

Namespace Comportamiento.Domain

Classes

[CashPaymentStrategy](#)

Implementación de la estrategia de pago en efectivo. Esta clase concreta realiza el pago utilizando el método de pago en efectivo.

[CreditCardPaymentStrategy](#)

Implementación de la estrategia de pago con tarjeta de crédito. Esta clase concreta realiza el pago utilizando el método de pago con tarjeta de crédito.

[DebitCardPaymentStrategy](#)

Implementación de la estrategia de pago con tarjeta de débito. Esta clase concreta realiza el pago utilizando el método de pago con tarjeta de débito.

[PaymentContext](#)

Clase que representa el contexto en el que se utilizará una estrategia de pago. Esta clase delega el procesamiento del pago a la estrategia de pago proporcionada.

Interfaces

[IPaymentStrategy](#)

Interfaz que define el contrato para las estrategias de pago. Cualquier clase que implemente esta interfaz debe proporcionar una implementación para el método [Pay\(double\)](#) que permite realizar un pago.

Class CashPaymentStrategy

Namespace: [Comportamiento.Domain](#)

Assembly: Comportamiento.Domain.dll

Implementación de la estrategia de pago en efectivo. Esta clase concreta realiza el pago utilizando el método de pago en efectivo.

```
public class CashPaymentStrategy : IPaymentStrategy
```

Inheritance

[object](#) ← CashPaymentStrategy

Implements

[IPaymentStrategy](#)

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

Pay(double)

Realiza el pago utilizando efectivo. Este método imprime en la consola el monto del pago realizado en efectivo.

```
public void Pay(double amount)
```

Parameters

amount [double](#)

El monto que el cliente desea pagar.

Class CreditCardPaymentStrategy

Namespace: [Comportamiento.Domain](#)

Assembly: Comportamiento.Domain.dll

Implementación de la estrategia de pago con tarjeta de crédito. Esta clase concreta realiza el pago utilizando el método de pago con tarjeta de crédito.

```
public class CreditCardPaymentStrategy : IPaymentStrategy
```

Inheritance

[object](#) ← CreditCardPaymentStrategy

Implements

[IPaymentStrategy](#)

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

Pay(double)

Realiza el pago utilizando tarjeta de crédito. Este método imprime en la consola el monto del pago realizado con tarjeta de crédito.

```
public void Pay(double amount)
```

Parameters

amount [double](#)

El monto que el cliente desea pagar.

Class DebitCardPaymentStrategy

Namespace: [Comportamiento.Domain](#)

Assembly: Comportamiento.Domain.dll

Implementación de la estrategia de pago con tarjeta de débito. Esta clase concreta realiza el pago utilizando el método de pago con tarjeta de débito.

```
public class DebitCardPaymentStrategy : IPaymentStrategy
```

Inheritance

[object](#) ← DebitCardPaymentStrategy

Implements

[IPaymentStrategy](#)

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

Pay(double)

Realiza el pago utilizando tarjeta de débito. Este método imprime en la consola el monto del pago realizado con tarjeta de débito.

```
public void Pay(double amount)
```

Parameters

amount [double](#)

El monto que el cliente desea pagar.

Interface IPaymentStrategy

Namespace: [Comportamiento.Domain](#)

Assembly: Comportamiento.Domain.dll

Interfaz que define el contrato para las estrategias de pago. Cualquier clase que implemente esta interfaz debe proporcionar una implementación para el método [Pay\(double\)](#) que permite realizar un pago.

```
public interface IPaymentStrategy
```

Methods

Pay(double)

Método que define cómo se realiza un pago utilizando una estrategia concreta. Este método debe ser implementado por todas las clases que implementen la interfaz [IPaymentStrategy](#).

```
void Pay(double amount)
```

Parameters

amount [double](#) 

El monto que el cliente desea pagar.

Class PaymentContext

Namespace: [Comportamiento.Domain](#)

Assembly: Comportamiento.Domain.dll

Clase que representa el contexto en el que se utilizará una estrategia de pago. Esta clase delega el procesamiento del pago a la estrategia de pago proporcionada.

```
public class PaymentContext
```

Inheritance

[object](#) ← PaymentContext

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

ProcessPayment(double)

Procesa el pago utilizando la estrategia de pago previamente seleccionada. Si no se ha establecido una estrategia de pago, se lanza una excepción.

```
public void ProcessPayment(double amount)
```

Parameters

amount [double](#)

El monto que se desea pagar.

Exceptions

[ArgumentException](#)

Se lanza si la estrategia de pago es nula.

SetPaymentStrategy(IPaymentStrategy)

Establece la estrategia de pago a utilizar en el contexto. Este método permite cambiar la estrategia en tiempo de ejecución.

```
public void SetPaymentStrategy(IPaymentStrategy paymentStrategy)
```

Parameters

paymentStrategy [IPaymentStrategy](#).

La estrategia de pago a usar.

Namespace Comportamiento.Tests

Classes

[PaymentTests](#)

Class PaymentTests

Namespace: [Comportamiento.Tests](#)

Assembly: Comportamiento.Tests.dll

```
public class PaymentTests
```

Inheritance

[object](#) ← PaymentTests

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Methods

GivenAnInvalidPaymentType_WhenProcessPayment_ThrowsArgumentOutOfRangeException(int, double)

```
[TestCase(4, 4000)]  
public void GivenAnInvalidPaymentType_WhenProcessPayment_ThrowsArgumentOutOfRangeException(int  
paymentType, double amount)
```

Parameters

paymentType [int](#)

amount [double](#)

GivenCashPayment_WhenProcessPayment_ThenPaymentIsProcessed(int, double)

```
[TestCase(1, 1000)]  
public void GivenCashPayment_WhenProcessPayment_ThenPaymentIsProcessed(int paymentType,  
double amount)
```

Parameters

paymentType [int](#)

amount [double](#)

GivenCreditCardPayment_WhenProcessPayment_ThenPaymentIsProcessed(int, double)

```
[TestCase(2, 2000)]
```

```
public void GivenCreditCardPayment_WhenProcessPayment_ThenPaymentIsProcessed(int  
paymentType, double amount)
```

Parameters

paymentType [int](#)

amount [double](#)

GivenDebitCardPayment_WhenProcessPayment_ThenPaymentIsProcessed(int, double)

```
[TestCase(3, 3000)]
```

```
public void GivenDebitCardPayment_WhenProcessPayment_ThenPaymentIsProcessed(int paymentType,  
double amount)
```

Parameters

paymentType [int](#)

amount [double](#)