End 2.0

不允许用分部类扩展编译好的类或其他程序集中的类。只能利用分部类在同一个程序集中将一个类的实现拆分成多个文件。

Begin 3.0

6.12.2 分部方法

C# 3.0 引入**分部方法**的概念，对 C# 2.0 的分部类进行了扩展。分部方法只能存在于分部类中，而且和分部类相似，主要作用是为代码生成提供方便。

假定代码生成工具能根据数据库中的 Person 表为 Person 类生成对应的 Person.Designer.cs 文件。该工具检查表并为表中每一列创建属性。问题在于，工具经常都不能生成必要的验证逻辑，因为这些逻辑依赖于未在表定义中嵌入的业务规则。所以，Person 类的开发者需要自己添加验证逻辑。Person.Designer.cs 是不好直接修改的，因为假如文件被重新生成（例如，为了适应数据库中新增的一个列），所做的更改就会丢失。相反，Person 类的代码结构应独立出来，使生成的代码在一个文件中，自定义代码（含业务规则）在另一个文件中，后者不受任何重新生成动作的影响。如上一节所示，分部类很适合将一个类打散成多个文件。但这样可能还不够。经常还需要**分部方法**。

分部方法允许声明方法而不需要实现。但如果包含了可选的实现，该实现就可放到某个姊妹分部类定义中，该定义可能在单独的文件中。代码清单 6.49 展示了如何为 Person 类声明和实现分部方法。

代码清单 6.49 为 Person 类声明和实现分部方法

```
// File: Person.Designer.cs
public partial class Person
{
    #region Extensibility Method Definitions
    partial void OnLastNameChanging(string value);
    partial void OnFirstNameChanging(string value);
    #endregion

    // ...
    public System.Guid PersonId
    {
        // ...
    }
    private System.Guid _PersonId;

    // ...
    public string LastName
    {
        get
        {
            return _LastName;
        }
        set
        {
            if ((_LastName != value))
            {
```