Inheritance sample

The code

You can find the latest version of SharpDocx’ source code [on GitHub](https://github.com/egonl/SharpDocx ).

SharpDocx is based on Microsoft’s [Open XML SDK](https://github.com/dotnet/Open-XML-SDK). The Package property in DocumentBase gives you full access to the Word document. It contains headers, footers, images and the document itself, among other things.

SharpDocx replaces each code block with an empty placeholder (a Text element). Calling Write sets the Text property of this element. If you want, you can use the placeholder to insert other things as well, such as hyperlinks. See the CreateHyperlink method for an example.

If you’re interested in the structure of a Word document or C# code representing parts of it, you should definitely have a look at the [Open XML SDK 2.5 Productivity Tool for Microsoft Office](https://github.com/dotnet/Open-XML-SDK/releases/tag/v2.5).

Create and access custom properties and methods

# Introduction

When you call DocumentFactory.Create, SharpDocx generates a new class based on code in the view. If you want, you can specify your own base class for the generated class. This class should derive from DocumentBase. In this sample the base class is MyDocument.

var doc = DocumentFactory.Create<MyDocument>(viewPath);

doc.MyProperty = "The Code";

doc.Generate(documentPath);

The class hierarchy looks like this:  
