**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.

**CODE:**

interface IPaymentProcessor

{

void ProcessPayment(decimal amount);

}

class PayPal

{

public void SendPayment(decimal amount) => Console.WriteLine($"Paid {amount} via PayPal");

}

class PayPalAdapter : IPaymentProcessor

{

private PayPal payPal = new PayPal();

public void ProcessPayment(decimal amount) => payPal.SendPayment(amount);

}

class Program

{

static void Main()

{

IPaymentProcessor processor = new PayPalAdapter();

processor.ProcessPayment(100.50m);

}

}

**OUTPUT:**

