Overriding Properties and Methods

Visual Studio 2008

A derived class inherits the properties and methods defined in its base class. This is useful because you can reuse these items when appropriate for the derived class. If the property or method in the base class is marked with the **Overridable** keyword, you can define a new implementation for the member in the derived class. Use the **Overrides** keyword to shadow the member by redefining it in the derived class. This is useful when you cannot use the member "as is."

In practice, overridden members are often used to implement polymorphism. For more information, see Polymorphism.

The following rules apply to overriding methods.

- You can only override members that are marked with the **Overridable** keyword in their base class.
- By default, properties and methods are **NotOverridable**.
- Overridden members must have the same arguments as the inherited members from the base class.
- The new implementation of a member can call the original implementation in the parent class by specifying **MyBase** before the method name.

> **Note:** Overload, override, and shadow are similar concepts that can be easy to confuse. For more information, see Introduction to Objects in Visual Basic.

**Example**

Suppose you want to define classes to handle payroll. You could define a generic **Payroll** class that contains a **RunPayroll** method that calculates payroll for a typical week. You could then use **Payroll** as a base class for a more specialized **BonusPayroll** class, which could be used when distributing employee bonuses.

The **BonusPayroll** class can inherit, and override, the **PayEmployee** method defined in the base **Payroll** class.

The following example defines a base class, **Payroll**, and a derived class, **BonusPayroll**, which overrides an inherited method, **PayEmployee**. A procedure, **RunPayroll**, creates and then passes a **Payroll** object and a **BonusPayroll** object to a function, **Pay**, that executes the **PayEmployee** method of both objects.

```vb
Const BonusRate As Decimal = 1.45D
Const PayRate As Decimal = 14.75D

Class Payroll
    Overridable Function PayEmployee( _
        ByVal HoursWorked As Decimal, _
        ByVal PayRate As Decimal) _
        As Decimal
        PayEmployee = HoursWorked * PayRate
    End Function
End Class

Class BonusPayroll
    Inherits Payroll
    Overrides Function PayEmployee( _
        ByVal HoursWorked As Decimal, _
        ByVal PayRate As Decimal) _
        As Decimal
        ' The following code calls the original method in the base
        ' class, and then modifies the returned value.
        PayEmployee = MyBase.PayEmployee(HoursWorked, PayRate) * BonusRate
    End Function
End Class

Sub RunPayroll()
    Dim PayrollItem As Payroll = New Payroll
    Dim BonusPayrollItem As New BonusPayroll
    Dim HoursWorked As Decimal = 40

    MsgBox("Normal pay is: " & _
        PayrollItem.PayEmployee(HoursWorked, PayRate))
    MsgBox("Pay with bonus is: " & _
        BonusPayrollItem.PayEmployee(HoursWorked, PayRate))
End Sub
```
See Also

Concepts
Overloaded Properties and Methods
Override Modifiers
Shadowing in Visual Basic
Other Resources
Polymorphism

Community Additions

© 2014 Microsoft