C# order guideline

(based on StyleCop4.7 rule SA1201)

Elements at the file root level or within a namespace must be positioned in the following order:

- Extern Alias Directives
- Using Directives
- Namespaces
- Delegates
- Enums
- Interfaces
- Structs
- Classes

Within a class, struct, or interface, elements must be positioned in the following order:

- Fields
- Constructors
- Finalizers (Destructors)
- Delegates
- Events
- Enums
- Interfaces
- Properties
- Indexers
- Methods
- Structs
- Classes

Complying with a standard ordering scheme based on element type can increase the readability and maintainability of the file and encourage code reuse.

When implementing an interface, it is sometimes desirable to group all members of the interface next to one another. This will sometimes require violating this rule, if the interface contains elements of different types. This problem can be solved through the use of partial classes.

1. Add the partial attribute to the class, if the class is not already partial.

2. Add a second partial class with the same name. It is possible to place this in the same file, just below the original class, or within a second file.

3. Move the interface inheritance and all members of the interface implementation to the second part of the class.

For example:

```
/// <summary>
/// Represents a customer of the system.
/// </summary>
public partial class Customer
{
    // Contains the main functionality of the class.
}

/// <content>
/// Implements the ICollection class.
/// </content>
public partial class Customer : ICollection
{
    public int Count
    {
        get { return this.count; }
    }

    public bool IsSynchronized
    {
        get { return false; }
    }

    public object SyncRoot
    {
        get { return null; }
    }

    public void CopyTo(Array array, int index)
    {
        throw new NotImplementedException();
    }
}
```