.99);

        paypal.processPayment(250.50);

        System.out.println("\n--- Payment Summary Complete ---");

    }

}

**PaymentProcessor.java:-**

public interface PaymentProcessor {

    void processPayment(double amount);

}

**PayPalAdapter.java:-**

public class PayPalAdapter implements PaymentProcessor {

private PayPalGateway paypal;

public PayPalAdapter(PayPalGateway paypal) {

this.paypal = paypal;

}

@Override

public void processPayment(double amount) {

paypal.payUsingPayPal(amount);

}

}

**PayPalGateway.java:-**

public class PayPalGateway {

    public void payUsingPayPal(double amountInRupees) {

        System.out.println(" Paid INR" + amountInRupees + " via PayPal.");

    }

}

**StripeAdapter.java:-**

public class StripeAdapter implements PaymentProcessor {

    private StripeGateway stripe;

    public StripeAdapter(StripeGateway stripe) {

        this.stripe = stripe;

    }

    @Override

    public void processPayment(double amount) {

        stripe.makeStripePayment(amount);

    } }

**StripeGateway.java:-**

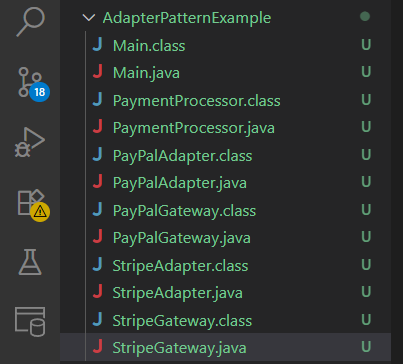
public class StripeGateway {

public void makeStripePayment(double amount) {

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

}

}



**Output :-**

