

Diseño Basado en Componentes

Programación Orientada a Objetos (OO) en VB.NET

Information Engineering 

Ingeniería Informática
Universidad Carlos III de Madrid

Diseño Basado en Componentes.
Curso 2008 / 09

Beneficios de OOP

- Conceptos orientados a objetos.
 - **Abstracción:** Identificación de objetos a través de sus aspectos conceptuales.
 - **Encapsulación:** Separación de la interfaz del objeto de su implementación.
 - **Herencia:** Partiendo de una superclase (clase padre) se crean nuevas subclases (clases hijas)
 - **Polimorfismo:** Ante el mismo mensaje se realizarán distintas acciones según el objeto al que sea aplicado.
 - **Sobrecarga:** Varios métodos con idéntico nombre dentro de la misma clase, distinguiéndose por la lista de parámetros.

OO: Clase

- Especificación de un conjunto de objetos con **estructura** y **comportamiento** comunes.
- Comportamiento (operaciones): operaciones o servicios que pueden realizar los objetos de la clase, en respuesta a mensajes recibidos de otros objetos.
- Estado (atributos): valores almacenados en el objeto de la clase, resultado de las operaciones pasadas.
- Los atributos y las operaciones relacionadas se empaquetan juntas y se definen como un tipo.
- Molde para crear objetos.

Clase "Vehículo"

Vehículo	
Estado	matrícula marca modelo año ...
Comportamiento	arrancar() acelerar() cambiarMarcha() frenar() apagar() ...

Information Engineering

OOP: Objeto

- Cuando se crea una "instancia" variable de una clase se le llama **OBJETO**.

Clase Figura

Molde para fabricar Figuras rectangulares

Objeto Figura1

Objeto Figura2

Objeto Figura3

Instancias de la clase Figura (iguales en Forma, pero con propiedad Color distinta)

Information Engineering

Objeto de tipo "Vehículo"

miCoche:Vehículo

matricula = "2322 CLS"
marca = "Toyota"
modelo = "Corolla"
año = 2005
...

Information Engineering

Clases en VB.NET (I)

- Normalmente un fichero de código por clase (aunque se pueden incluir varias clases en un mismo fichero).
- Una clase sólo puede heredar de una única clase (es decir, no se permite herencia múltiple).

```
Public Class NombreClase  
    'código de la clase  
End Class
```

Information Engineering

Clases en VB.NET (II)

- Control de acceso a los miembros de una clase:
 - **Public:** El miembro clase será accesible por todo el código de la clase, sus subclases y otras clases.
 - **Private:** El miembro de la clase sólo será accesible por el código de la clase.
 - **Protected:** La clase será accesible desde el código de la clase y desde cualquier subclase.
 - **Friend:** La clase será accesible por todo el código de su proyecto o ensamblado.
 - **Protected Friend:** Combinación de los dos modificadores de acceso.

Information Engineering

Información de la Clase (I)

- **Campo:** variable, generalmente de ámbito público, accesible desde el exterior de la clase.

```
Public Class Vehiculo
    'Campo de la clase para guardar el
    'identificador del vehículo
    Public identificador As Integer
    ...
End Class
```

Information Engineering

Información de la Clase (II)

- **Propiedad:** definida por dos elementos: una variable de propiedad y un procedimiento de propiedad.
 - **Variable de propiedad:** variable con ámbito privado a nivel de la clase que se encarga de guardar el valor de la propiedad (atributo).
 - Sólo lectura: **ReadOnly Property**
 - Sólo escritura: **WriteOnly Property**
 - **Procedimiento de propiedad:** encargado de realizar las operaciones de acceso (GET) y asignación (SET) de valores a la variable de propiedad.

Information Engineering

Información de la Clase (III)

```
Public Class Vehiculo
    Private _matr As String
    Public Property Matrícula() As String
        Get
            Return _matr
        End Get
        Set (ByVal Value As String)
            _matr = Value
        End Set
    End Property
    ...
End Class
```

Diagram annotations:

- Variable de Propiedad:** points to `Private _matr As String`
- Propiedad:** points to `Public Property Matrícula() As String`
- Procedimiento de Propiedad:** points to the `Get` and `Set` blocks.
- Lectura:** points to the `Return _matr` line.
- Escritura:** points to the `_matr = Value` line.

Information Engineering

Métodos de la Clase (I)

- Miembros de una clase que definen el comportamiento de los objetos.
- Procedimiento tipo **Sub**.
- Procedimiento tipo **Function**.
- Parámetros por valor: **ByVal**.
- Parámetros por referencia: **ByRef**.

Information Engineering

Métodos de la Clase (II)

- Procedimiento "Sub"

```
Sub Prueba (ByVal param1 as Integer, ByVal param2 as Integer)
...
End Sub
```

- Procedimiento "Function"

```
Function Prueba (ByVal param1 as Integer) as Integer
    Dim varAux as Integer
    ...
    'Prueba = varAux
    Return varAux
End Function
```

Information Engineering

Métodos de la clase (III)

- **Sobrecarga** de métodos (mismo nombre, distinta lista de parámetros): Para indicar que el método está sobrecargado, utilizar la palabra clave **Overloads** en la declaración del método.
- **Sobreescritura** de métodos (**polimorfismo**):
 - Para indicar que el método puede ser sobrescrito, utilizar la palabra clave **Overridable** en la declaración del método.
 - Para indicar que el método sobrescribe a otro ya existente, utilizar la palabra clave **Overrides** en la declaración del método.
 - Para indicar que el método no puede ser sobrescrito, utilizar la palabra clave **NotOverridable** en la declaración del método.

