**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.

## Answer:-

GPayAdapter.java

public class GPayAdapter implements PaymentProcessor {

    GPayGateway gpay;

    public GPayAdapter(GPayGateway gpay) {

        this.gpay = gpay;

    }

    @Override

    public void processPayment(double amt) {

        gpay.gPaySend(amt);

    }

}

GPayGateway.java

public class GPayGateway {

     public void gPaySend(double amt) {

        System.out.println("Sending payment of $" + amt);

    }

}

Main.java

public class Main {

    public static void main(String[] args) {

        GPayGateway gpay = new GPayGateway();

        PayTmGateway paytm = new PayTmGateway();

        PhonePayGateway pp = new PhonePayGateway();

        PaymentProcessor gpayP = new GPayAdapter(gpay);

        PaymentProcessor paytmP = new PayTmAdapter(paytm);

        PaymentProcessor phonePayP = new PhonePayAdapter(pp);

        System.out.print("Google Pay: ");

        gpayP.processPayment( 50.0 );

        System.out.print("\nPayTm: ");

        paytmP.processPayment( 75.5 );

        System.out.print("\nPhonePe: ");

        phonePayP.processPayment( 100.25 );

    }

}

PaymentProcessor.java

public interface PaymentProcessor {

    void processPayment(double amount);

}

PayTmAdapter.java

public class PayTmAdapter implements PaymentProcessor{

    PayTmGateway paytm;

    public PayTmAdapter(PayTmGateway paytm) {

        this.paytm = paytm;

    }

    @Override

    public void processPayment(double amt) {

        paytm.paytmSend(amt);

    }

}

PayTmGateway.java

public class PayTmGateway {

    void paytmSend(double amt) {

        System.out.println("Sending payment of $" + amt);

    }

}

public class PhonePayAdapter implements PaymentProcessor{

    PhonePayGateway phonepay;

    public PhonePayAdapter(PhonePayGateway phonepay) {

        this.phonepay = phonepay;

    }

    @Override

    public void processPayment(double amt) {

        phonepay.phonePaySend(amt);

    }

}

PhonePayGateway.java

public class PhonePayGateway {

    public void phonePaySend(double amt) {

        System.out.println("Sending payment of $" + amt);

    }

}

Output:

