**Implementing the Adapter Pattern**

**PaymentProcessor.java**

package AdapterPatternExample.adapter;

public interface PaymentProcessor {

void processPayment(double amount);

}

**PayPal.java**

package AdapterPatternExample.adapter;

public class PayPal {

public void sendPayment(double amount) {

System.out.println("PayPal processed payment of $" + amount)

}

}

**Stripe.java**

package AdapterPatternExample.adapter;

public class Stripe {

public void makePayment(double amount) {

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

}

}

**PayPalAdapter.java**

package AdapterPatternExample.adapter;

public class PayPalAdapter implements PaymentProcessor {

private PayPal payPal;

public PayPalAdapter(PayPal payPal) {

this.payPal = payPal;

}

@Override

public void processPayment(double amount) {

payPal.sendPayment(amount);

}

**StripeAdapter.java**

package AdapterPatternExample.adapter;

public class StripeAdapter implements PaymentProcessor {

private Stripe stripe;

public StripeAdapter(Stripe stripe) {

this.stripe = stripe;

}

@Override

public void processPayment(double amount) {

stripe.makePayment(amount);

}

**Main.java**

package AdapterPatternExample.adapter;

public class Main {

Public static void main(String[] args) {

PaymentProcessor paypalProcessor = new PayPalAdapter(new PayPal());

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

paypalProcessor.processPayment(100.0);

stripeProcessor.processPayment(250.5);

}

}

**Output**

