

# Programación Orientada a Objetos 3

## Ejemplo Control Bancario (1)

### Contenido

**Control Bancario**

**Creación de la clase CuentaBancaria**

**Creación de la clase Cliente**

**Creación de la clase Banco**



# Programa de Ejemplo - 1

- Control Bancario
  - Desarrollar un programa de consola en C# que permita llevar el control de cuentas bancarias en un banco



# Creación de la clase CuentaBancaria

- Clase CuentaBancaria
  - Una variable para guardar el saldo de la cuenta
    - **private double saldo**
  - No debe haber saldos negativos
  - Constructor que reciba saldo de la cuenta y lo asigne a saldo
    - **public CuentaBancaria(double saldo)**
  - Metodos públicos:
    - **public double obtenSaldo()** . regresa saldo actual
    - **public void deposita(double cantidad)** . realizar deposito a la cuenta
    - **public bool retira(double cantidad)**. Para realizar un retiro a la cuenta, regresa true si se pudo, false si no





## Outline



### Clase

### CuentaBancaria

```
using System;
using System.Collections.Generic;
using System.Text;
namespace Programa01
{
    public class CuentaBancaria
    {
        private double saldo;
        public CuentaBancaria(double saldo)
        {
            this.saldo = saldo;
        }
        public double ObtenSaldo()
        {
            return saldo;
        }
        public void Deposita(double cantidad)
        {
            saldo += cantidad;
        }
        public bool Retira(double cantidad)
        {
            if (saldo >= cantidad)
            {
                saldo -= cantidad;
                return true;
            }
            else return false;
        }
    }
}
```



## Outline



**Clase:**

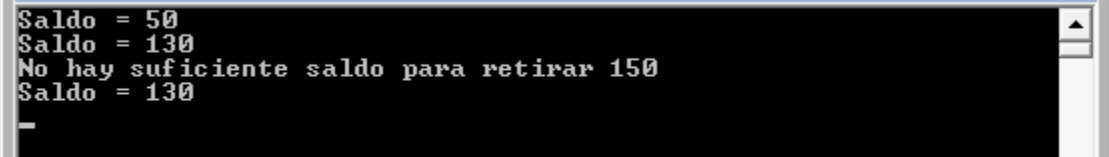
**PruebaCuenta**

```
using System;
using System.Collections.Generic;
using System.Text;
namespace Programa01
{
    class PruebaCuenta
    {
        static void Main(string[] args)
        {
            CuentaBancaria cuenta = new CuentaBancaria(50.0);
            bool resultado;
            Console.WriteLine("Saldo = {0}", cuenta.ObtenSaldo());
            cuenta.Deposita(80.0);
            Console.WriteLine("Saldo = {0}", cuenta.ObtenSaldo());
            resultado = cuenta.Retira(150.0);

            if (resultado == false)
                Console.WriteLine("No hay suficiente saldo para retirar " + 150.0);

            Console.WriteLine("Saldo = {0}", cuenta.ObtenSaldo());

            Console.ReadLine();
        }
    }
}
```



```
Saldo = 50
Saldo = 130
No hay suficiente saldo para retirar 150
Saldo = 130
_
```

# Creación de la clase Cliente

- Clase Cliente
  - Tres atributos privados
    - **private string nombre**
    - **private string apellidoPaterno**
    - **private CuentaBancaria cuenta**
  - Constructor que reciba dos parametros (app, nom) e inicialize los atributos de la clase
    - **public Cliente(string nom, string apat)**
  - Métodos públicos para acceder a los atributos de la clase
    - **public string obtenNombre()**
    - **public string obtenApellidoPat()**
  - Métodos públicos para recuperar y establecer atributo **cuenta**
    - **public obtenCuenta()**
    - **public estableceCuenta(CuentaBancaria cta)**





## Outline

**Clase:**

**Cliente**

```
using System;
using System.Collections.Generic;
using System.Text;
namespace Programa01
{
    public class Cliente
    {
        private string nombre;
        private string apellidopaterno;
        private CuentaBancaria cuenta;
        public Cliente(string nom, string apat)
        {
            this.nombre = nom;
            this.apellidopaterno = apat;
        }
        public string ObtenNombre()
        {
            return nombre;
        }
        public string ObtenApellidoPat()
        {
            return apellidopaterno;
        }
        public CuentaBancaria ObtenCuenta()
        {
            return cuenta;
        }
        public void EstableceCuenta(CuentaBancaria cta)
        {
            this.cuenta = cta;
        }
    }
}
```



