

# Namespace Bank.Domain

## Classes

### [BankAccount](#)

Representa una cuenta bancaria con operaciones de débito y crédito.

# Class BankAccount


Namespace: [Bank.Domain](#)

Assembly: Bank.Domain.dll








Representa una cuenta bancaria con operaciones de débito y crédito.

```
public class BankAccount
```

## Inheritance

[object](#)  ← BankAccount

## Inherited Members

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

## Constructors

### BankAccount(string, double)

Inicializa una nueva instancia de [BankAccount](#).

```
public BankAccount(string customerName, double balance)
```

## Parameters

**customerName** [string](#) 

Nombre del cliente.

**balance** [double](#) 

Saldo inicial.

## Fields

### CreditAmountLessThanZeroMessage

Mensaje de error cuando el monto a acreditar es menor que cero.

```
public const string CreditAmountLessThanZeroMessage = "Credit amount is less than zero"
```

Field Value

[string](#)

## DebitAmountExceedsBalanceMessage

Mensaje de error cuando el monto a debitar excede el saldo.

```
public const string DebitAmountExceedsBalanceMessage = "Debit amount exceeds balance"
```

Field Value

[string](#)

## DebitAmountLessThanZeroMessage

Mensaje de error cuando el monto a debitar es menor que cero.

```
public const string DebitAmountLessThanZeroMessage = "Debit amount is less than zero"
```

Field Value

[string](#)

## Properties

### Balance

Obtiene el saldo de la cuenta.

```
public double Balance { get; }
```

Property Value

[double](#)

## CustomerName

Obtiene el nombre del cliente.

```
public string CustomerName { get; }
```

Property Value

[string](#)

## Methods

### Credit(double)

Acredita un monto en la cuenta.

```
public void Credit(double amount)
```

Parameters

**amount** [double](#)

Monto a acreditar.

### Debit(double)

Debita un monto de la cuenta.

```
public void Debit(double amount)
```

Parameters

**amount** [double](#)

Monto a debitar.

# Namespace Bank.Domain.Tests

## Classes

[BankAccountTests](#)


# Class BankAccountTests

Namespace: [Bank.Domain.Tests](#)








Assembly: Bank.Domain.Tests.dll

```
public class BankAccountTests
```

## Inheritance

[object](#)  ← BankAccountTests

## Inherited Members

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

## Methods

### Constructor\_SetsCustomerNameAndBalanceCorrectly()

```
[Test]  
public void Constructor_SetsCustomerNameAndBalanceCorrectly()
```

### Credit\_WhenAmountIsLessThanZero\_ShouldThrowArgumentOutOfRangeException()

```
[Test]  
public void Credit_WhenAmountIsLessThanZero_ShouldThrowArgumentOutOfRangeException()
```

### Credit\_WithValidAmount\_UpdatesBalance()

```
[Test]  
public void Credit_WithValidAmount_UpdatesBalance()
```

## Debit\_WhenAmountIsLessThanZero\_ShouldThrowArgumentOutOfRangeException()

[Test]

```
public void Debit_WhenAmountIsLessThanZero_ShouldThrowArgumentOutOfRangeException()
```

## Debit\_WhenAmountIsMoreThanBalance\_ShouldThrowArgumentOutOfRangeException()

[Test]

```
public void Debit_WhenAmountIsMoreThanBalance_ShouldThrowArgumentOutOfRangeException()
```

## Debit\_WithValidAmount\_UpdatesBalance()

[Test]

```
public void Debit_WithValidAmount_UpdatesBalance()
```