Information Engineering

Métodos Constructores

- Método público "Sub" con el nombre **New()**.
- Se invoca cuando se crea el objeto (**instanciación**).
- Útil para tareas de configuración iniciales sobre el objeto.
- Pueden estar sobrecargados.

```
Public Class Vehiculo
    ...
    Public Sub New (ByVal marca as String,
                  ByVal modelo As String)
        _marca = marca
        _modelo = modelo
    End Function
End Class
```

Information Engineering

Me y MyBase (I)

- **Me**: Se utiliza para tener acceso a los miembros de la propia clase (ej., Me.calcularSalario(), Me._marca).
- **MyBase**: Se utiliza para tener acceso a los miembros de una clase base desde su correspondiente subclase.

Information Engineering

Me y MyBase (II)

```
Public Class Administrativo : Inherits Empleado
    Public Overloads Sub calcularIncentivos(ByVal horas As Integer)
        ' Se llama a la clase base con MyBase para hacer
        ' en primer lugar los mismos cálculos de incentivos
        ' de la clase Empleado
        MyBase.calcularIncentivos()
        ' Después se hacen los cálculos propios de esta clase
        Me._incentivos += horas * 15
    End Sub
End Class
```

Information Engineering

Clases Abstractas

- No permite la instanciación directa de objetos a partir de ella.
- Se debe utilizar la palabra clave **MustInherit** en el momento de su declaración.
- Los miembros abstractos de la clase irán declarados utilizando la palabra clave **MustOverride**.

```
Public MustInherit Class Vehiculo
    Public Sub calcularCredito(ByVal balance As Integer)
        Me._credito = balance * 0,4
    End Sub
    Public MustOverride Sub calcularImpuestos ()
End Class
```

Information Engineering

Clases Selladas o No Heredables

- Toda clase declarada en el código es en principio heredable.
- Para hacer que una clase no pueda tener subclases, se debe utilizar la palabra clave **NotInheritable** en el momento de su declaración.

```
Public NotInheritable Class Fichero
    'Código de la clase
End Class
```

Information Engineering

Objetos en VB.NET (I)

- Objetos en VB.NET:
 - Declaración e instanciación en distintas líneas de código.
 - Declaración e instanciación en la misma línea de código.
 - Declaración e instanciación simultánea.

Information Engineering

Objetos en VB.NET (II)

- Declaración e instanciación en diferentes líneas de código.

```
Dim miCoche As Vehículo  
miCoche = New Vehículo("Toyota", "Corolla")
```

Objeto

Clase

Palabra clave (instanciación)

Constructor de la clase Coche con sus parámetros

Declaración
Instanciación

Information Engineering

Objetos en VB.NET (III)

- Declaración e instanciación en la misma línea de código.

```
Dim miCoche As Vehículo = New Vehículo("Toyota", "Corolla")
```

Objeto

Clase

Palabra clave (instanciación)

Constructor de la clase Coche con sus parámetros

Declaración e Instanciación

Information Engineering

Objetos en VB.NET (IV)

- Declaración e instanciación simultánea.

```
Dim miCoche As New Vehículo("Toyota", "Corolla")
```

Palabra clave (instanciación)

Constructor de la clase Coche con sus parámetros

Objeto

Clase

Declaración e Instanciación

Information Engineering

Objetos en VB.NET (V)

- Acceso a un campo de la clase

```
Sub Main ()  
    Dim miCoche As Vehículo  
    miCoche = new Vehículo ()  
    miCoche.identificador = 975  
End Sub
```

Information Engineering

Objetos en VB.NET (VI)

- Acceso a una propiedad

```
Sub Main ()
    Dim miCoche As Vehiculo
    miCoche = new Vehiculo ("Toyota", "Corolla")
    miCoche.Matricula = "8937 CZL"
    Console.WriteLine ("Matricula coche: " & _
        miCoche.Matricula)
    Console.ReadLine()
End Sub
```

Information Engineering

Interfaces

- Naturaleza declarativa: lista de propiedades y métodos que serán codificados en una o varias clases.
- Una clase puede implementar múltiples interfaces, pero solo puede heredar de una única clase.
- Las interfaces pueden proporcionar polimorfismo.
- Se utilizará la palabra clave **Interface** en su declaración.
- La clase que la implemente utilizará la palabra clave **Implements** en su declaración.

```
Public Interface Cadena
    ReadOnly Property Longitud() As Integer
    Function ObtenerValor() As String
End Interface
Public Class Empleado : Implements Cadena
    'Código de la clase
End Class
```

Information Engineering

Namespaces en VB.NET

- Un "namespace" o espacio de nombres, es un contenedor lógico que permiten organizar de manera óptima las clases dentro de un proyecto o ensamblado.

```
Namespace Personal
    Public Class Empleado
        'Código de la clase
    End Class

Imports Gestion.Personal
Public Class Principal
    Public Shared Sub Main
        Dim Emp As Empleado
        ...
    End Sub
End Class
```

Diseño Basado en Componentes

Programación Orientada a Objetos (OO) en VB.NET

Information Engineering 

Ingeniería Informática
Universidad Carlos III de Madrid

Diseño Basado en Componentes.
Curso 2008 / 09