**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.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named **AdapterPatternExample**.
2. **Define Target Interface:**
   * Create an interface **PaymentProcessor** with methods like **processPayment()**.
3. **Implement Adaptee Classes:**
   * Create classes for different payment gateways with their own methods.
4. **Implement the Adapter Class:**
   * Create an adapter class for each payment gateway that implements PaymentProcessor and translates the calls to the gateway-specific methods.
5. **Test the Adapter Implementation:**
   * Create a test class to demonstrate the use of different payment gateways through the adapter.

**Solution:**using System;

public interface IPaymentProcessor

{

void ProcessPayment(decimal amount);

}

public class StripeGateway

{

public void MakeStripePayment(decimal amount)

{

Console.WriteLine($"Paid {amount} using Stripe");

}

}

public class PayPalGateway

{

public void SendPayPalPayment(decimal amount)

{

Console.WriteLine($"Paid {amount} using PayPal");

}

}

public class StripeAdapter : IPaymentProcessor

{

private readonly StripeGateway stripe;

public StripeAdapter(StripeGateway stripe)

{

this.stripe = stripe;

}

public void ProcessPayment(decimal amount)

{

stripe.MakeStripePayment(amount);

}

}

public class PayPalAdapter : IPaymentProcessor

{

private readonly PayPalGateway paypal;

public PayPalAdapter(PayPalGateway paypal)

{

this.paypal = paypal;

}

public void ProcessPayment(decimal amount)

{

paypal.SendPayPalPayment(amount);

}

}

public class Program

{

public static void Main()

{

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

stripeProcessor.ProcessPayment(100.0m);

IPaymentProcessor paypalProcessor = new PayPalAdapter(new PayPalGateway());

paypalProcessor.ProcessPayment(200.0m);

}

}

