Step 1: Create a New Java Project

Create a new Java project named AdapterPatternExample.

Step 2: Define Target Interface

Create an interface PaymentProcessor with methods like processPayment():

public interface PaymentProcessor {

void processPayment(String paymentMethod, double amount);

}

Step 3: Implement Adaptee Classes

Create classes for different payment gateways with their own methods:

public class PayPalGateway {

public void payWithPayPal(String paymentMethod, double amount) {

System.out.println("Processing payment of $" + amount + " using PayPal with " + paymentMethod);

}

}

public class StripeGateway {

public void chargeWithStripe(String paymentMethod, double amount) {

System.out.println("Processing payment of $" + amount + " using Stripe with " + paymentMethod);

}

}

public class BankTransferGateway {

public void transferWithBank(String paymentMethod, double amount) {

System.out.println("Processing payment of $" + amount + " using Bank Transfer with " + paymentMethod);

}

}

Step 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:

public class PayPalAdapter implements PaymentProcessor {

private PayPalGateway payPalGateway;

public PayPalAdapter(PayPalGateway payPalGateway) {

this.payPalGateway = payPalGateway;

}

@Override

public void processPayment(String paymentMethod, double amount) {

payPalGateway.payWithPayPal(paymentMethod, amount);

}

}

public class StripeAdapter implements PaymentProcessor {

private StripeGateway stripeGateway;

public StripeAdapter(StripeGateway stripeGateway) {

this.stripeGateway = stripeGateway;

}

@Override

public void processPayment(String paymentMethod, double amount) {

stripeGateway.chargeWithStripe(paymentMethod, amount);

}

}

public class BankTransferAdapter implements PaymentProcessor {

private BankTransferGateway bankTransferGateway;

public BankTransferAdapter(BankTransferGateway bankTransferGateway) {

this.bankTransferGateway = bankTransferGateway;

}

@Override

public void processPayment(String paymentMethod, double amount) {

bankTransferGateway.transferWithBank(paymentMethod, amount);

}

}

Step 5: Test the Adapter Implementation

Create a test class to demonstrate the use of different payment gateways through the adapter:

public class AdapterPatternTest {

public static void main(String[] args) {

PayPalGateway payPalGateway = new PayPalGateway();

PayPalAdapter payPalAdapter = new PayPalAdapter(payPalGateway);

payPalAdapter.processPayment("Credit Card", 100.0);

StripeGateway stripeGateway = new StripeGateway();

StripeAdapter stripeAdapter = new StripeAdapter(stripeGateway);

stripeAdapter.processPayment("Debit Card", 50.0);

BankTransferGateway bankTransferGateway = new BankTransferGateway();

BankTransferAdapter bankTransferAdapter = new BankTransferAdapter(bankTransferGateway);

bankTransferAdapter.processPayment("Bank Account", 200.0);

}

}

OUTPUT

Processing payment of $100.0 using PayPal with Credit Card

Processing payment of $50.0 using Stripe with Debit Card

Processing payment of $200.0 using Bank Transfer with Bank Account