Java	C#

interface IControl
{
    void Paint();
}
interface ITextBox **extends** IControl
{
    void SetText(String text);
}
interface IListBox **extends** IControl
{
    void SetItems(String[] items);
}
interface IComboBox **extends** ITextBox, IListBox {}
	interface IControl
	{
	    void Paint();
	}
	interface ITextBox: IControl
	{
	    void SetText(string text);
	}
	interface IListBox: IControl
	{
	    void SetItems(string[] items);
	}
	interface IComboBox: ITextBox, IListBox {}

class Control {...}
interface IControl
{
    void Paint();
}
interface IDataBound
{
    void Bind(Binder b);
}
public class EditBox **extends** Control
                  implements IControl, IDataBound
{
    public void Paint() {...}
    public void Bind(Binder b) {...}
}
	class Control {...}
	interface IControl
	{
	    void Paint();
	}
	interface IDataBound
	{
	    void Bind(Binder b);
	}
	public class EditBox: Control, IControl, IDataBound
	{
	    public void Paint() {...}
	    public void Bind(Binder b) {...}
	}

interface modifiers(outer interface)
    public
    default(no modifier)
	interface modifiers
	    new
	    public
	    protected
	    internal
	    private

access interface members
interface ICloneable
{
    Object Clone();
}
interface IComparable
{
    int CompareTo(Object other);
}
class ListEntry **implements** ICloneable, IComparable
{
    public Object Clone() {...}
    public int CompareTo(Object other) {...}
}
	access interface members
	interface ICloneable
	{
	    object Clone();
	}
	interface IComparable
	{
	    int CompareTo(object other);
	}
	class ListEntry: ICloneable, IComparable
	{
	    object ICloneable.Clone() {...}
	    int IComparable.CompareTo(object other) {...}
	}

public interface ITeller
{
    void Next ();
}
 	
public interface IIterator
{
    void Next ();
}
 	
public class Clark implements ITeller, IIterator
{
    void ITeller.Next () {//error
    }
    void IIterator.Next () {//error
    }
}
(lack of flexibility and easy to be conflicted)
code one: void Next() {}//ok
	public interface ITeller
	{
	    void Next ();
	}
	
	public interface IIterator
	{
	    void Next ();
	}
	
	public class Clark : ITeller, IIterator
	{
	    void ITeller.Next () {
	    }
	    void IIterator.Next () {
	    }
	}
	Clark clark = new Clark ();
	((ITeller)clark).Next();