**Exercise 4: Implementing the Adapter Pattern**

**Program:**

interface PaymentProcessor {

    void processPayment(double amount);

}

class PaypalGateWay{

    public void paypalPayment(double dollors){

        System.out.println("paypal payment processing of Dollors"+dollors);

    }

}

class PaytmGateWay{

    public void PaytmPayment(double rupees){

        System.out.println("paypal payment processing of Rupees"+rupees);

    }

}

class AmazonPayGateWay{

    public void AmazonPayment(double amazonCredits){

        System.out.println("paypal payment processing of "+amazonCredits);

    }

}

class PaypalAdaptor implements PaymentProcessor{

    PaypalGateWay paypal;

    PaypalAdaptor(PaypalGateWay paypal){

        this.paypal = paypal;

    }

    public void processPayment(double amount){

        paypal.paypalPayment(amount);

    }

}

class PaytmAdaptor implements PaymentProcessor{

    PaytmGateWay paytm;

    PaytmAdaptor(PaytmGateWay paytm){

        this.paytm = paytm;

    }

    public void processPayment(double amount){

        paytm.PaytmPayment(amount);

    }

}

class AmazonPayAdaptor implements PaymentProcessor{

    AmazonPayGateWay amazon;

    AmazonPayAdaptor(AmazonPayGateWay amazon){

        this.amazon = amazon;

    }

    public void processPayment(double amount){

        amazon.AmazonPayment(amount);

    }

}

public class Main {

    public static void main(String[] args) {

        PaymentProcessor paypal = new PaypalAdaptor(new PaypalGateWay());

        paypal.processPayment(123.50);

        PaymentProcessor paytm = new PaytmAdaptor(new PaytmGateWay());

        paytm.processPayment(345.80);

        PaymentProcessor amazon = new AmazonPayAdaptor(new AmazonPayGateWay());

        amazon.processPayment(987.95);

    }

}

