

Polimorfismo no Visual Basic por Marcio Belo

Uma das maiores deficiências apontadas por desenvolvedores em geral com relação à versão atual do Visual Basic (6.0) é a ausência de herança. Porém, quero demonstrar neste artigo uma capacidade muito pouco explorada desta linguagem e que permite construir uma hierarquia de objetos polimórficos.

O que é Polimorfismo ?

Sem querer polemizar este tópico, uma vez que existem autores que apontam mais de um tipo de polimorfismo, uma característica consensual e apontada como determinante do conceito é a capacidade de uma variável de objeto poder referenciar-se a diferentes classes, sendo estas classes derivadas de uma classe base única.

O VB 6 suporta Polimorfismo ?

Segundo puristas de OO, a resposta é não, uma vez que a forma tradicional de prover polimorfismo é através de herança, chamado de polimorfismo de herança. Entretanto, o VB6 possui a capacidade de implementar o chamado polimorfismo de interface, técnica amplamente utilizada na implementação de componentes da tecnologia ActiveX da Microsoft.

Esta técnica consiste em criar uma classe abstrata – que define propriedades e métodos comuns – e criar classes derivadas que implementem essa interface definida na classe base. Classe abstrata pode ser entendida como aquela que não é criada para ser instanciada (gerar um objeto). Ela tem o objetivo de apenas definir um molde para criação de classes derivadas.

Exemplo Prático

Imagine um cenário onde queremos modelar uma aplicação de Recursos Humanos. Este sistema deve ter a capacidade de tratar dois tipos de funcionários: Analistas e Programadores, cada um desses tipos tem uma forma diferente de cálculo para o salário, embora ambos ganhem um salário base mais uma gratificação.

Sendo assim, teríamos um classe base chamada de Funcionário. O código a seguir definiria a classe pseudo-abstrata Funcionario:

```
Option Explicit

Public Function CalcularSalario() As Currency
End Function

Public Property Get Nivel() As Integer
End Property

Public Property Let Nivel(ByVal vNewValue As Integer)
End Property
```

Classifiquei-a como classe pseudo-abstrata pois o VB6 não suporta classes/métodos abstratos. No VB6, qualquer classe pode ser instanciada.

Agora implementaremos duas classes derivadas da classe Funcionário, que especializariam a classe base para implementar características específicas para Analista e para Programador:

Código da Classe Programador:

```
Option Explicit
Implements Funcionario
Const SalBase = 450
```

```

Const GanhoPorNivel = 5
Private mintNivel As Integer

Private Function Funcionario_CalcularSalario() As Currency
    Funcionario_CalcularSalario = SalBase + (mintNivel * GanhoPorNivel)
End Function

Private Property Let Funcionario_Nivel(ByVal RHS As Integer)
    mintNivel = RHS
End Property

Private Property Get Funcionario_Nivel() As Integer
    Funcionario_Nivel = mintNivel
End Property

```

Código da Classe Analista:

```

Option Explicit
Implements Funcionario

Const SalBase = 1000
Const GanhoPorNivel = 10
Private mintNivel As Integer

Private Function Funcionario_CalcularSalario() As Currency
    Funcionario_CalcularSalario = SalBase + (mintNivel * GanhoPorNivel)
End Function

Private Property Let Funcionario_Nivel(ByVal RHS As Integer)
    mintNivel = RHS
End Property

Private Property Get Funcionario_Nivel() As Integer
    Funcionario_Nivel = mintNivel
End Property

```

Uma vez definidas as nossas classes, vamos implementar a interface que criaria a instância da classe funcionário e utilizaria a característica polifórmica para calcular o salário de um funcionário.



Verifique o código que calcula o salário do Funcionário:

```

Option Explicit

Private Sub cmdCalcSal_Click()
    Dim objFunc As Funcionario

    If optNivel(0).Value Then
        Set objFunc = New Analista
    Else
        Set objFunc = New Programador
    End If
    objFunc.Nivel = CInt(txtNivel.Text)
    MsgBox "Salário: " & CStr(objFunc.CalcularSalario())
    Set objFunc = Nothing
End Sub

```

Repare que é declarada uma variável para a classe Funcionário. Dependendo do botão de opção que estiver selecionado - Analista ou Programador – a respectiva classe derivada é instanciada utilizando esta variável.

Polimorfismo no Visual Basic.NET

Com a introdução pela Microsoft da versão VB.NET (anteriormente conhecida como versão 7.0), a linguagem Visual Basic dará um enorme salto com relação aos recursos de orientação à objetos, embutindo conceitos como herança e polimorfismo de forma limpa.

Marcio Belo (mbelo@pcrj.rj.gov.br) é analista de sistemas da IplanRio, Microsoft Certified Professional e professor universitário da UniverCidade.