Design patterns and principles

SuperSet ID:6412063

Exercise 8: Implementing the Strategy Pattern

Code:

public interface PaymentStrategy {

void pay(double amount);

}

public class CreditCardPayment implements PaymentStrategy {

private String cardNumber;

public CreditCardPayment(String cardNumber) {

this.cardNumber = cardNumber;

}

public void pay(double amount) {

System.out.println("Paid ₹" + amount + " using Credit Card (Card Number: " + cardNumber + ")");

}

}

public class PayPalPayment implements PaymentStrategy {

private String email;

public PayPalPayment(String email) {

this.email = email;

}

public void pay(double amount) {

System.out.println("Paid ₹" + amount + " using PayPal (Email: " + email + ")");

}

}

public class PaymentContext {

private PaymentStrategy paymentStrategy;

public void setPaymentStrategy(PaymentStrategy paymentStrategy) {

this.paymentStrategy = paymentStrategy;

}

public void makePayment(double amount) {

if (paymentStrategy == null) {

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

} else {

paymentStrategy.pay(amount);

}

}

}

public class PaymentSystemTest {

public static void main(String[] args) {

PaymentContext context = new PaymentContext();

context.setPaymentStrategy(new CreditCardPayment("1234-5678-9012-3456"));

context.makePayment(500.00);

context.setPaymentStrategy(new PayPalPayment("user@example.com"));

context.makePayment(750.00);

}

}

Output:

A close up of text

Description automatically generated