

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

C#锐利体验2.0:

局部类型、空属类型、静态类

李建忠

www.lijianzhong.com

上海祝成科技 高级讲师

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

Agenda

- 局部类型
- 空属类型
- 静态类
- 讲座总结
- Q&A

局部类型的引入

- 没有局部类型的时候（C# 1.0）：

```
class App {  
    public void Foo() {.....}  
    .....  
    public void Bar() {.....}  
    .....  
}
```

- 有了局部类型之后（C# 2.0）：

```
partial class App {  
    public void Foo()  
{.....}  
    .....  
}
```

```
partial class App {  
    public void Bar()  
{.....}  
    .....  
}
```

局部类型简介

- 局部类型允许我们将一个类型（类、结构或者接口）分成几个部分，分别实现在几个不同的.cs文件中。
- 局部类型适用于以下情况：
 - 类型特别大，不宜放在一个文件中实现；
 - 一个类型中一部分代码为自动化工具生成的代码，不宜与我们自己编写的代码混合在一起。
- 局部类型是一个纯语言层的编译处理，不影响任何执行机制——事实上C#编译器在编译的时候仍会将各个部分的局部类型合并成一个完整的类。

局部类型的一些限制

- 局部类型只适用于类、结构、或接口，不支持委托或枚举。
- 同一个类型的各个部分必须都有修饰符`partial`。
- 使用局部类型时，一个类型的各个部分必须位于相同的命名空间中。
- 一个类型的各个部分必须被同时编译。换言之，**C#**不支持先编译一个类型的某些部分，然后再编译一个类型的某些部分。

局部类型的几个注意点

- 关键字`partial` 是一个上下文关键字，只有和`class`、`struct`、`interface`放在一起时才有关键字的含义。因此`partial`的引入不会影响现有代码中名称为`partial`的变量。
- 局部类型的个数并不必须为2个或2个以上，也可以为1个——虽然这时候已经没有“局部”的必要☺
- 局部类型的各个部分一般是分开放在几个不同的`.cs`文件中，但C#编译器允许我们将它们放在同一个`.cs`文件中。

在局部类型上应用特性

- 局部类型上的特性具有“累加”效应。换言之，如下的代码：

```
[Attr1, Attr2("hello")]  
partial class A { }
```

```
[Attr3, Attr2("goodbye")]  
partial class A { }
```

- 相当于在A类上应用了如下特性：

```
[Attr1, Attr2("hello"), Attr3, Attr2("goodbye")]  
class A { }
```

局部类型上的修饰符

- 一个类型的各个部分上的访问保护修饰符必须保持一致性。
- 如果一个类型有一个部分使用了**abstract**修饰符，那么整个类将被视为抽象类。
- 如果一个类型有一个部分使用了**sealed**修饰符，那么整个类将被视为密封类。
- 一个类的各个部分不能使用相互矛盾的修饰符，比如不能在一个部分上使用**abstract**，又在另一个部分上使用**sealed**。

局部类型的基类或接口

- 一个类型的各个部分上指定的基类必须一致。某些部分可以不指定基类，但如果指定，则必须相同。
- 局部类型上的接口具有“累加”效应，换言之，如下的代码：

```
partial class C: IA, IB {...}  
partial class C: IC {...}  
partial class C: IA, IB {...}
```

- 相当于如下代码：

```
class C: IA, IB, IC {...}
```

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

代码演示

局部类型在Visual Studio 2005中的应用

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

Agenda

- 局部类型
- 空属类型
- 静态类
- 讲座总结
- Q&A

空属类型简介

- 空属类型允许一个值类型具有“空值”意义，从而方便很多场合的运算，如数据库中的空字段。
- 空属类型演示：

```
int i = 123;  
int? x = i;  
double? y = x;  
int? z = (int?)y;
```

空属类型的几个注意点

- 空属类型实际上是一个泛型类型 **System.Nullable<T>**。空属类型的基础类型就是 **System.Nullable<T>** 的类型参数，其中 T 必须为值类型。
- 空属类型如果值不为空，可以运用同样的基础类型所具有的运算，如 +, -, *, /
- 空属类型的 **HasValue** 属性用来判断类型是否为空，如果不为空，则可以通过 **Value** 属性来获取它的基础类型的值。

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

代码演示

空属类型的运算

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

Agenda

- 局部类型
- 空属类型
- 静态类
- 讲座总结
- Q&A

静态类简介

- 静态类是只用于包含静态成员的类型，它既不能实例化，也不能被继承。它相当于一个sealed abstract类。
- 静态类的演示

```
static class MyUtility
{
    public const int data;
    public static void Foo() {
        .....
    }
}
```

静态类的几个注意点

- 静态类不能有实例构造器。
- 静态类不能有任何实例成员。
- 静态类上不能使用abstract或sealed修饰符。
- 静态类默认继承自System.Object根类，不能显式指定任何其他基类。
- 静态类不能指定任何接口实现。
- 静态类的成员不能有protected 或 protected internal访问保护修饰符

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

Agenda

- 局部类型
- 空属类型
- 静态类
- 讲座总结
- Q&A

讲座总结

- 局部类型是一个纯语言层的编译处理，不影响任何执行机制——事实上C#编译器在编译的时候仍会将各个部分的局部类型合并成一个完整的类。
- 空属类型允许一个值类型具有“空值”意义，从而方便很多场合的运算，如数据库的空字段。空属类型实际上是一个泛型类System.Nullable<T>。
- 静态类是只用于包含静态成员的类型，它既不能实例化，也不能被继承。它相当于一个sealed abstract类。

您的潜力，我们的动力

Microsoft
微软(中国)有限公司

Agenda

- 局部类型
- 空属类型
- 静态类
- 讲座总结
- Q&A

Q&A

如需提出问题，请单击“提问”按钮并在随后显示的浮动面板中输入问题内容。一旦完成问题输入后，请单击“提问”按钮。

问题和解答 (无问题)

在此会议中尚未解答任何问题。

要向演示者提问，请在此处键入问



提问(A)

删除(D)

问题管理器(Q)

您的潜力，我们的动力

Microsoft®
微软(中国)有限公司

Microsoft®

msdn


MSDN Webcasts