**STRATEGY PATTERN**

interface PaymentStrategy {

void pay(double amount);

}

class CreditCardPayment implements PaymentStrategy {

private String cardNumber;

private String cardHolder;

public CreditCardPayment(String cardNumber, String cardHolder) {

this.cardNumber = cardNumber;

this.cardHolder = cardHolder;

}

@Override

public void pay(double amount) {

System.out.println("Paid $" + amount + " using Credit Card.");

System.out.println("Card Holder: " + cardHolder + ", Card Number: \*\*\*\*" + cardNumber.substring(cardNumber.length() - 4));

}

}

class PayPalPayment implements PaymentStrategy {

private String email;

public PayPalPayment(String email) {

this.email = email;

}

@Override

public void pay(double amount) {

System.out.println("Paid $" + amount + " using PayPal.");

System.out.println("PayPal Account: " + email);

}

}

class PaymentContext {

private PaymentStrategy strategy;

public void setPaymentStrategy(PaymentStrategy strategy) {

this.strategy = strategy;

}

public void processPayment(double amount) {

if (strategy == null) {

System.out.println("No payment strategy selected.");

} else {

strategy.pay(amount);

}

}

}

public class Main {

public static void main(String[] args) {

PaymentContext context = new PaymentContext();

PaymentStrategy creditCard = new CreditCardPayment("1234567890123456", "Alice Johnson");

context.setPaymentStrategy(creditCard);

context.processPayment(250.00);

System.out.println();

PaymentStrategy paypal = new PayPalPayment("abc@gmail.com");

context.setPaymentStrategy(paypal);

context.processPayment(100.00);

}

}

