

# Namespace Bank.Domain

## Classes

### [BankAccount](#)

Represents a bank account with basic debit and credit operations.

# Class BankAccount


Namespace: [Bank.Domain](#)

Assembly: Bank.Domain.dll








Represents a bank account with basic debit and credit operations.

```
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)

Initializes a new instance of the [BankAccount](#) class.

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

## Parameters

**customerName** [string](#) 

The name of the customer.

**balance** [double](#) 

The initial balance.

## Fields

# DebitAmountExceedsBalanceMessage

Message shown when debit amount exceeds balance.

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

Field Value

[string](#)

# DebitAmountLessThanZeroMessage

Message shown when debit amount is less than zero.

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

Field Value

[string](#)

## Properties

### Balance

Gets the current balance.

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

Property Value

[double](#)

### CustomerName

Gets the name of the customer.

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

Property Value

[string](#)

## Methods

### Credit(double)

Credits the specified amount to the account.

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

Parameters

amount [double](#)

The amount to credit.

Exceptions

[ArgumentOutOfRangeException](#)

Thrown when amount is less than zero.

### Debit(double)

Debits the specified amount from the account.

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

Parameters

amount [double](#)

The amount to debit.

# Exceptions

## [ArgumentOutOfRangeException](#)

Thrown when amount is less than zero or exceeds balance.