**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.

using System;

public interface IPaymentStrategy

{

void Pay(decimal amount);

}

public class CreditCardPayment : IPaymentStrategy

{

public void Pay(decimal amount)

{

Console.WriteLine($"Paid {amount} using Credit Card");

}

}

public class PayPalPayment : IPaymentStrategy

{

public void Pay(decimal amount)

{

Console.WriteLine($"Paid {amount} using PayPal");

}

}

public class PaymentContext

{

private IPaymentStrategy strategy;

public PaymentContext(IPaymentStrategy strategy)

{

this.strategy = strategy;

}

public void SetStrategy(IPaymentStrategy strategy)

{

this.strategy = strategy;

}

public void ExecutePayment(decimal amount)

{

strategy.Pay(amount);

}

}

public class Program

{

public static void Main()

{

PaymentContext context = new PaymentContext(new CreditCardPayment());

context.ExecutePayment(250.00m);

context.SetStrategy(new PayPalPayment());

context.ExecutePayment(125.50m);

}

}