## Outline



**Clase:**

**PruebaCliente**

```
using System;
using System.Collections.Generic;
using System.Text;
namespace Programa01
{
    class PruebaCliente
    {
        static void Main(string[] args)
        {
            CuentaBancaria cuenta = null;
            Cliente clientazo;

            //Crea una cuenta bancaria con saldo de 500
            Console.WriteLine("Creando al cliente Juan Perez");
            clientazo = new Cliente("Juan", "Perez");

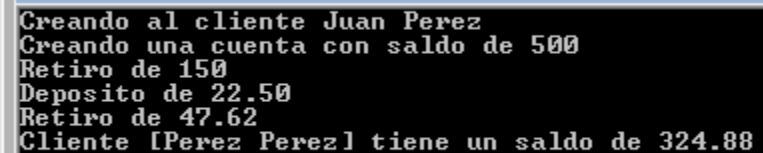
            Console.WriteLine("Creando una cuenta con saldo de 500");
            clientazo.EstableceCuenta(new CuentaBancaria(500));
            cuenta = clientazo.ObtenCuenta();

            Console.WriteLine("Retiro de 150");
            cuenta.Retira(150);

            Console.WriteLine("Deposito de 22.50");
            cuenta.Deposita(22.5);

            Console.WriteLine("Retiro de 47.62");
            cuenta.Retira(47.62);

            // Imprime saldo final de la cuenta
            Console.WriteLine("Cliente [{0} {1}] tiene un saldo de {2}",
                clientazo.ObtenNombre(), clientazo.ObtenApellidoPat(),
                cuenta.ObtenSaldo());
            Console.ReadLine();
        }
    }
}
```



```
Creando al cliente Juan Perez
Creando una cuenta con saldo de 500
Retiro de 150
Deposito de 22.50
Retiro de 47.62
Cliente [Perez Perez] tiene un saldo de 324.88
```



## Creando la clase Banco

- Clase Banco
  - Tres atributos privados
    - **private Cliente[] clientes**
    - **private int numeroDeClientes**
  - Constructor publico que inicialice el arreglo a un maximo de 10 clientes y establezca numero de clientes a cero 0.
  - Metodo publico
    - **public agregaCliente(string nom, string apat)**
      - Debe construir un objeto **Cliente** a partir de los parámetros recibidos y colocarlo en arreglo **clientes**, también debe incrementar el atributo **numeroDeClientes**
    - **public obtenCliente(int indice)**
      - Debe regresar el cliente asociado al índice como parámetro
    - **public obtenNumeroDeClientes()**
      - Regresa el número de clientes en el arreglo **clientes**

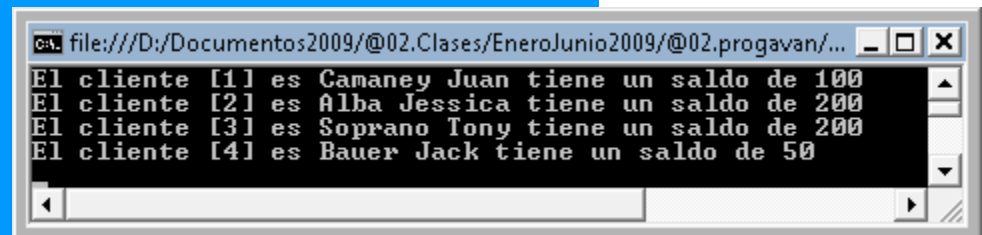


# Outline

Clase:

**PruebaBanco**

```
using System;
using System.Collections.Generic;
using System.Text;
namespace Programa01
{
    class PruebaBanco
    {
        static void Main(string[] args)
        {
            Banco mibanco = new Banco();
            CuentaBancaria cta;
            mibanco.AgregaCliente("Juan", "Camaney");
            mibanco.AgregaCliente("Jessica", "Alba");
            mibanco.AgregaCliente("Tony", "Soprano");
            mibanco.AgregaCliente("Jack", "Bauer");
            cta = new CuentaBancaria(100);
            mibanco.ObtenCliente(0).EstableceCuenta(cta);
            cta = new CuentaBancaria(200);
            mibanco.ObtenCliente(1).EstableceCuenta(cta);
            cta = new CuentaBancaria(300);
            mibanco.ObtenCliente(2).EstableceCuenta(cta);
            cta.Retira(100);
            cta = new CuentaBancaria(0);
            mibanco.ObtenCliente(3).EstableceCuenta(cta);
            cta.Deposita(50);
            for (int i = 0; i < mibanco.ObtenNumeroDeClientes(); i++)
            {
                Cliente cliente = mibanco.ObtenCliente(i);
                cta = cliente.ObtenCuenta();
                Console.WriteLine("El cliente [{0}] es {1} {2} tiene un saldo de {3}",
                    (i + 1),
                    cliente.ObtenApellidoPat(),
                    cliente.ObtenNombre(), cta.ObtenSaldo());
            }
            Console.ReadLine();
        }
    }
}
```



```
cmd: file:///D:/Documentos2009/@02.Clases/EneroJunio2009/@02.progavan/...
El cliente [1] es Camaney Juan tiene un saldo de 100
El cliente [2] es Alba Jessica tiene un saldo de 200
El cliente [3] es Soprano Tony tiene un saldo de 200
El cliente [4] es Bauer Jack tiene un saldo de 50
```