**DESIGN PATTERNS AND PRINCIPLES**

**Exercise 8: Implementing the Strategy Pattern**

**Scenario:**

You are developing a payment system where different payment methods (e.g., Credit Card, PayPal) can be selected at runtime. Use the Strategy Pattern to achieve this.

**Steps:**

1. **Create a New Java Project:**
   * Create a new Java project named StrategyPatternExample.
2. **Define Strategy Interface:**
   * Create an interface PaymentStrategy with a method pay().
3. **Implement Concrete Strategies:**
   * Create classes CreditCardPayment, PayPalPayment that implement PaymentStrategy.
4. **Implement Context Class:**
   * Create a class PaymentContext that holds a reference to PaymentStrategy and a method to execute the strategy.
5. **Test the Strategy Implementation:**
   * Create a test class to demonstrate selecting and using different payment strategies.

**DESCRIPTION:**

This program demonstrates the Strategy Pattern to switch between payment methods at runtime.

* PaymentStrategy defines the pay() method.
* CreditCardPayment and PayPalPayment are different payment strategies.
* PaymentContext uses a selected strategy to process the payment.
* The client chooses and changes the strategy dynamically.

**PROGRAM:**

public class StrategyPatternExample {

public static void main(String[] args) {

PaymentContext context = new PaymentContext();

context.setStrategy(new CreditCardPayment("1234-5678-9012-3456", "Keerthi", "123"));

context.processPayment(1500.00);

System.out.println("---");

context.setStrategy(new PayPalPayment("keerthi@example.com"));

context.processPayment(2500.00);

}

}

interface PaymentStrategy {

void pay(double amount);

}

class CreditCardPayment implements PaymentStrategy {

private String cardNumber;

private String cardHolder;

private String cvv;

public CreditCardPayment(String cardNumber, String cardHolder, String cvv) {

this.cardNumber = cardNumber;

this.cardHolder = cardHolder;

this.cvv = cvv;

}

public void pay(double amount) {

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

}

}

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 account: " + email);

}

}

class PaymentContext {

private PaymentStrategy strategy;

public void setStrategy(PaymentStrategy strategy) {

this.strategy = strategy;

}

public void processPayment(double amount) {

if (strategy != null) {

strategy.pay(amount);

} else {

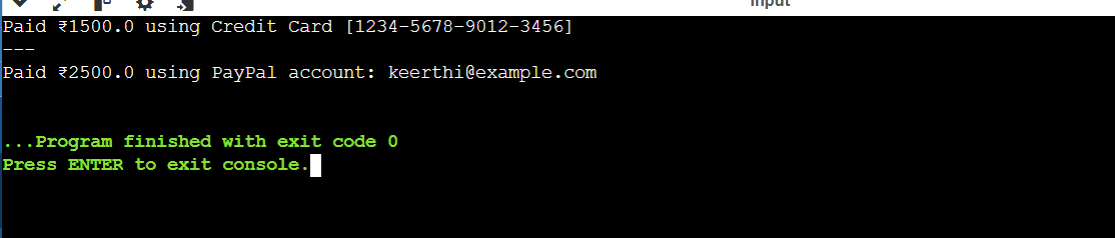
System.out.println("No payment strategy selected.");

}

}

}

**OUTPUT:**

****