Re: OOP Question

Source:
http://www.tech−archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2004−06/3977.html

---

From: Jay B. Harlow [MVP − Outlook] (*Jay_Harlow_MVP_at_msn.com*)
Date: 06/17/04

Date: Thu, 17 Jun 2004 09:59:04 −0500

Chris,
As OHM suggested, the OOP way to code classes for small tables of data is System.Data.DataTable, I would consider keeping both of your DataTables in a System.Data.DataSet, specifically a Typed DataSet.

Martin Fowler's book "Patterns of Enterprise Application Architecture" from Addison Wesley http://www.martinfowler.com/books.html#eaa explains when you may want to use a traditional Domain Model & Data Mapper pattern:
http://www.martinfowler.com/eaaCatalog/domainModel.html
http://www.martinfowler.com/eaaCatalog/dataMapper.html

verses a Table Module & Data Gateway patterns:
http://www.martinfowler.com/eaaCatalog/tableModule.html
http://www.martinfowler.com/eaaCatalog/tableDataGateway.html

Martin also offers a couple of other useful patterns that can be used instead of or in conjunction with the above patterns.

FWIW: The System.Data.DataTable is an implementation of a Record Set pattern:
http://www.martinfowler.com/eaaCatalog/recordSet.html

Rockford Lhotka's book "Expert One−on−One Visual Basic .NET Business Objects" from A! Press provides a pre−implemented variation of Fowler's Domain Model & Data Mapper patterns.
http://www.lhotka.net/

Generally if there is no real logic behind my domain objects, I would use the DataSet OOM coupled with a Table Module & Data Gateway patterns. As the classes themselves are not really living up to their potential! :−) The Table Module & Data Gateway patterns may be implemented in a single class or two classes. Again I would consider using a Typed DataSet.

However if there is significant logic behind my domain objects, I would then favor the Domain Model & Data Mapper patterns.

Depending on the needs of the project I would consider Fowler's other
patterns...

Hope this helps
Jay

"Chris" <cevans@bugmonitor.com> wrote in message
news:B28Ac.9681$IF3.4920@bignews5.bellsouth.net...
> *Can someone tell me the OOP way to code classes for small tables of data?*
>
> *For example, i'm loading several tables that will be stored in the*
> *current.cache of an asp.net app. One table is credit cards; another is*
> *countries.*
>
> *Do i need two classes for each of these? A parent and a child class*
(similar
> *to tables and table in ado.net) or is there a better way to do it?*
>
> *See code below...*
>
> *Thanks,*
>
> *Chris*
> *BugMonitor.com*
>
> *Public Class Category*
> *Public Attrib1 As Integer*
> *Public Attrib2 As String*
> *End Class*
>
> *Public Class Categories*
> *Public Shared Category As Category()*
> *Public Shared Count As Int16*
>
> *Public Shared Function GetAll() As Categories*
> *Dim bCategory As Category*
> *While dr.Read*
> *bCategory = New Category*
> *With bCategory*
> *.Attrib1 = val1*
> *.Attrib2 = val2*
> *End With*
> *al.Add(bCategory)*
> *Count += 1S*
> *End While*
>
> *Dim bCategories As New Categories*
> *bCategories.Category = DirectCast(al.ToArray(GetType(Category)),*
> *Category())*
>
> *Return bCategories*

> *End Function*
>
> *End Class*
>
>