

Re: OOP object collections

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2008-02/msg01040.html>

- *From:* "sloan" <sloan@xxxxxxxx>
 - *Date:* Fri, 8 Feb 2008 13:05:04 -0500
-

You got it!

Again

<http://sholliday.spaces.live.com/Blog/cns!A68482B9628A842A!158.entry>

I have the .Sort (for Zebras) done in the ZebraController example.

```
//Lets let the Biz Layer do the Sorting  
returnCollection.Sort(new  
Comparers.ZebraComparer(GranadaCoder.Applications.TrinugWCFDemoVersion1.Interfaces.Comparers.ZebraType.
```

"RSH" <way_beyond_oops@xxxxxxxx> wrote in message
news:uJ2ybSnaIHA.1164@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

```
Oops...I hadnt finished the CompareTo method...here is what I came up  
with...  
public int CompareTo(IEmployee other)  
  
{  
  
switch (m_Parent.Employees.SortBy)  
  
{  
  
case EmployeeSortType.Id:  
  
return this.m_Id.CompareTo(other.Id);  
  
case EmployeeSortType.FirstName:  
  
return this.m_FirstName.CompareTo(other.FirstName);
```

Re: OOP object collections

```
case EmployeeSortType.LastName:  
  
return this.m_LastName.CompareTo(other.LastName);  
  
default:  
  
return 0;  
  
}  
  
}
```

Everything seems to work as desired and I have all or the different sort methods working.

"RSH" <way_beyond_oops@xxxxxxxx> wrote in message
news:eHDMxInaIHA.5984@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Sloan,

Im struggling a bit with trying to add a custom sort method. I want to be able to sort by LastName, FirstName or ID.
...and Im having a little trouble figuring out where the different components go.

here is kind of where I am heading:

```
Employee Classes:  
public interface IEmployee : IComparable<IEmployee>  
  
{  
  
string FirstName { get; set; }  
  
int Id { get; set; }  
  
string LastName { get; set; }  
  
}  
  
public sealed class Employee : Workshop.IEmployee  
  
{  
  
private int m_Id;  
  
private string m_LastName;  
  
private string m_FirstName;  
  
private ICompany m_Parent = null;
```

Re: OOP object collections

```
public Employee(ICompany Parent, int Id, string LastName, string
FirstName)

{

m_Id = Id;

m_LastName = LastName;

m_FirstName = FirstName;

m_Parent = Parent;

}

public int Id { get { return m_Id; } set { m_Id = value; } }

public string LastName { get { return m_LastName; } set { m_LastName =
value; } }

public string FirstName { get { return m_FirstName; } set { m_FirstName =
value; } }

public int CompareTo(IEmployee other)

{

switch (m_SortBy)

{

case EmployeeSortType.Id:

return this.m_Id.CompareTo(other.Id);

case EmployeeSortType.FirstName:

return this.m_FirstName.CompareTo(other.FirstName);

case EmployeeSortType.LastName:

return this.m_LastName.CompareTo(other.LastName);

default:

return 0;

}

}
```

Re: OOP object collections

```
}  
  
public enum EmployeeSortType  
{  
    Id, FirstName, LastName  
}
```

Company Class

```
public interface ICompany  
{  
    int Id { get; }  
    string Name { get; set; }  
    IEmployeeCollection Employees { get; }  
    void PrintEmployees();  
    void LoadEmployees();  
}  
  
public interface IEmployeeCollection : IList<IEmployee>  
{  
    EmployeeSortType SortBy { get; set; }  
    void Sort();  
}  
  
public sealed class Company : ICompany  
{  
    private int m_Id;
```

Re: OOP object collections

```
private string m_Name;

private IEmployeeCollection m_EmployeeCollection = null;

public Company(int Id, string Name)

{

    m_Id = Id;

    m_Name = Name;

}

public int Id { get { return m_Id; } }

public string Name { get { return m_Name; } set { m_Name = value; } }

public IEmployeeCollection Employees

{

    get

    {

        return m_EmployeeCollection;

    }

}

public void LoadEmployees()

{

    m_EmployeeCollection = new EmployeeCollection();

    CompanyController.GetEmployees(this);

}

public void PrintEmployees()

{

    foreach (Employee employee in m_EmployeeCollection)

    {

        Console.WriteLine(string.Format("{0}\t{1}\t{2}", employee.Id,
```

Re: OOP object collections

```
employee.LastName, employee.FirstName));  
  
}  
  
}  
  
}  
  
public class EmployeeCollection : List<IEmployee>, IEmployeeCollection  
{  
  
    private EmployeeSortType m_SortBy = EmployeeSortType.LastName;  
  
    public EmployeeSortType SortBy { get { return m_SortBy; } set { m_SortBy  
= value; } }  
  
}
```

Company Controller

```
public static class CompanyController  
{  
  
    public static ICompany GetCompany(int companyId)  
  
    {  
  
        return new Company(1001, "Bills House of Bugs");  
  
    }  
  
    public static void GetEmployees(ICompany company)  
  
    {  
  
        IEmployee employee = null;  
  
        employee = new Employee(company, 1000, "House", "William");  
  
        company.Employees.Add(employee);  
  
        employee = new Employee(company, 1001, "Appleyard", "Katie");  
  
        company.Employees.Add(employee);  
  
        employee = new Employee(company, 1002, "Costen", "Rosa");  
  
    }  
  
}
```

Re: OOP object collections

```
company.Employees.Add(employee);  
employee = new Employee(company, 1003, "Herhuth", "Ron");  
company.Employees.Add(employee);  
}  
}
```