**Introducción a POO en VBnet**

Una clase está compuesta por:

* Atributos - Propiedades
* Métodos - Comportamientos

Class CCuenta

*'Atributos*

Private nombre As String

Private cuenta As String

Private saldo As Double

Private tipoDeInterés As Double

……

*'Métodos*

Public Sub New()

End Sub

Public Sub New(nom As String, cue As String, \_

sal As Double, tipo As Double)

asignarNombre(nom)

asignarCuenta(cue)

ingreso(sal)

asignarTipoDeInters(tipo)

End Sub

End Class

Los atributos, generalmente se declaran **Private**, para que no se pueda acceder a ellos desde fuera de la clase.

Esta ocultación es lo que se denomina **ENCAPSULAMIENTO**. Con esto, se protege la estructura interna del objeto y se obliga a programar pensando y creando objetos para utilizar su interfaz pública.

Los métodos la mayoría de ellos se declaran públicos, para poder acceder desde cualquier punto del proyecto.

Los métodos son los que dan funcionalidad al objeto. Son los procedimientos (**Sub** o **Function**).

**Método Sobrecargado**: Si existe más un método con el mismo nombre, pero que difieren en el número de parámetros o en el tipo de alguno de ellos, se dice que el método está **SOBRECARGADO**.

Para acceder a los métodos de una clase, crearemos previamente un objeto.

Crear un Objeto: La sintaxis es:

Dim objeto As Calumno = New Calumno(xxx, xxx, xxx)

objeto.metodo(xxx)

El **CONSTRUCTOR** es un método esencial y especial de la clase que es llamado automáticamente siempre que se crea un objeto de esa clase. Su función es iniciar el objeto.

Se distingue fácilmente porque tiene el nombre **New** y no puede retornar un valor (procedimiento de tipo **Sub**)

Public Sub New ()

End Sub

Un constructor por omisión de una clase, es un constructor sin parámetros que no hace nada. El Objeto será iniciado con los valores predeterminados por el sistema:

* Atributos numéricos a **ceros**
* Atributos alfanuméricos y las referencias a objetos a **Nothing**.

Los constructores, salvo en casos excepcionales, deben declararse siempre públicos para que puedan ser invocados desde cualquier parte.

Podemos añadir también un constructor a la clase con el fin de poder iniciar los atributos de cada nuevo objeto con unos valores determinados pasados como argumentos, en el instante en el que solicita crearlos. (Este sería un ejemplo de un método que está **sobrecargado**)

Hemos dicho que los atributos se declaran privados y los métodos, la mayoría de ellos se declaran públicos.

Cuando creamos objetos, cada objeto que creemos de esa clase mantiene su propia copia de los atributos para almacenar sus datos particulares; pero, de los métodos sólo hay una copia para todos los objetos, ya que son comunes a todos ellos.

**Atributos y Métodos SHARED**

Cuando un atributo de la clase, almacena información común a todos los objetos de esa clase, no tiene sentido que se guarde la misma información en todos los objetos.

Para especificar un atributo común a todos los objetos de su clase, cuando se declare el atributo, hay que anteponer la palabra reservada **Shared** al nombre del mismo (**Static** en otros lenguajes).

Private Shared TipoDeInteres As Double

Análogamente, un método declarado **Shared** es un método de la clase. Por lo tanto, no se ejecuta para un objeto particular, sino que se utiliza para actuar sobre un atributo **Shared** declarado privado, o bien para realizar alguna operación genérica al margen de los objetos de clase.

Public Shared Sub setTipoDeInteres(xxx)

xxx

End Sub